

Solwise Ltd.

Setup instructions for

Solwise SAR703 ADSL Router



Notification is hereby given that Solwise Ltd. reserves the right to modify, change, update or revise this document from time to time as required without the prior obligation to notify any person, company or organization. Further, Solwise makes no warranty or representation, either express or implied, with respect to merchantability, or fitness of its products for a particular purpose.

Solwise Ltd.

13/15 Springfield Way
Anlaby
Hull HU10 6RJ
UK

Tel 0845 458 4558 (local rate)
Fax 0845 458 4559
Email sales@solwise.co.uk
Http www.solwise.co.uk

Setup instructions for Solwise SAR703 ADSL Router

1.	INTRODUCTION.....	7
2.	GETTING STARTED	8
2.1.	Connecting to your network and line	8
2.2.	Configuration Software.....	9
3.	BASIC ROUTER CONFIGURATION.....	11
3.1.	Setup for Kingston Communications (KC).....	11
4.	CLIENT TCP SETUP	12
4.1.	Installing TCP protocol on your PC	12
4.2.	Checking your TCP/IP settings.....	13
5.	TESTING	14
5.1.	Line Negotiation	14
5.2.	Doing a Ping Test.....	14
5.2.1.	Winipcfg.....	14
5.3.	Running your Browser.....	15
6.	PORT FORWARDING.....	15
7.	FIRMWARE UPGRADE	16
8.	USING TELNET OR TERMINAL MODE	17
8.1.	Using TELNET via Ethernet interface.....	17
8.2.	Using terminal program via serial console port.....	18
9.	CLI CONFIGURATION	20
9.1.	Setting up PPP Over ATM (RFC2364) using CLI.....	20
9.2.	Add NAT to PPP over ATM.....	21
9.3.	PPTP Tunnelling Configuration	23

10. MANAGING THE ADSL ROUTER	25
10.1. Booting the ADSL Router from Ethernet Network.....	25
10.2. Upgrading on-board flash memory from Ethernet network	25
10.3. SNMP.....	25
11. ADSL LINK PERFORMANCE STATISTICS	26
12. COMMAND SETS FOR COMMAND LINE INTERFACE	26
12.1. Command line interface conventions.....	26
12.2. Basic system command sets	26
12.2.1. <process>, <process> <command>	26
12.2.2. help	28
12.2.3. (history mechanism)	28
12.2.4. restart	29
12.2.5. system.....	29
12.3. Commands for ISFS and FLASHFS process	29
12.3.1. ISFS and FLASHFS overview	29
12.3.2. isfs cat flashfs cat.....	29
12.3.3. isfs ls flashfs ls.....	29
12.3.4. isfs rm.....	30
12.3.5. flashfs update.....	30
12.4. Commands for Bridge process	30
12.4.1. device add.....	30
12.4.2. device delete	30
12.4.3. device list.....	31
12.4.4. ethertype	31
12.4.5. filter	31
12.4.6. filterage.....	32
12.4.7. flush.....	32
12.4.8. portfilter.....	32
12.4.9. status.....	32
12.4.10. spanning disable enable	33
12.4.11. spanning forwarddelay	33
12.4.12. spanning hellotime.....	33
12.4.13. spanning maxage	34
12.4.14. spanning port <number>.....	34
12.4.15. spanning port <number> disabled enable	34
12.4.16. spanning port <number> pathcost	34
12.4.17. spanning port <number> priority.....	35
12.4.18. spanning priority.....	35
12.4.19. spanning status	35
12.5. Commands for IP process.....	36
12.5.1. arp.....	36
12.5.2. config.....	36
12.5.3. device	37
12.5.4. disable.....	39
12.5.5. enable	39
12.5.6. get.....	40
12.5.7. ipatm abort.....	40
12.5.8. ipatm arp.....	40
12.5.9. ipatm arpserver	40
12.5.10. ipatm files.....	41
12.5.11. ipatm lifetime	41

12.5.12.	ipatm pvc	41
12.5.13.	iphostname	42
12.5.14.	norelay	42
12.5.15.	ping	43
12.5.16.	portname	43
12.5.17.	relay	44
12.5.18.	rip accept	45
12.5.19.	rip allowed	45
12.5.20.	rip boot	45
12.5.21.	rip hostroutes	46
12.5.22.	rip killrelay	46
12.5.23.	rip poison	46
12.5.24.	rip relay	46
12.5.25.	rip relays	47
12.5.26.	rip send	47
12.5.27.	route	47
12.5.28.	routeflush	48
12.5.29.	routes	49
12.5.30.	stats	49
12.5.31.	subnet	49
12.6.	Commands for NAT process	50
12.6.1.	ip nat	50
12.6.2.	nat interfaces	50
12.6.3.	nat inbound	50
12.6.4.	nat info	51
12.6.5.	nat protocol	52
12.6.6.	nat sessions	52
12.6.7.	nat stats	52
12.7.	Commands for PPP process	53
12.7.1.	Console object types	53
12.7.2.	<channel> clear	53
12.7.3.	<channel> disable	54
12.7.4.	<channel> discard	54
12.7.5.	<channel> echo	54
12.7.6.	<channel> echo every	54
12.7.7.	<channel> enable	54
12.7.8.	<channel> hdlc	55
12.7.9.	<channel> info	55
12.7.10.	<channel> interface	55
12.7.11.	<channel> lcpmaxconfigure	55
12.7.12.	<channel> lcpmaxfailure	55
12.7.13.	<channel> lcpmaxterminate	56
12.7.14.	<channel> llc	56
12.7.15.	<channel> pvc	56
12.7.16.	<channel> qos	56
12.7.17.	<channel> remoteip	57
12.7.18.	<channel> svc	57
12.7.19.	<channel> theylogin	57
12.7.20.	<channel> wellogin	58
12.7.21.	bcp	58
12.7.22.	interface <n> localip	58
12.7.23.	interface <n> stats	58
12.7.24.	user	59
12.8.	Commands for SNMP configuration	59
12.8.1.	access	59
12.8.2.	config	60
12.8.3.	trap	60
12.9.	Commands for ADSL process	60
12.9.1.	show rate	60

12.9.2.	show defect.....	60
12.9.3.	down.....	61
12.9.4.	gasp.....	61
12.9.5.	mode glite.....	61
12.9.6.	mode.....	61
12.9.7.	mode multi.....	61
12.9.8.	show error.....	61
12.9.9.	show perf.....	61
12.9.10.	up.....	62
12.9.11.	show id.....	62
13.	RESET TO FACTORY DEFAULTS.....	62
14.	APPENDIX A PRODUCT SPECIFICATIONS.....	62
14.1.	Power Adapter.....	63
15.	APPENDIX B TROUBLESHOOTING.....	63
15.1.	How to Restore Defaults.....	63
15.2.	B.1 Diagnostics with the LEDs.....	63
15.3.	B.2 Problems when configure the Modem via the console port.....	64
15.4.	B.3 Problems when connecting to the Modem via Ethernet.....	64
15.5.	B.4 Problems when accessing the Internet or remote network.....	64
16.	APPENDIX C GLOSSARY.....	65
17.	APPENDIX D GOVERNMENT COMPLIANCE NOTICES.....	69

1. Introduction

1.1 Overview

The SAR ADSL Router features multi-mode ADSL technology that provides a downstream rate of up to 8M bps over existing copper wire lines, which is more than 100 times faster than a traditional 56K analogue modem. And it can be connected to your PC or LAN through the 10Base-T or 100Base-T Ethernet interface.

It is designed to meet both the needs of single user, and multiple users at small office and home office who want fast Internet access. A wide variety of features and interoperability offer scalability and flexibility for all the applications

1.2 Features and Compatibility

The SAR series Router provides the following features:

- Multi-mode ADSL technology supports ITU-T G.hs, G.dmt, G.lite and ANSI T1.413 issue 2 to provide interoperability with most DSLAM equipment.
- ATM (Asynchronous Transfer Mode) protocol allows the QoS(Quality of Service) transmission over a network
- Support for text-based and Windows-GUI based console management over Telnet and serial connection
- Support for remote configuration by your network administrator via IP network.
- Support IEEE 802.1d transparent bridging with spanning tree algorithm.
- Bridge filtering allows a network administrator to control the flow of packets across the router
- NAT : let multiple users on the LAN share one Internet connection simultaneously
- SNMP agent: allows monitoring and configuration by a standard SNMP manager.
- BOOTP/TFTP enable the remote configuration
- Point-to-Point Protocol (PPP)
- RFC 1483 Link Protocol
- Password Authentication Protocol (PAP) and Challenge Handshake Authentication Protocol (CHAP) security under PPP protocol
- IP routing support includes the RIP(Routing Information Protocol) which allows the exchange of routing information on a TCP/IP network
- Flash memory for Software upgrade
- Status LEDs for easy monitoring and troubleshooting
- DNS relay: allows for automatic name resolution when no DNS information is configured by the user.

1.3 What's in the package?

- One ADSL Router
- One 12VDC Adapter
- One RJ-11 Telephone Cable
- One 10Base-T Ethernet straight-through Cable
- One 9-pin to 9-pin RS-232 Cable
- One Software CD containing the User's Guide and configuration software

All packages have been checked carefully for their completeness and functionality before shipped. Please contact the place of purchase if any of the above listed items are missing or damaged.

1.4 Front Panel

The ADSL Router has five status LEDs for diagnostics. You can monitor the LEDs during operation. Following table shows the ADSL Router status LEDs and identifies what each LED light means.

Function	Behaviour	Definition
POWER	Dark	Power off
	Light	Power on
ADSL	Flashing slowly	ADSL training in progress
	Light	ADSL link is establish and ready to transfer data
PC	Dark	Ethernet link absent or power off
	Light	Ethernet link present
RX	Flashing	Receiving data from ADSL link
TX	Flashing	Transmitting data to ADSL link

1.5 Rear Panel

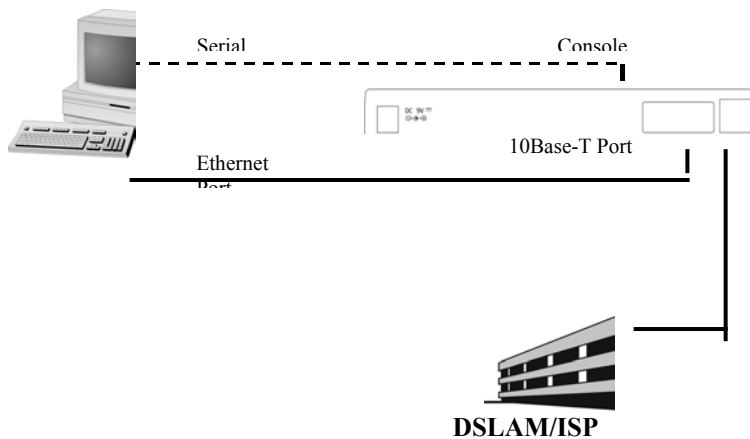
The rear panel of the ADSL Router consist of power jack, Console Port connector, Ethernet connect and ADSL link jack which they means as below:

Function	Definition
ADSL	ADSL jack connect to DSL line from TelCo.
10Base-T	Ethernet interface connect to PC or HUB for LAN.
Console	This is RS232C interface and use to management ADSL Router.
DC12V	The power jack connects to Adapter from wall outlet.

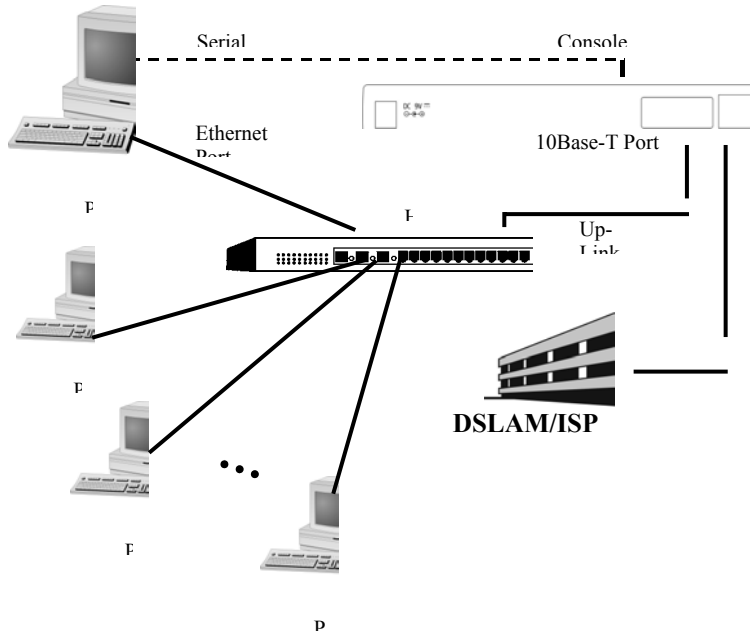
2. Getting Started

2.1. Connecting to your network and line

A Single PC



Multiple PC's



The LAN port on the SAR router is the type designed to be connected to an 'uplink' (or crossover) port on a hub. This means you can also connect the router LAN port direct to the port on a PC LAN card using a standard cat5 LAN cable. If you wish to connect the router to a 'normal' port on a hub then you will need to use a crossover cable or adapter.

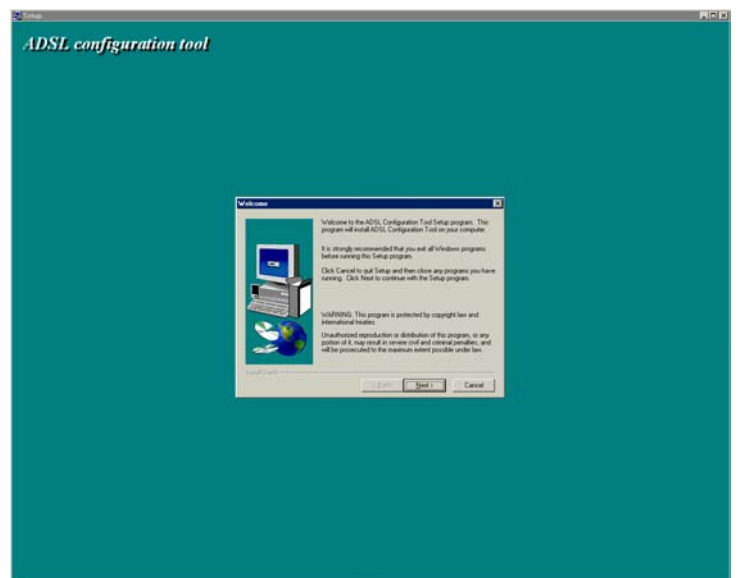
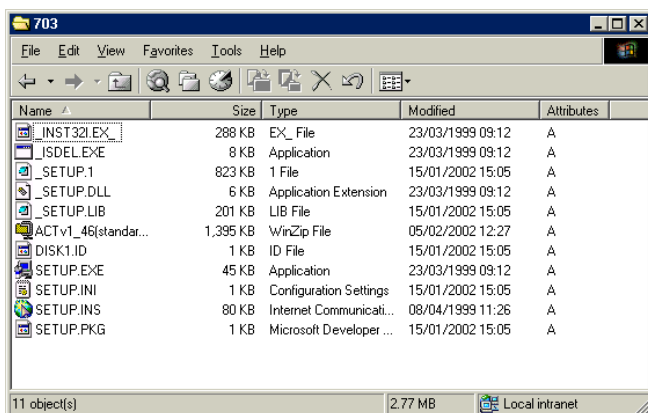
Use the supplied RJ11 phone cable to connect from the ADSL socket on your router to your ADSL phone socket.

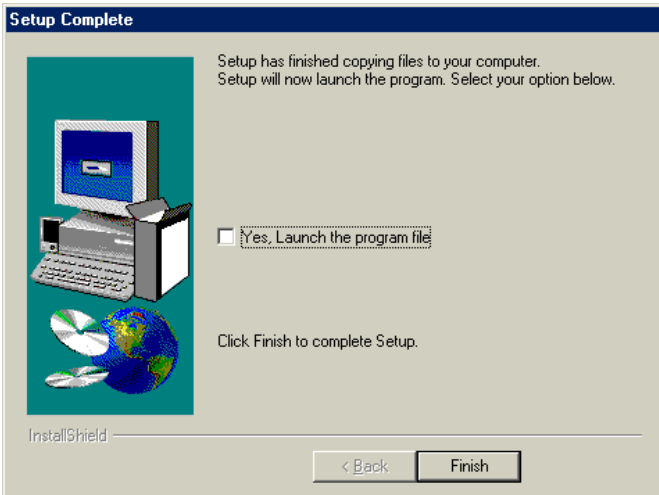
Connect the power jack from the power supply to the power socket on the router and then plug the power supply into a suitable UK power socket: The amber POWER LED on the front of the router should light up.

Configuration can be carried out via Telnet (password is 'password' and default IP address of the router is 192.168.7.1) or using a serial port link to the 9 pin console port on the rear of the router (port settings 9600,8,1,n,no flowcontrol). For those users wanting a more *involved* configuration process configuration *can* be done using a command line interface via telnet or a terminal application (e.g. Windows HyperTerminal) using. However, a more user friendly configuration method is to use the Windows based software tool supplied on the CD which 'talks' to the router via the console port. A 9 pin to 9 pin serial cable is supplied for this purpose and needs to be connected from the Console port of the router to a suitable com port on your PC.

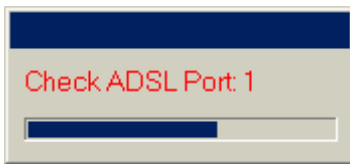
2.2. Configuration Software

The configuration software must be run on an MS Windows PC. Insert the supplied software CD into your CD drive. Then go to the '703' directory and run the setup programme:

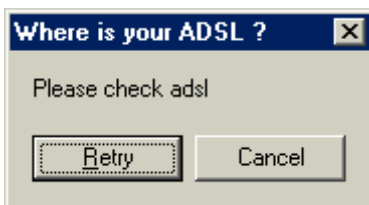




Continue with the software installation, answering the questions as required. Then ensure that the router is connected correctly to your com port and switched on before launching the programme...

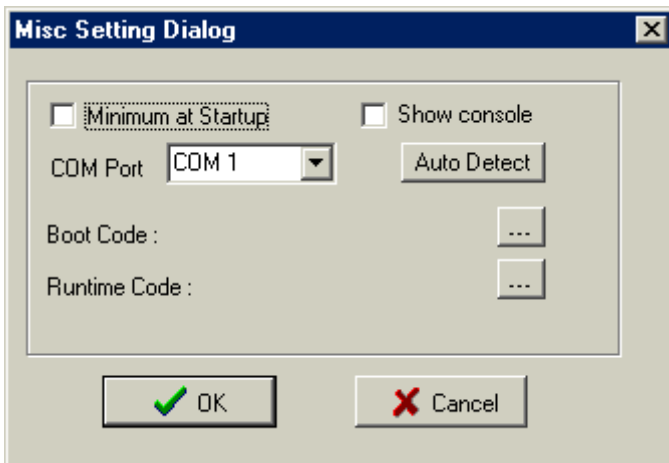


As the software starts up in will automatically try com port 1 to see if it can find the router.

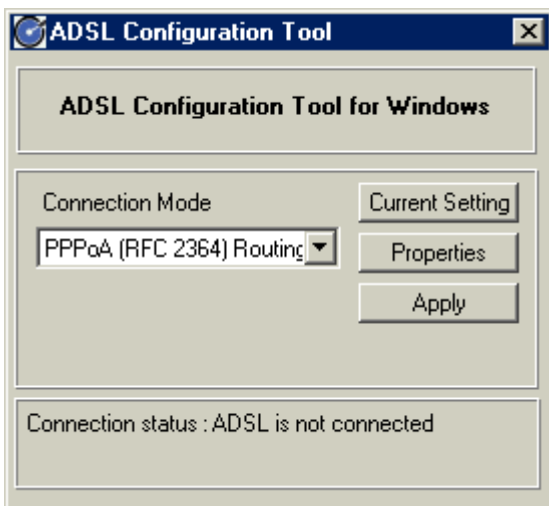


If it fails for any reason, e.g. your router is connected to a different com port then an error message is shown.

In this case just click on Retry.....



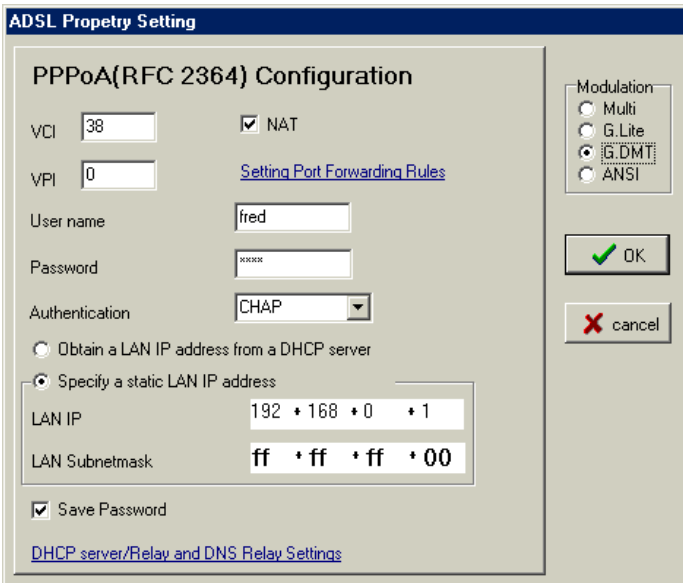
Then click on Auto Detect and the software will automatically search the com ports on you PC for the router.



If successful then the following setup screen is shown (or similar):

3. Basic Router Configuration

Assuming that you've successfully installed and launched your software (as above) then the following screen is shown:



For normal UK operation select PPPoA (RFC 2364) as the Connection Mode and then on Properties to set the router settings:

VCI: For BT use 38, for KC setup see below.

VPI: For BT use 0, for KC setup see below.

Tick NAT (see notes in advanced setup if your want to run without NAT mode).

Enter your User name and Password.

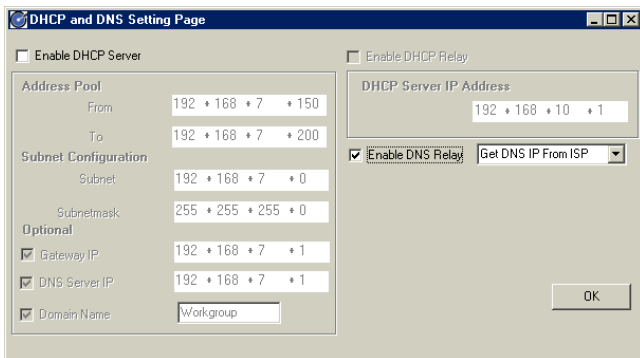
Select CHAP for Authentication.

Assuming you aren't running a DHCP server on your LAN (the normal situation) then enter a suitable IP address for your router and a network mask. Please note that if you already use TCP/IP on your LAN then you need to choose a suitable IP address for the router which fits in with the addresses used by your other network clients. For this example use the address 192.168.0.1 (as

shown).

Click on Save Password and select G.DMT as the Modulation mode.

Now click on DHCP server/Relay and DNS Settings....

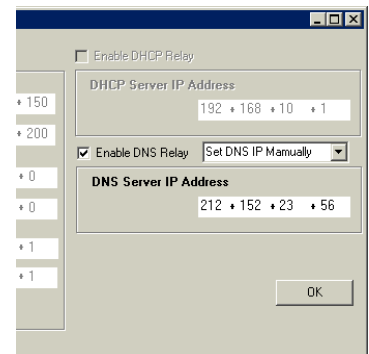


Click on Enable DNS Relay...

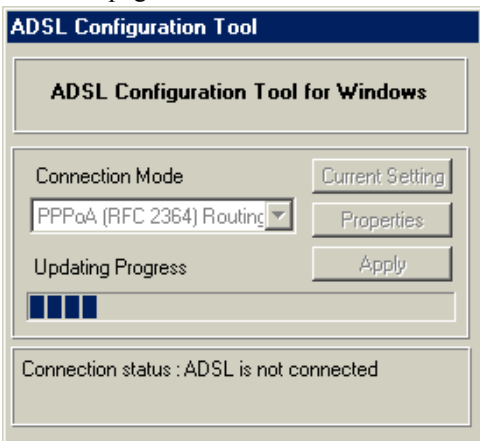
With 'DNS Relay' you tell your clients (PC's) to send their DNS requests to the router (as against sending your requests *through* the router to the ISP's DNS). The router then forwards those requests to the true DNS and then passes any replies back to the client. The advantage of doing this is it simplifies TCP setup on your clients: One less address to find and set.

The default is 'Get DNS IP From ISP' – this means the router will automatically find the DNS address from

your ISP when it connects, The other option is 'Set DNS IP Manually' – in this case you will have to manually enter the DNS address:



Then return to the main properties screen by clicking on OK. Finally click on OK on the main page to return to the front screen ...



Back on the front screen click on Apply to send the new settings across to the router. The router will automatically reset after the new settings have been uploaded.

When the upload has finished close the Configuration software.

That completes the basic router configuration.

3.1. Setup for Kingston Communications (KC)

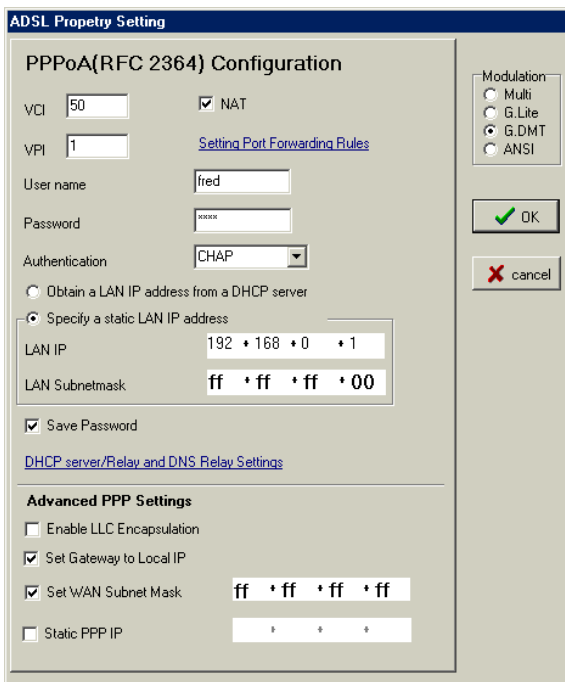
The KC version of ADSL is slightly different from the standard BT offering. Therefore you need to use different settings:

First of all

VCI: For KC use a value of 50

VPI: For KC use a value of 1

Next you need to alter the encapsulation (don't worry if you don't know what this means – it's just describes the way that the PPPoA packets are wrapped). The default router setting for this is Auto i.e. the route will negotiate automatically with the ADSL connection until it finds the encapsulation which suites. This works with BT but not with KC. For KC you need to use 'LLC' encapsulation. In order to configure the router to use 'LLC' you need to enter the 'Advanced PPP Settings' screen. To do this go into the Properties page. Then triple click on the title on the screen that says 'PPPoA(RFC 2364) Configuration'; this means click three times quickly with the left-mouse-button (it takes a few attempts to get this right!). Then the Advanced section is displayed:



For operation on a KC ADSL line tick the box 'Enable LLC Encapsulation'.

Then continue to enter the other settings as detailed above.

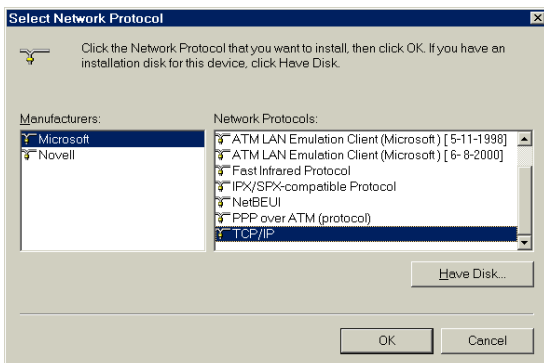
4. Client TCP Setup

4.1. Installing TCP protocol on your PC

If you already use TCP/IP on your PC for LAN use then this section can be skipped.

Goto Start/Settings/Control Panel/Network....

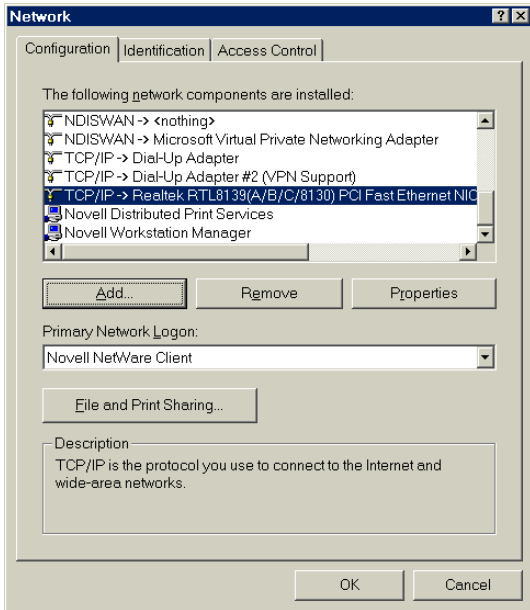
If you already see a line showing TCP/IP protocol such as the example shown left then you can skip this section.



Otherwise click on Add, then select Protocol and then click on Add.. again. Then on the Select Network Protocol screen select Microsoft/TCP/IP...

Click on OK and Windows will then add the protocol to your network setup. You now need to check the settings of the TCP/IP protocol to ensure they are compatible with the router setup. Details on how to do this are shown in the next section.

4.2. Checking your TCP/IP settings

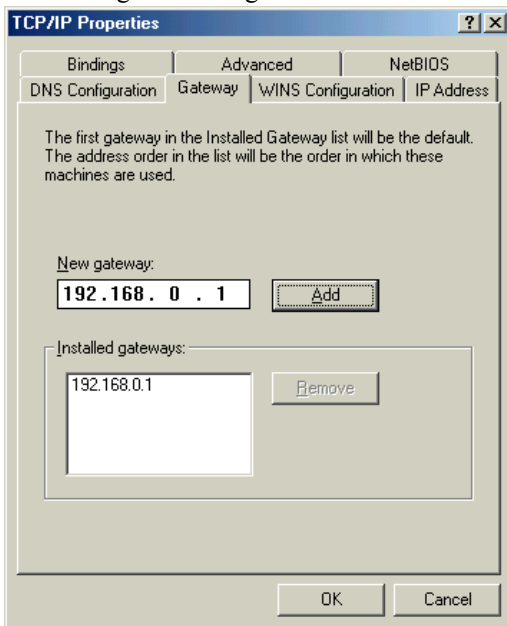


The recommended IP address of the SAR router is 192.168.0.1 (default address is 192.168.7.1 but previous section shows how to change this) on subnet mask 255.255.255.0. In simple terms this means that, in order for your PC to talk to the router, the IP address of your PC's should be in the range from 192.168.0.2 to 192.168.0.254. If you already use TCP as your default network protocol and you don't use IP settings in the required range then you will have to either permanently alter the settings of your computers to suite or change the default address of the router. If you wish to alter the settings of all your PC's to suite then it is probably best to ask the person in charge of your network setup to do this for you

Goto network settings (Start/Settings/Control Panel/Network).

Scroll down the list of network settings until you find the entry showing TCP/IP. There may be several such entries (see example); the one you need is the entry associated with your network card. Select it and then click on Properties:

On the first screen (IP Address) enter a suitable address (e.g. as above) and the subnet mask. You need to ensure that each PC on your LAN has an IP address which is both unique and within the subnet range of the routers address e.g. in the range 192.168.0.2 to 192.168.0.254 (assuming the IP



address for the router is 192.168.0.1 as described above).

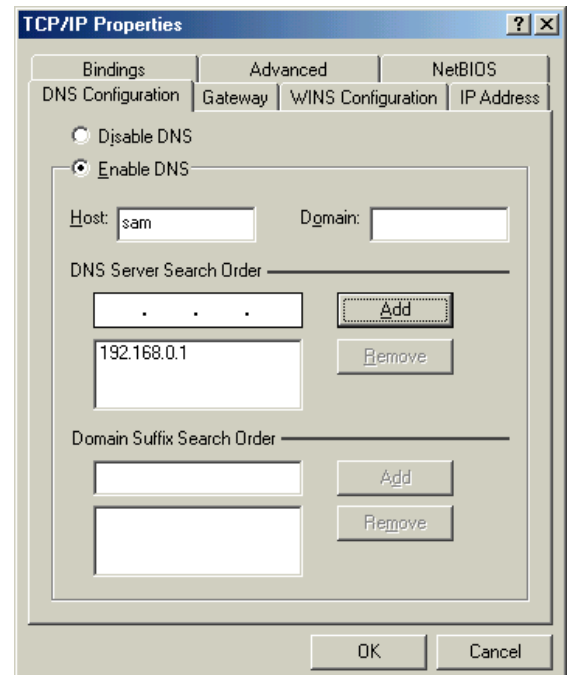
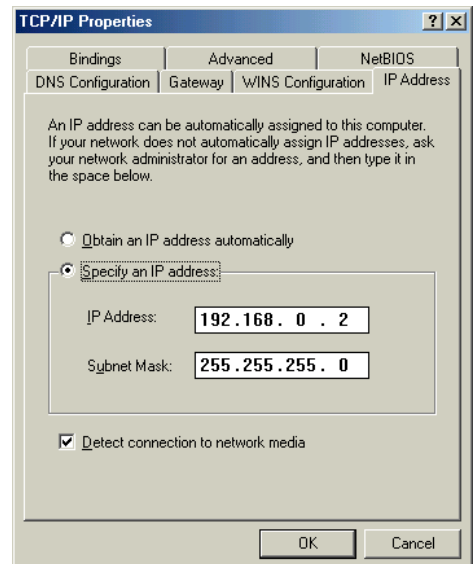
Next, for each PC you must enter a Gateway address. This is the address of the router and tells your PC where to send internet TCP requests:

Finally you must setup DNS Configuration on each PC:

Each PC **MUST** have the address for DNS entered in it's TCP setting. If your PC doesn't have a DNS setting then it will not be able to find any internet sites so it's important that you have this set correctly!

On the DNS Configuration screen you must Enable DNS and then enter a Host name; this can be anything you like – just a made up name will do!

If you've followed the instructions given above then your router will automatically be doing a feature called 'DNS relay'. With DNS relay you tell your clients (PC's) to send their DNS requests to the router (as against sending your requests *through* the router to the ISP's DNS). The router then forwards those requests to the true DNS and then passes any replies back to the client. Therefore, you need to 'add' the address of the router to the DNS list.



Finally click on OK and then OK from the main Network menu.

Windows will now install the revised network settings; please note that your original Windows installation CD might be required. You should then reboot your PC.

That's all there is to it for the client TCP/IP setup. The next section covers basic testing.

5. Testing

5.1. Line Negotiation

After uploading the configuration or powering up the router the following things should happen.

The Power and PC lights should be on; the PC light indicates that the router can see you network and will flash as data goes up and down the LAN.

Initially the ADSL light will go out and then start to flash as it negotiates with the ADSL line. After the router has connected to you ADSL line the ADSL light should be full on.

If the ADSL light does not come on or stays flickering all the time then there might be something wrong with the line.

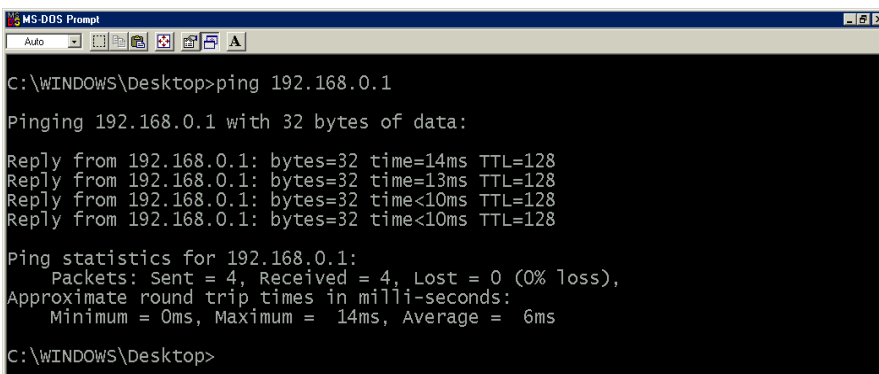
After line synchronisation the ADSL light stays on and then the router verifies the user name and password. It is possible to have a connection (Status light on) but no internet because of failure to logon.

If you now restart the ADSL Configuration software and click on Current Setting the Connection Status (on the configuration screen) should show your upload and download speeds. If the configuration screen shows ADSL not connected all the time then there is a fault.

5.2. Doing a Ping Test

Now drop into DOS (DON'T PANIC! Just do start/run enter 'command' and then click on OK) and at the '>' command prompt type 'ping' followed by the router address (remember we set the address of the router to 192.168.0.1 but if you've changed this remember to use the new address):

You should get a reply if not then you need to check the IP addresses (client address,



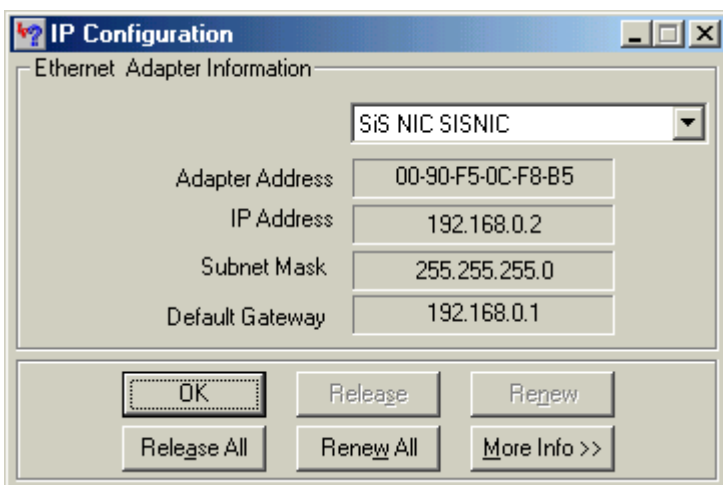
```
MS-DOS Prompt
C:\WINDOWS\Desktop>ping 192.168.0.1
Pinging 192.168.0.1 with 32 bytes of data:
Reply from 192.168.0.1: bytes=32 time=14ms TTL=128
Reply from 192.168.0.1: bytes=32 time=13ms TTL=128
Reply from 192.168.0.1: bytes=32 time<10ms TTL=128
Reply from 192.168.0.1: bytes=32 time<10ms TTL=128
Ping statistics for 192.168.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 14ms, Average = 6ms
C:\WINDOWS\Desktop>
```

gateway, DNS etc) used by your computer match the gateway.

5.2.1. Winipcfg

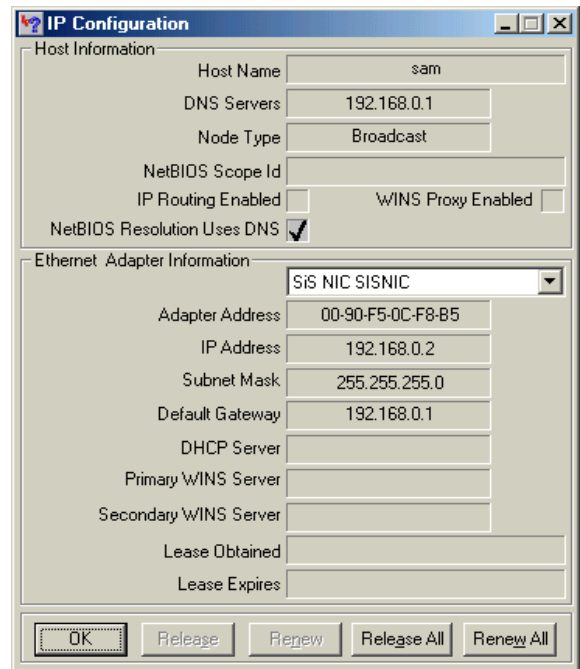
Winipcfg (or ipconfig) in Windows is a nice and useful application which will tell you the IP settings being used by your PC. Goto Start/Run and enter winipcfg:

Click on More Info....



Check that the DNS, IP Address, Subnet, and Default Gateway are what you expect them to be.

If there is an error with one of the settings then go back to your TCP settings and enter the correct setting.



5.3. Running your Browser

After checking you can do a ping and ensuring the router starts up okay (see above) you are ready to run your browser.

If you're running Explorer then click on your Explorer icon to start your browser.

When your browser starts up first check that it's using your LAN for internet access. To do this in Explorer goto Tools/Internet Options/Connections.

Then check that the setting 'Never dial a connection' is selected. No other changes need to be made. If you're using Netscape then no configuration is required for your browser to automatically use your LAN.

Finally enter a valid web site name on the 'Address' line e.g.

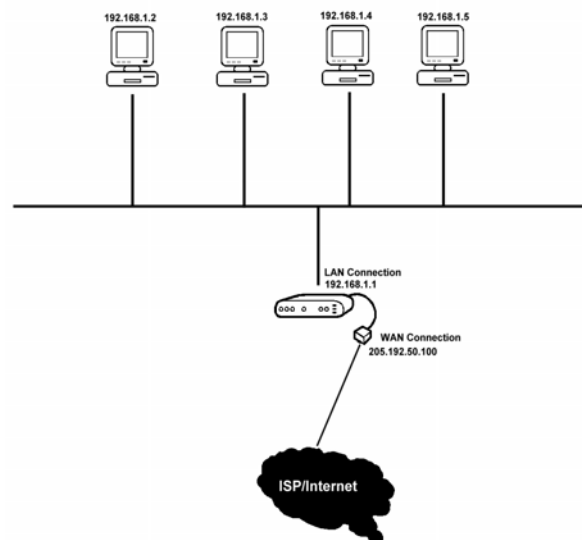
www.solwise.co.uk! It should work!!

6. Port Forwarding

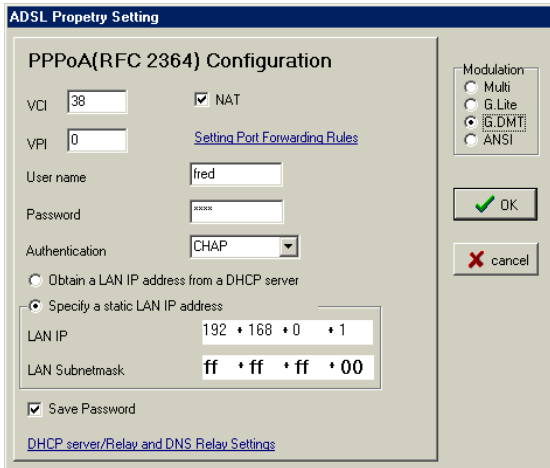
The normal way of using your SAR router is in NAT mode (see setup above).

Most users will use their router with NAT enabled because this allows them to 'share' their internet connectivity across their whole network without needing a block of static IP addresses from the ISP i.e. the ISP sees the whole of your LAN as a single IP address and the router automatically sorts out traffic to the correct local clients:

However using NAT has it's advantages and disadvantages. The advantages are it allows you to easily run multiple PC's through a single user ISP account and it acts as a natural firewall stopping unsolicited incoming traffic. However the disadvantage of NAT is



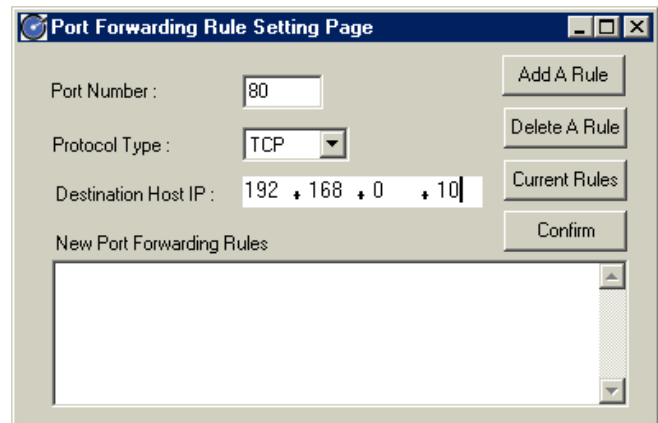
that some software needs the IP address of the PC to be ‘exposed’ to the outside world – this is just what NAT stops!



The normal way around this problem whilst still retaining the NAT mode is to use port forwarding. Port forwarding (also sometimes called pinholeing) tells the router to direct certain incoming traffic to specified local addresses. e.g. if you are wanting to run a local web server on address 192.168.0.10 and you want external internet users to be able to see your server then you need to instruct the router to forward all port 80 TCP requests (the port used by http) to your local server.

To setup port forwarding connect your router to the com port of your PC start the ADSL Configuration programme and click on Properties:

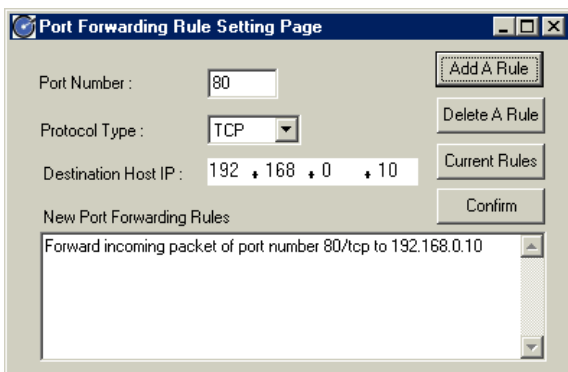
Now click on the ‘Setting Port Forwarding Rules’ link:



Enter the port rule and then click on ‘Add A Rule’:

The Rule will show in the list (see opposite).

When you’ve added all the rules you need click on Confirm. Then OK on the Properties screen and then Apply to send the new configuration across to the router.



Now, whenever anyone tries to access your IP address (That’s the external IP address as allocated to you by your ISP) using port 80 the router will automatically forward the request to your local client at address 192.168.0.10.

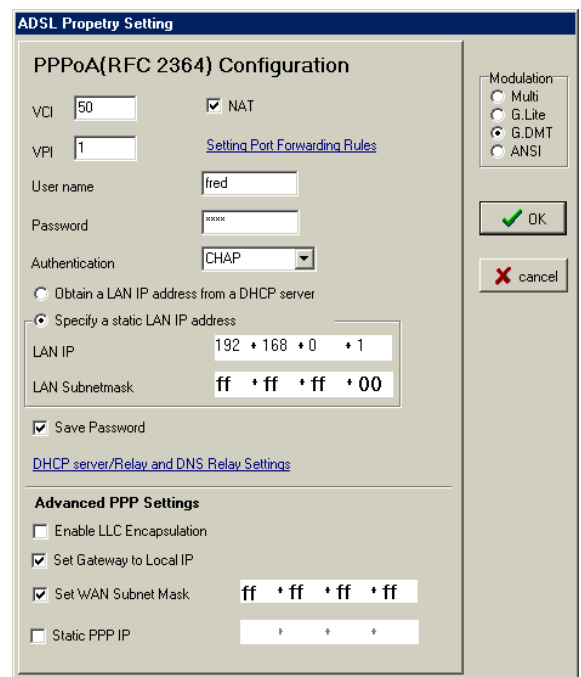
7. Advanced PPP Settings

The Advanced PPP Settings part of the configuration application allows you to alter the less common PPP settings. To do this go into the Properties page. Then triple click on the title on the screen that says ‘PPPoA(RFC 2364) Configuration’; this means click three times quickly with the left-mouse-button (it takes a few attempts to get this right!). Then the Advanced section is displayed:

The only settings you might need to alter on the screen are the LLC Encapsulation – but only tick this if you have an ADSL line from Kingston Communications (do not alter for BT).

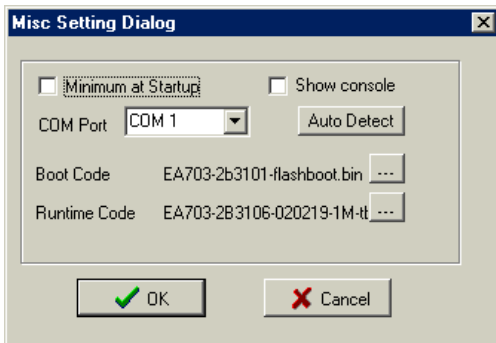
Also the Static PPP IP. This allows you to enter a static IP for your ISP connection (the default is normally to use the dynamic IP address supplied by the ISP).

You are advised to NOT alter any other settings on this section.



8. Firmware Upgrade

As new features are added to the router you may need to upgrade the router software (firmware). This is done using the Windows Configuration software via the console/serial port. Start the configuration software (see above for details). Check via the Properties that the router has a valid LAN IP address. Make sure that the router is also connected via the LAN to your PC (either direct to the network card in your PC or via your network).



Now click with right-mouse-button on the ADSL Configuration Tool startup window and then select Misc Setting from the drop down menu shown:

Click on Boot Code and select the boot rom bin file.

Click on Runtime Code and select the appropriate bin file.

Click on OK..

Next from main startup Window do right-mouse-button and then select Upgrade Firmware.

Now please wait whilst the new firmware is loaded.

After the new firmware is installed you are advised to do a reset to factory defaults and then re-enter your settings (this is because the new firmware may use different settings) – see below on how to enter defaults.

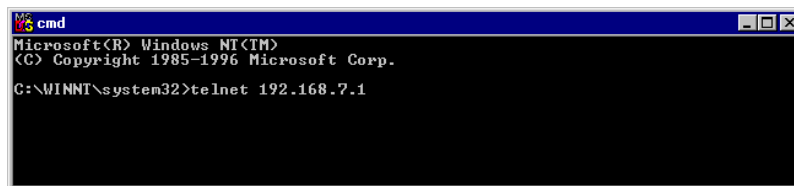
9. Using Telnet or Terminal Mode

9.1. Using TELNET via Ethernet interface

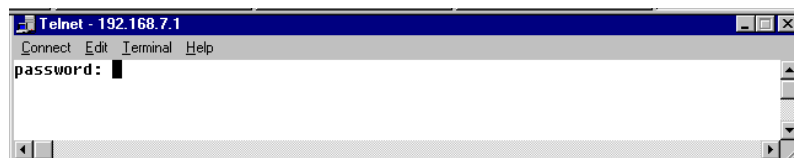
To access the command line interface via Ethernet interface, you can use TELNET to log in the Router from the local Ethernet network using the Ethernet IP address that assigned to your ADSL Router. The Ethernet IP of the ADSL Router is default set to **192.168.7.1**.

Select Start->Programs->MS-DOS Prompt.

Find the IP address of the Router's Ethernet port. Then use TELNET to login the Router. For example, TELNET 192.168.7.1



You will see that a telnet dialog pops up asking for password (case sensitive), then enter password ↵ (“password” is the login password)



Then you will see the following prompt, EA-703 >



Now you are ready to configure the Router by using command line interface (CLI) commands (see below).

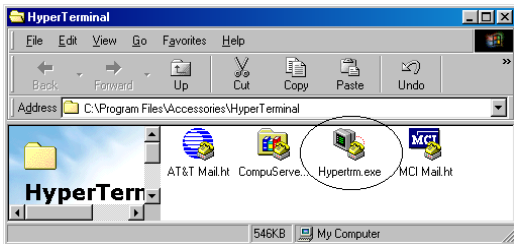
9.2. Using terminal program via serial console port

A terminal can be connected directly to the Serial console port. This requires the use of a terminal emulation software package such as Microsoft HyperTerminal. By default setting, the Router is configured to communicate at a baud rate of 9600. Any standard terminal that support baud rate of 9600 can be connected to the Router's console port. Please configure your serial port as:

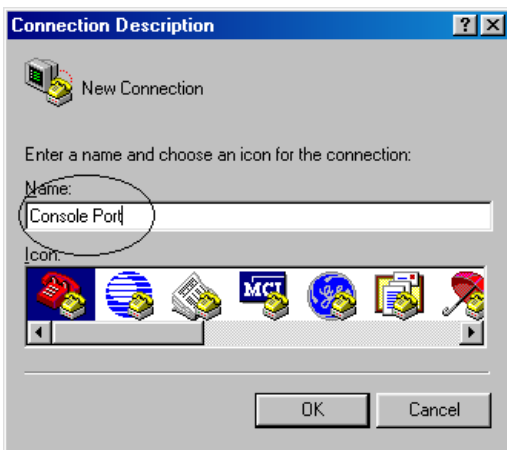
BPS	: 9600
Data bits	: 8
Parity	: None
Stop Bits	: 1
Flow Control	: None

Following steps provide the instructions to log on to the Router via Microsoft HyperTerminal.

Select **Start->Programs->Accessories->HyperTerminal**



Enter a connection name and click **OK**



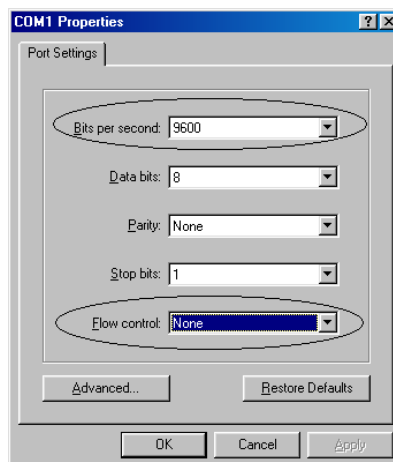
Select properly COM port and click **OK**



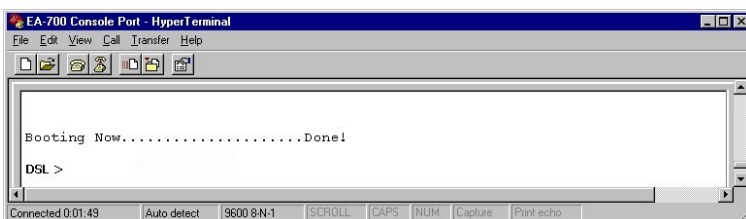
Enter the following parameters :

Bits per second	9600
Data bits	8
Parity	None
Stop bits	1
Flow Control	None

Then click **OK**



When the HyperTerminal window appears, you must press the enter key several time to get the command prompt for the Router's command line interface.

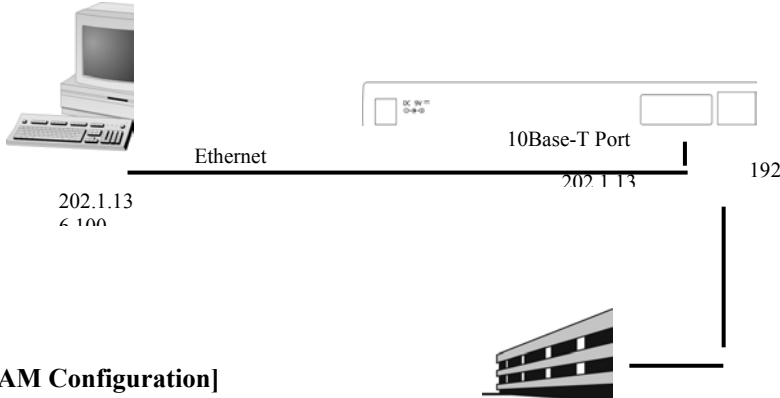


Now you are ready to configure the Router by using the command line interface (CLI) commands.

10. CLI Configuration

10.1. Setting up PPP Over ATM (RFC2364) using CLI

[System configuration]



[ISP/DSLAM Configuration]

IP Address :	192.168.1.1	DSLAM/NSP
Subnet Mask :	255.255.255.0	192.168.1.1
Default Gateway :	192.168.1.2	

[Local PC Configuration]

IP Address :	202.1.136.100
Subnet Mask :	255.255.255.0
Default Gateway :	202.1.136.254

The ADSL Router also can be setup to support RFC 2364(PPP over ATM) with following procedure. Before setup RFC 2364, you have to ensure remove existing RFC 1483 or RFC 1577 configuration with the procedure mentioned above.

- IP dial out over PPPoA

```
> ip device add Ethernet ether //edd 202.1.136.254 ↵
(This is the IP of Ethernet port of ADSL Router)
> ip device add ppp_device ether //ppp/DEVICE=1 ↵
> config save ↵
> restart ↵

> ppp 1 pvc 0 32 ↵
(Set channel 1 to VPI=0, VCI=32)
> ppp 1 wlogin <name> <password> ↵
(This is the login name and password of PPP server)
> ppp 1 enable ↵
> config save ↵
> restart ↵

> ip relay all ↵
> config save ↵
```

```
> restart ↵
```

- Remote bridging over PPPoA

```
> bridge device add edd ↵
```

```
> bridge device add ppp/DEVICE=2 ↵
```

```
> config save ↵
```

```
> restart ↵
```

```
> ppp 1 pvc 32 mac ↵
```

```
> ppp 1 interface 2 ↵
```

```
> ppp 1 enable ↵
```

```
> restart ↵
```

The RFC 2364 configuration also can be removed by following procedure. Please ensure to remove the RFC 2364 configuration before set the ADSL Router to other configuration.

- IP dial out over PPPoA

```
> ip device flush ↵
```

```
> config save ↵
```

```
> restart ↵
```

```
> ppp 1 pvc none ↵
```

```
> ppp 1 wlogin none ↵
```

```
> ppp 1 interface 0 ↵
```

```
> ppp 1 disable ↵
```

```
> restart ↵
```

```
> ip norelay ↵
```

```
> config save ↵
```

```
> restart ↵
```

10.2. Add NAT to PPP over ATM

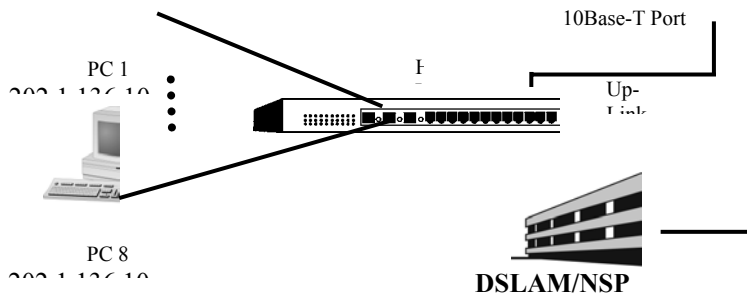
NAT is an IP address conversion feature that translates a PC's local (internal) address into a temporary global (outside/Internet) IP address. NAT is needed when a PC (or several PCs) on a Local Area Network wants to connect to the outside Internet to get to a remote network: NAT swaps the local IP address to a global IP address. Our version of NAT goes one step further by allowing several PCs to share one single IP address to the Internet, thus reducing connection costs. In effect, it allows a whole LAN to connect to the Internet as a single user.

[System configuration]



Ethernet
Port





[ISP/DSLAM configuration]

IP address : 192.168.102.3
 Subnet mask : 255.255.255.0
 Gateway : None

[Local PC 1 configuration]

IP address : 202.1.136.101
 Subnet mask : 255.255.255.0
 Gateway : 202.1.136.254

[Local PC 8 configuration]

IP address : 202.1.136.108
 Subnet mask : 255.255.255.0
 Gateway : 202.1.136.254

The following command tell you how to adding a Network Address Translation protocol to the PPP over ATM(RFC2364) configuration mentioned above. The following command must be added after the “ip device add ...” commands have been given and the Router restarted.

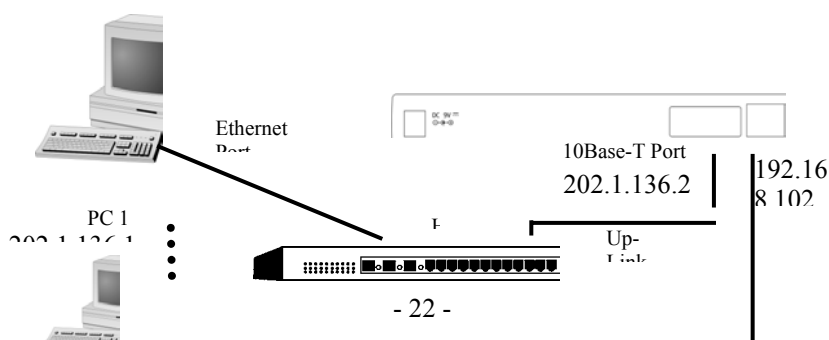
Enables NAT on a PPP over ATM (RFC2364)

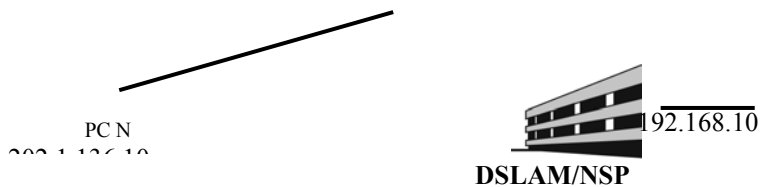
```
> ip nat add ppp_device ↵
```

Enables NAT to PPP over ATM (RFC2364) in Routing mode

The ADSL modem can be setup to adding NAT protocol to a Routing Mode configuration like RFC1483, RFC 1577, RFC 2364 or RFC 2516 with following procedure. The following procedure must be typed after ip device add command have been given and the ADSL Router restarted.

[System configuration]





[ISP/DSLAM configuration]

IP address : 192.168.102.3
 Subnet mask : 255.255.255.0
 Gateway : 192.168.102.2

[Local PC 1 configuration]

IP address : 202.1.136.1
 Subnet mask : 255.255.255.0
 Gateway : 202.1.136.254

[Local PC 8 configuration]

IP address : 202.1.136.100
 Subnet mask : 255.255.255.0
 Gateway : 202.1.136.254

- Add NAT to RFC 2364 to above Routing Mode example

```
> ip nat add ppp_device ↵
```

(ppp_device is the device name same as you configure in RFC 2364 example)
- Remove NAT to RFC 2364 to above RFC 2364 Routing Mode example

```
> ip nat delete ppp_device ↵
```

10.3. PPTP Tunnelling Configuration

The ADSL Router can be configured to supports PPTP as following procedure. But the ADSL Router currently provides the PPTP Access Concentrator (PAC) end of a PPTP tunnel. And the PC must run an OS, which supports PPTP, providing a PPTP Network Server (PNS). Both Win 98 and Win 2000 provide

- PPTP dial out

In the following example, the PC connects to the ADSL modem firstly must be configured as:

IP address of PNS : 192.168.10.1
 Subnet mask of PNS : 255.255.255.0
 Gateway of PNS : 192.168.10.2
 IP address of PAC : 192.168.10.2

```
> ip device add Ethernet ether //edd 192.168.10.2 ↵
> config save ↵
> restart ↵
```

```

> ppp 2 pvc 0 32 ↵
(Set channel 2 to VPI=0, VCI=32)

> ppp 2 interface 0 ↵
> ppp 2 tunnel 1 pptp out ↵
> ppp 2 enable ↵

> pptp bind 192.168.10.2 ↵
> pptp 1 create listen ↵

> config save ↵
> restart ↵

```

- PPTP dial in

In the following example, the PC connects to the ADSL Router firstly must be configured as:

```

IP address of PNS      : 192.168.10.1
Subnet mask of PNS    : 255.255.255.0
Gateway of PNS        : 192.168.10.2
IP address of PAC     : 192.168.10.2

```

```

> ip device add Ethernet ether //edd 192.168.10.2 ↵
> config save ↵
> restart ↵

> ppp 2 pvc 0 32 listen ↵
(Set channel 2 to VPI=0, VCI=32)
> ppp 2 interface 0 ↵
> ppp 2 tunnel 1 pptp in ↵
> ppp 2 enable ↵

> pptp bind 192.168.10.2 ↵
> pptp 1 create 192.168.10.1 ↵
> config save ↵
> restart ↵

```

The PPTP configuration can be removed by following procedure. Please ensure to remove the PPTP configuration before set the ADSL modem to other configuration

- Remove PPTP either dial out or dial in

```

> ip device flush ↵
> config save ↵
> restart ↵

> ppp 2 pvc none ↵
> ppp 2 interface 0 ↵
> ppp 2 tunnel 0 ↵
> ppp 2 disable ↵

> pptp 1 delete ↵
> pptp bind none ↵
> config save ↵
> restart ↵

```


- Remote bridging over PPPoA

```
> config reset bridge ↵
> config save ↵
> restart ↵
```

```
> ppp 1 pvc none ↵
> ppp 1 interface 0 ↵
> ppp 1 disable ↵
> restart ↵
```

11. Managing The ADSL Router

11.1. Booting the ADSL Router from Ethernet Network

By default, the Router is configured to boot from the on-board flash memory. But it is possible boot via Ethernet network as well. The executable image is stored in the local PC and is downloaded to the Router via Ethernet network upon every reset. For this, the Router needs to be configured and also a TFTP/BOOTP utility needs to be installed on the local PC.

- **Router Configuration**

- Turn off the Router and turn it on again
- Keep the * key be pressed
- At the 'Boot from Ethernet, USB or Flash? (E/U/F)' prompt type **E** ↵

- **Local PC Configuration**

To download the software you need a TFTP/BOOTP server. You can use any. Be aware that every time the board is restarted the image will be transferred from the local PC to the Router.

11.2. Upgrading on-board flash memory from Ethernet network

You can update the on-board flash memory after you booting the new firmware from the Ethernet network by issue the following commands.

```
> home ↵
> flashfs rewrite boot.bin ↵
(this command is available for update boot code only)
> flashfs update ↵
> flashfs ls ↵
```

You should see some messages about the file name and file sizes that stored in the Router. If you get "no flash content" something went wrong.

11.3. SNMP

The ADSL Router provides SNMP agent support standard MIBs. SNMP is also used internally for configuration of the router. The active SNMP agent within the Router accepts SNMP requests for status, statistics, and configuration updates. Communication with the SNMP agent occurs over the LAN or WAN connection. Any management application using SNMP over UDP/IP (User Datagram Protocol/Internet Protocol) has access to the local SNMP agent. The following MIBs are supported:

- MIB II (RFC 1213)
- Bridge MIB (RFC 1493)
- PPP/LCP MIB (RFC 1471)
- PPP/Security MIB (RFC 1472)
- PPP/IP MIB (RFC 1473)
- PPP/Bridge MIB (RFC 1474)

12. ADSL Link Performance Statistics

To see the ADSL link performance statistics, you can use the serial console port or the Ethernet interface to access the command line interface.

After power up wait till **ADSL** LED glows steadily. This condition indicates Router has reached “**SHOWTIME**”. Now CLI process commands can be issued at the prompt to retrieve various measurements.

13. Command Sets for Command Line Interface

13.1. Command line interface conventions

- Command line length may be up to 90 characters long.
- The command line interface is case-sensitive
- Parameters in between [and] are optional
- Parameters in between < and > must be entered
- The command line interface prompts for commands with a prompt that indicates the identity of the system. It determines how to indicate the identity as follows :
 - If the SNMP MIB variable sysName.0 exists and is non-empty, that is used first;
 - Otherwise, if a local IP address exists, that is used;
 - Otherwise, the local MAC address is used.

For example, the prompt might look like one of the following

```
DSL>
192.168.7.1>
0:30:eb:ff:0:ff>
```

13.2. Basic system command sets

13.2.1. <process>, <process> <command>

Syntax:

```
<process> <command>
<process>
<process> version
home
home <command>
```

Description:

In these commands, “<process>” can be any of a list of process names known to the console as following :

ip

ppp
snmp
config
bridge
nat
adsl
isfs
flashfs

The former variant sends the command to the process. The latter variant remembers the process name, and sends subsequent commands to the process, as if they had been preceded by the process name, until the command “home” is issued. The prompt is changed to reflect this; moreover, if a “help” command with no arguments is issued, it is passed to the process as usual, but then information about the “home” command is appended to the process’s output by the console.

Example:

```
DSL> isfs help
Commands are:
ls                rm                cat
Type 'help all' or 'help <command>' for more details
DSL> isfs
DSL isfs> help
Commands are:
ls                rm                cat
Type 'help all' or 'help <command>' for more details
DSL isfs> home
DSL>
```

When the console is at the prompt of a particular process, the command "home <command>" or "home <process> <command>" may be used to execute a command as if the user had typed "home" followed by "<command>" or "<process> <command>". However, the console will remain at the same process prompt. The command "home <process>" will change the prompt from the current process to a new process "<process>".

Example:

```
DSL> config
DSL config> help
Commands are:
print            reset            save
Type 'help all' or 'help <command>' for more details
DSL config> home help
Commands are:
adsl            bridge  config  flashfs  ip
isfs            nat          ppp          restart  snmp
system
Type 'help all' or 'help <command>' for more details
DSL config> home flashfs help
Commands are:
cat            ls            update
Type 'help all' or 'help <command>' for more details
DSL config> home isfs
```

```
DSL isfs> help
Commands are:
ls          rm          cat
Type 'help all' or 'help <command>' for more details
DSL isfs> home
DSL>
```

13.2.2. help

Syntax:

```
help
help <cmd>
help all
<process> help
<process> help <cmd>
<process> help all
```

Description:

Displays a summary of available commands, more detailed information on a particular command, or more detailed information on all commands.

Example:

```
DSL> ip help
Commands are:
arp          config          device disable
enable      help  ipatm  nat
norelay     ping  relay  rip
route       routes          stats  subnet
Type "help all" or "help <command>" for more details
DSL> ip help arp
arp syntax:
arp <cmd> - execute arp subcommand
arp help - list subcommands available
```

13.2.3. (history mechanism)

Syntax:

```
.
```

Description:

Repeats the previous console command.

Example:

```
DSL> ip help arp
arp syntax:
arp <cmd> - execute arp subcommand
arp help - list subcommands available
DSL> .
arp syntax:
arp <cmd> - execute arp subcommand
arp help - list subcommands available
```

13.2.4. restart

Syntax:

```
restart
```

Description:

Reboots the Router

13.2.5. system

Syntax:

```
system
```

Description:

Displays the system type, firmware version and other information.

13.3. *Commands for ISFS and FLASHFS process*

13.3.1. ISFS and FLASHFS overview

The Router requiring storage of configuration data should make use of the ISFS file system. The FLASHFS file system provides permanent storage of files and is not normally used other than at start of day or when re-writing the FLASH. In addition to configuration files, FLASHFS stores the firmware image, which is loaded after system restart.

After system restart and during system initialisation, FLASHFS files are copied into ISFS so that they are accessible by application processes. Typically, applications use the ISFS files to store their configuration data. Changes made to the configuration can be written back into ISFS, and subsequently FLASHFS, with the 'config save' command. During a FLASHFS update, all configuration files in ISFS are written back to FLASH irrespective of whether they have changed or not.

Normally the firmware image is not rewritten. The FLASHFS configuration files can be considered the 'master' copies, and the ISFS files the run time copies. If the ISFS copies are written back to the FLASHFS, the current settings will be preserved. It is possible to read files from FLASHFS directly though this use is deprecated.

13.3.2. isfs cat | flashfs cat

Syntax:

```
isfs cat <file>
```

```
flashfs cat <file>
```

Description:

The `cat` command allows a console user to view the contents of the specified file. Only printable characters are displayed, non-printable characters are represented by a '.' character. Printable characters include all standard printable characters together with carriage return, line feed, and tab.

No output formatting is performed, and no scroll lock function implemented.

Example:

```
cat ipaddresses
```

13.3.3. isfs ls | flashfs ls

Syntax:

```
isfs ls
```

```
flashfs ls [-l]
```

Description:

The `ls` command allows a console user to list the files present in the filesystem.

The FLASHFS '-l' option displays more detailed information (logical address within FLASH and linked list information).

Example:

```
ls
```

13.3.4. isfs rm

Syntax:

```
isfs rm <file>
```

Description:

The `rm` command allows the user to remove a file from the ISFS file system. The memory used to store the file is freed. A subsequent FLASHFS update will write the new, shorter, ISFS files into FLASHFS, providing an implicit `rm` function for FLASHFS.

Note: If the file removed is the only file that would be stored in FLASHFS as type `fixed`, the file will remain in FLASHFS as the fixed file area will not be re-written during an update.

Example:

```
> isfs rm foo
```

13.3.5. flashfs update

Syntax:

```
flashfs update
```

Description:

The `update` command instructs FLASHFS to update the FLASH memory from the files contained in the ISFS file system.

Example:

```
> flashfs update
```

13.4. Commands for Bridge process

13.4.1. device add

Syntax:

```
device add <device>
```

Description:

This command adds a device to the bridge configuration. Attempts to add the bridge itself or an existing device to the bridge are rejected. Attempts to add unsupported devices are rejected. There is a limit on the number of devices that can be attached to the bridge. If a device is successfully added to the bridge, it will only become active after the configuration is saved and the system is rebooted. If the device being added is from a process which supports multiple devices, the `/DEVICE` attribute must be specified as part of the device name. The table below shows devices, which may be attached to the bridge, although not all systems may support all devices.

<code>lec1</code>	Forum LAN emulation	<code>alecjade</code>
<code>edd</code>	Ethernet driver	<code>bun_ethernet</code>
<code>ppp</code>	Point-to-Point protocol	<code>pp</code>

Configuration saving saves this information.

Example:

```
DSL bridge> device add edd
DSL bridge> device add ppp/DEVICE=2
```

13.4.2. device delete

Syntax:

```
device delete <device>
```

Description:

This command deletes a device from the bridge configuration. The changes will only take place after the configuration is saved and the system is rebooted. The syntax of the device name is the same as that for the `device add` command.

Configuration saving saves this information.

Example:

```
DSL bridge> device delete edd
```

13.4.3. device list

Syntax:

```
device list
```

Description:

This command lists all the devices that are currently attached to the bridge. It does not show the stored configuration (which can be seen with the `config print` command).

Example:

```
DSL bridge> device list
```

13.4.4. ethertype

Syntax:

```
ethertype [<port> any|ip|pppoe]
```

Description:

This command enables filtering of Ethernet packets according to the `ETHER_TYPE` field in the header. Only packets of the type specified using this command will be **sent** on the port specified; packets of all types will always be **received**. By default, all bridge ports are set to “any”, which means that the type of the packet will never be checked. The meaning of the other options is as follows:

Option Permitted `ETHER_TYPE` values

“ip”	0x0800 – IP
	0x0806 – ARP
“pppoe”	0x8863, 0x8864 – PPP Over Ethernet (RFC 2516)

The port is specified as an integer, as displayed by the `device list` command. When using this command in the `initbridge` configuration file, ports are numbered in the order in which the `device add` commands are given, starting from 1.

If no arguments are given, the current settings for each port are displayed.

Example:

```
DSL bridge> ethertype 2 any
```

13.4.5. filter

Syntax:

```
filter
```

Description:

This command shows the current contents of the bridge’s filter table. The MAC entries for each device are shown in turn together with the time that the MAC address was last seen by the bridge. The command also shows the current filter ageing time, in seconds, and the number of creation failures since the system was started. Creation failures occur when there is no room left in the filter table for a new entry.

Example:

```
DSL bridge> filter
```

13.4.6. filterage

Syntax:

```
filterage [<age>]
```

Description:

This command sets, or displays if no arguments are given, the filter table ageing time. The ageing time is the time after which MAC addresses are removed from the filter table when there has been no activity. The time is specified in seconds and may be any integer value in the range 10...100,000 seconds. This value may also be changed through SNMP. Changing the value of filterage has immediate effect.

Configuration saving saves this information. By default the filter ageing time is set to 300 seconds.

Example:

```
DSL bridge> filterage
```

13.4.7. flush

Syntax:

```
flush [<port>]
```

Description:

This command allows the MAC entries for a specified port, or all ports, to be removed from the filter table. The port number for a device may be determined using the `device list` or `status` commands. If the port number is omitted, all entries for all ports are removed from the filter table.

Example:

```
DSL bridge> flush
```

13.4.8. portfilter

Syntax:

```
portfilter [<source port> all|<destination ports>]
```

Description:

The `portfilter` command allows control over the bridge's forwarding and broadcasting behaviour. By default, when a multicast or an unknown packet is received on a port (referred to above as the source port), it will be forwarded to all other bridge ports (referred to above as the destination ports). Each bridge port may have its behaviour modified separately. The first example below configures the bridge so that packets arriving on port 2 will only be forwarded to ports 3, 4 and 5, and packets arriving on port 3 will only be forwarded to port 1. All other ports retain their default behaviour. Note that this command does not force packets arriving on the source port to be sent to all specified destination ports. The bridge retains its learning behaviour, so unicast packets, once their destination is known to the bridge, will still only be sent to one port. Note also that the bridge itself (for example when attached to the IP router) will always forward to all ports, and will always be forwarded to by all ports. The default behaviour can be restored by calling this command with the argument "all", as shown in the second example. The ports are specified as integers, as displayed by the `device list` command. When using this command in the `initbridge` configuration file, ports are numbered in the order in which the `device add` commands are given, starting from 1. If no arguments are given, the current settings for each port are displayed.

Example 1:

```
DSL bridge> portfilter 2 3 4 5
```

```
DSL bridge> portfilter 3 1
```

Example 2:

```
DSL bridge> portfilter 2 all
```

```
DSL bridge> portfilter 3 all
```

13.4.9. status

Syntax:

```
Status
```


Description:

This command shows the status of the bridge and its ports. The status information for a port includes the SNMP type information about time exceeded packets, packets discarded, etc. It also includes the broadcast history of the port over the last five seconds and the high water mark of packets queued on the bridge for this device.

Example:

```
DSL bridge> status
```

13.4.10.spanning disable | enable

Syntax:

```
spanning disable
spanning enable
```

Description:

When spanning tree operation is disabled, the bridge operates in transparent mode and all bridge ports are set to the forwarding state.

When spanning tree operation is enabled, the state of the bridge's ports is controlled by the spanning tree process.

The `status` command reports the state of the spanning tree process.

Configuration saving saves this information. By default, spanning tree operation is enabled.

Example:

```
DSL bridge> spanning disable
DSL bridge> spanning enable
```

13.4.11.spanning forwarddelay

Syntax:

```
spanning forwarddelay [<time>]
```

Description:

Reads or sets the time in seconds, in which the bridge remains in the listening or learning states, and is used when the bridge is or is attempting to become the root bridge. The forward delay time may be any value between 4 and 30 but it is also constrained by the maximum age and hello times. The forward delay time may also be changed by SNMP command. The `maxage`, `hellotime` and `forwarddelay` times are constrained as follows:

$$2 \times (\text{forwarddelay} - 1) \geq \text{maxage}$$

$$\text{maxage} \geq 2 \times (\text{hellotime} + 1)$$

Configuration saving saves this information. By default the forward delay time is set to 15 seconds.

Example:

```
DSL bridge> spanning forwarddelay 10 ; Sets the forwarding
delay to 10 seconds.
```

13.4.12.spanning hellotime

Syntax:

```
spanning hellotime [<time>]
```

Description:

Reads or sets the time in seconds, after which the spanning tree process sends notification of topology changes to the root bridge, and is used when the bridge is or is attempting to become the root bridge. The hello time may be any value between 1 and 10 and is also constrained by the `forwarddelay` and `maxage` times. The hello time may also be changed by SNMP command.

Configuration saving saves this information. By default the hello time is set to 2 seconds.

Example:

```
DSL bridge> spanning hellotime 5 ; Sets the hello time  
to 5 seconds
```

13.4.13. spanning maxage

Syntax:

```
spanning maxage [<time>]
```

Description:

Reads or sets the maximum age of received spanning tree protocol information before it is discarded, and is used when the bridge is or is attempting to become the root bridge. The maxage time may be any value between 6 and 40 and is also constrained by the forwarddelay and hellotime times. The maxage time may also be changed by SNMP command.

Configuration saving saves this information. By default the maxage time is set to 20 seconds.

Example:

```
DSL bridge> spanning maxage 6 ; Sets the maxage  
time to 6 seconds
```

13.4.14. spanning port <number>

The port commands, described in subsequent sections, control the configuration of the bridge's ports so far as the operation of the spanning tree protocol is concerned. Ports are numbered from 1. Every port on the bridge may be specified by typing `all` instead of a port number.

13.4.15. spanning port <number> disabled | enable

Syntax:

```
spanning port <number> disable | enable
```

Description:

Allows a port to be disabled or enabled. The state of a port may also be changed by SNMP command. A port, which is enabled will take part in the operation of the spanning tree protocol. If enabled, the physical port may be "enabled" or "disabled" as demanded by the operation of the protocol.

Configuration saving saves this information. By default ports are enabled.

Example:

```
DSL bridge> spanning port 1 enable ; Enables port 1 on  
the bridge.
```

13.4.16. spanning port <number> pathcost

Syntax:

```
spanning port <number> pathcost [<cost>]
```

Description:

Reads or sets the cost of using this port. The cost may be any number between 1 and 65535. The cost of the port is used when deciding which is the best path to the root bridge. The cost of a port may also be changed by SNMP command.

Configuration saving saves this information. By default a cost of 10 is assigned to a port

Example:

```
DSL bridge> spanning port 2 pathcost ; Displays the path  
cost for port 2 on the bridge
```

13.4.17. spanning port <number> priority

Syntax:

```
spanning port <number> priority [<portpriority>]
```

Description:

Reads or sets the priority of the port. The priority may be any value between 0 and 255. The priority is used in conjunction with the pathcost to determine the best root to the root bridge. The higher the priority number, the less significant, in protocol terms, the port. The port priority may also be changed by SNMP command.

Configuration saving saves this information. By default a port has a priority of 128.

Example:

```
DSL bridge> spanning port 1 priority ;Displays the
priority for port 1
on the bridge
```

13.4.18. spanning priority

Syntax:

```
spanning priority [<bridgepriority>]
```

Description:

Reads or sets the priority of the bridge. The priority may be any value in the range 0 to 65535. The higher the priority number, the less significant, in protocol terms, the bridge. Where two bridges have the same priority, their MAC address is compared and the smaller MAC address is treated as more significant. The priority of the bridge may be changed by SNMP command.

Configuration saving saves this information. By default the bridge is assigned a priority of 32768.

Example:

```
DSL bridge> spanning priority 4000 ; Sets the bridge
priority to 4000.
```

13.4.19. spanning status

Syntax:

```
spanning status
```

Description:

Reports the status of the spanning tree. If spanning tree operation is disabled, a message is printed to that effect and no other information is displayed. When spanning tree operation is enabled, the following information is displayed:

- The identifier of the bridge.
- The identifier of the root bridge.
- The root port for this bridge.
- The root path cost: how far the bridge is from the root
- The various spanning tree time values as defined by the current root bridge:
- The maximum age of spanning tree information before it is discarded: max age time.
- The amount of time between configuration protocol packets: hello time.
- The amount of time delay when ports are changing state: forward delay time.
- For each port:
 - The identifier of the designated bridge
 - The identifier of the designated port for the designated bridge
 - The identifier of the designated root bridge

Example:

```
DSL bridge> spanning status
```

13.5. Commands for IP process**13.5.1. arp****Syntax:**

```
arp add <i/f> <IP address> <MAC address>
arp delete <i/f> <IP address>
arp flush
arp [list]
arp help [all|<cmd>]
```

Description:

Allows display and manipulation of the ARP table: the list of IP addresses and corresponding MAC addresses obtained by ARP on Ethernet-like interfaces. Normally there is no need to add entries to the table with “arp add”, since they should be discovered by the ARP protocol. Displaying the table with “arp list” (or just “arp”) is sometimes useful, and deleting an entry with “arp delete”, or the whole table with “arp flush”, can sometimes speed up recovery from temporary problems if something unusual has happened. Entries added with “arp add” do not time out like those discovered by use of the ARP protocol, but they are deleted by “arp flush” and will not survive a restart (they are not saved by configuration saving). Note that the ARP table is used only for destinations on directly connected Ethernet-like networks, not for those reached through routers (although the ARP table may be used to discover the MAC address of the router).

Example:

```
DSL> ip arp add ether 192.168.50.1 8:0:20:19:9A:D9
DSL> ip arp
arp add flane 192.168.2.63 00:20:2b:e0:03:87 # 8m58s
arp add flane 192.168.2.109 00:20:2b:03:08:b1 # 2m24s
arp add ether 192.168.50.1 08:00:20:19:9a:d9 # forever
arp add ether 192.168.50.57 00:20:af:2e:fa:3c # 3m25s
DSL> ip arp delete flane 192.168.2.109
DSL> ip arp list
arp add flane 192.168.2.63 00:20:2b:e0:03:87 # 8m46s
arp add ether 192.168.50.1 08:00:20:19:9a:d9 # forever
arp add ether 192.168.50.57 00:20:af:2e:fa:3c # 3m13s
DSL> ip arp flush
DSL> ip arp
# flane ARP table is empty
# ether ARP table is empty
DSL> ip arp
arp add flane 192.168.2.108 00:20:2b:03:0a:72 # 10m58s
# ether ARP table is empty
```

(The last example shows that the MAC address for 192.168.2.108 has been automatically added again, having been discovered by means of the ARP protocol.)

13.5.2. config**Syntax:**

```
config [save]
```

Description:

Displays the IP configuration (not including the “snmp” configuration), or saves it in flash memory. The functionality of the “config” command is also accessible in the standard way through the config process (e.g. “config print ip”), if that process is present. However, when accessed through the config process, the “snmp” configuration is included.

Example:

```
DSL> ip config
device add ether ether //nice mtu 1500 192.168.2.1
device add vlane ether //lane mtu 1500 192.168.55.1
subnet add vlane.home . 192.168.55.0 ff:ff:ff:00
subnet add ether.home . 192.168.2.0 ff:ff:ff:00
rip send ether 2
rip send vlane 2
rip accept ether 1 2
rip accept vlane 1 2
autoloop on
route add default 0.0.0.0 192.168.2.7 00:00:00:00 2 # MAN
relay ether ether
relay ether vlane
relay vlane vlane
ipatm lifetime 60
# IP host table:
# Port table:
router 520/UDP
snmp 161/UDP
tftp 69/UDP
telnet 23/TCP
DSL> ip config save
Updating flash filing system ...
done
ip: configuration saved
```

13.5.3. device

Syntax:

```
device
device add <i/f> <type> [<file>] [mtu <size>] [<IP address>]
device delete <i/f>
device flush
```

Description:

Displays the interfaces that IP is configured to use, or adds an interface to the configuration, or deletes an interface, or all interfaces, from the configuration. However, the commands to change the configuration do not take effect immediately (except when the “device add” command is run at start-up from the initialisation file). It is necessary to save the configuration (e.g. with “ip config save”) and restart the system (e.g. with “ip restart”) before they take effect. “device” will display both the current interfaces and those that have been configured but are not yet in effect.

(Other commands apply only to the devices in effect, rather than to those configured; when adding a device, for example, one may need to issue the “device add” command, then the “config save” and reboot, then issue any other configuration commands that depend on the existence of the device, and then “config save” again.)

“<i/f>” is an arbitrary label for the interface, which is used in referring to it in subsequent commands. (It is often chosen to be the same as “<type>”, though this is perhaps slightly confusing.)

“<type>” specifies the class of interface: Ethernet-like, IP-over-ATM, or loopback. For an Ethernet-like or IP-over-ATM interface, “<file>” specifies the file name that will be opened to access the underlying device. For a loopback interface, “<file>” is not used, and can just be specified as “-“ or omitted altogether.

Several different values of “<type>” specify the same class of interface; they differ in that each implies a different default value for “<file>”. As a result, for the most common interface configurations, “<file>” can be omitted, and one need only specify the appropriate value of “<type>”. The supported values for “<type>” are

Class	<type>	Default file
Ethernet	ether	//nice or //ethernet or //edd
	vlane	//lane
	flane	//lec1
	bridge	//bridge
	IP-over-ATM	atm
	atmpvc	//atm
Loopback	loop	-

“<mtu>” specifies the MTU (maximum transmission unit); that is, the size of the largest datagram (excluding media-specific headers) that IP will attempt to send through the interface. The value specified will be ignored if it is larger than the maximum supported by the interface class, which is currently 1500 except for the loopback interface, unless the IP-over-ATM MTU has been changed; normally there is no point in setting the MTU less than this, so the “<mtu>” option is of little use.

“<IP address>” is the IP address that this system uses on the interface ; if it is not specified, the interface will be disabled until an IP address is supplied with the “ip enable” command. For a loopback interface, the address should be set to 127.0.0.1. (All addresses of the form 127.*.* will then be recognised as loopback addresses, as is normal practice.) For non-loopback interfaces, the subnet mask for the local network will be assumed to be ff:ff:ff:00 (e.g. a class C network); if the correct subnet mask is other than this then it will need to be set with the “subnet” command.

If there is no initialisation file //isfs/resolve (or //isfs/arptable) at all, then default interfaces are configured as if by the “device” commands

```
device add ether ether //edd
device add ether ether //nice (otherwise)
device add atm atm //q93b
```

but in each case only if the file concerned (“//edd”, “//nice”, or “//q93b”) can be opened. Furthermore, if the IP process is given a command line then each argument will be treated as a possible Ethernet-like file to open, given names “ether1”, “ether2”, and so on. For example, if the IP process is defined in the system file as “Process ip is tcp_ip/ip //bridge //lec1 ” (and “//bridge” and “//lec1” can be opened), then the equivalents of the commands

```
device add ether1 ether //bridge
device add ether2 ether //lec1
```

will be processed, in addition to the others above.

Configuration saving saves the interface configuration.

Example:

```
DSL> ip device
```

```

# type dev file IP address
device ether ether //nice mtu 1500 192.168.2.1
device vlane ether //lane mtu 1500 192.168.55.1
DSL> ip device add loop loop 127.0.0.1
Change will have no effect until after config save and restart.
DSL> ip device delete vlane
Change will have no effect until after config save and restart.
DSL> ip device
# type dev file IP address
device ether ether //nice mtu 1500 192.168.2.1
device vlane ether //lane mtu 1500 192.168.55.1 # DELETED
device loop loop - mtu 2048 127.0.0.1 # ADDED
Additions/deletions will have no effect until after config save and restart.

```

13.5.4. disable

Syntax:

```
disable [/f]
```

Description:

Disables all interfaces, or just a specified interface.

Example:

```

DSL> ip disable vlane
DSL> ip device
# type dev file IP address
device ether ether //nice mtu 1500 192.168.2.1
device vlane ether //lane mtu 1500 192.168.55.1 # DISABLED

```

13.5.5. enable

Syntax:

```
enable [/f] [mtu <size>] [<IP address>]
```

Description:

Enables all interfaces, or just a specified interface. Can also be used to set the MTU and IP address on an interface when enabling it (or change them on an interface that is already enabled); see the “device” command for details on these.

Configuration saving saves the MTU and IP addresses, but not the disabled/enabled state.

Example:

```

DSL> ip enable vlane 192.168.56.3
ip/vlane: IP address 192.168.56.3
DSL> ip device
# type dev file IP address
device ether ether //nice mtu 1500 192.168.2.1
device vlane ether //lane mtu 1500 192.168.56.3

```

13.5.6. get

Syntax:

```
get <file>
```

Description:

Reads and executes commands from a file. The commands in the file are in the same format as those documented in this chapter, with no “ip” prefix. They can contain comments, introduced by the “#” character. The “get” command is “hidden”, not shown by “ip help”.

Example:

```
DSL> ip get //isfs/cmdfile
```

13.5.7. ipatm abort

Syntax:

```
ipatm abort <n>
```

Description:

Closes an IP-over-ATM SVC; the number <n> is as displayed by “ipatm files”. If there is still traffic being sent to the destination concerned, IP will soon open a new SVC to the destination.

Example:

```
DSL> ip ipatm abort 14
```

13.5.8. ipatm arp

Syntax:

```
ipatm arp [list]
```

Description:

Lists the cached mappings from IP addresses to ATM addresses; only relevant when using IP-over-ATM with SVCs. (The “list” parameter is optional and makes no difference to the behaviour.)

Example:

```
DSL> ip ipatm arp
192.168.5.72 47.00.83.10.a2.b1.00.00.00.00.00.00.00.00.20.2b.01.00.07.00
192.168.5.33 47.00.83.10.a4.00.00.00.00.00.00.00.00.00.20.2b.01.00.19.00
192.168.5.111 47.00.83.10.e2.00.00.00.20.2b.01.01.a8.00.20.2b.01.01.a8.00
```

13.5.9. ipatm arpserver

Syntax:

```
ipatm arpserver [<i/f> [<ATM address>|here]]
```

Description:

Displays or sets the ATMARP server used for an interface, which must be an IP-over-ATM interface using SVCs. The interface name is optional when displaying: if omitted, the ATMARP servers for all such interfaces are listed. (Since currently there can only be one such interface, this behaviour is present only for possible consistency with future versions.) The parameter “here” causes no ATMARP server to be used; only the local ATMARP cache will be consulted when setting up an SVC. This will normally be used when this machine is the ATMARP server for the local network.

Configuration saving saves this information.

Example:

```
DSL> ip ipatm arpserver
```



```
ipatm arpserver atm here
DSL> ip ipatm arpserver atm 47.0.83.10.a2.0.0.0.0.0.0.0.0.0.0.20.2b.4.3.8.0
DSL> ip ipatm arpserver atm
ipatm arpserver atm 47.00.83.10.a2.00.00.00.00.00.00.00.00.00.00.20.2b.04.03.08.00
```

13.5.10.ipatm files

Syntax:

```
ipatm files
```

Description:

Lists the IP-over-ATM connections, listens, and slots for available connections.

Example:

```
DSL> ip ipatm files
i/f atm 0 transmissions queued, 6 free connections, 4 listeners
0: on atm Connected to 192.168.220.48, 2 rx buffers idle 0ms
1: on atm Listening, 1 rx buffers (in use)
2: on atm Listening, 1 rx buffers (in use)
3: on atm Listening, 1 rx buffers (in use)
4: on atm Listening, 1 rx buffers (in use)
5: on atm Idle, 0 rx buffers
6: on atm Idle, 0 rx buffers
7: on atm Idle, 0 rx buffers
8: on atm Idle, 0 rx buffers
9: on atm Idle, 0 rx buffers
10: on atm Idle, 0 rx buffers
```

13.5.11.ipatm lifetime

Syntax:

```
ipatm lifetime <secs>
```

Description:

Displays or sets idle time-out for IP-over-ATM SVCs: if there is no traffic on an SVC for this period, then it will be disconnected. (It might be disconnected before this period in order to make room for new connections.) There is no way to disable the time-out, but “ip ipatm lifetime 999999” will have much the same effect.

Configuration saving saves this information. The default lifetime is 60 seconds.

Example:

```
DSL> ip ipatm lifetime
Idle lifetime for connections: 1m
DSL> ip ipatm lifetime 90
Idle lifetime for connections: 1m30s
```

13.5.12.ipatm pvc

Syntax:

```
ipatm pvc
ipatm pvc add <i/f> <vci>/[<IP address>][/<pcr>] [<port>]
```

```
ipatm pvc delete <vci> [<port>]
ipatm pvc flush
```

Description:

Lists configured PVCs for use by IP-over-ATM; configures another; deletes one; or deletes all. “<i/f>” is the name of an interface configured for IP-over-ATM using PVCs.

“<vci>” is the VCI to use for the PVC. The range of possible VCIs depends on the system.

“<IP address>” is the IP address of the machine at the other end of the PVC. If it is not specified, TCP/IP will use Inverse ATMARP (RFC 1577) to determine the IP address; if it is specified, then Inverse ATMARP will not be used.

“<pcr>” is the peak cell rate, in cells per second. The default is 60000. (If neither r IP address nor PCR is specified, the “/” after the VCI can be omitted.)

“<port>” is the port name: it must be specified if the machine is a switch, and not otherwise. Configuration saving saves this information.

Example:

```
myswitch> ip ipatm pvc add atm 60 a3
myswitch> ip ipatm pvc add atm 61//50000 b1
myswitch> ip ipatm pvc add atm 62/192.168.4.32 b1
myswitch> ip ipatm pvc
ipatm pvc atm 60//60000 A3
ipatm pvc atm 61//50000 B1
ipatm pvc atm 62/192.168.4.32/60000 B1
```

13.5.13.iphostname

Syntax:

```
iphostname add <IP address> <name>
iphostname flush
iphostname list
iphostname help [all|<cmd>]
```

Description:

Sets up a mapping between an IP address and a symbolic name; deletes all such mappings; lists the mappings; or displays help on the “iphostname” command.

The symbolic names can be used in most IP commands where an IP address is required, and as values of the attributes LHOST and RHOST. They are also displayed and returned as attribute values in place of numerical addresses, when a suitable mapping exists.

The “iphostname” command is “hidden”, not shown by “ip help”.

Configuration saving saves this information.

13.5.14.norelay

Syntax:

```
norelay [all | <i/f> [<i/f>] [forward]]
```

Description:

Turns off forwarding between interfaces; see the “relay” command for more details.

The command “norelay” with no parameters is equivalent to “norelay all”: it turns off all forwarding.

Configuration saving saves this information.

Example:

```
DSL> ip relay
relay ether ether
relay ether vlane
relay vlane vlane
DSL> ip norelay ether vlane forward
relay ether ether
relay vlane ether forward
relay vlane vlane
```

13.5.15.ping

Syntax:

```
ping <IP address> [<tttl> [<size>]]
```

Description:

Sends an ICMP Echo message to the specified IP address.

“<tttl>” (default 30) is the TTL (time-to-live) to use. A crude “traceroute” functionality can be obtained by repeating the “ping” command with increasing TTL values, starting with 1.

“<size>” (default 56) is the data size of the Echo message. This does not include the IP header (20 bytes) and the ICMP header (8 bytes). TCP/IP waits 10 seconds for a reply to the message; if none arrives, it reports the lack of a reply. A reply is an ICMP Echo Reply message, or an ICMP error message reporting destination unreachable, time exceeded, or (as should never happen) a parameter problem. ICMP redirect and source quench messages are reported, but TCP/IP continues to wait for a final reply or time-out.

Example:

```
DSL> ip ping 192.168.4.13 1
ip: ping - 192.168.1.9 reports pkt #5834 to 192.168.4.13: time-to-live
exceeded
DSL> ip ping 192.168.4.13 2
ip: ping - reply received from 192.168.4.13
DSL> ip ping 192.168.77.77
ip: ping - no reply received
```

13.5.16.portname

Syntax:

```
portname add <name> <number>[/<protocol>]
portname flush
portname list
portname read <file>
portname help [all|<cmd>]
```

Description:

Sets up a mapping between a UDP or TCP port and a symbolic name; deletes all such mappings; lists the mappings; reads the mappings from a file; or displays help on the “portname” command. The symbolic names can be used as

values of the attributes LPORT and RPORT provided the protocol type (UDP or TCP) is appropriate. They are also displayed in place of port numbers, when a suitable mapping exists.

“<protocol>” should be either “UDP” or “TCP”; it can be omitted, but that is not very useful. For “portname read”, the file is in the same format as //isfs/services, which is the same as the output from “portname list”. The “portname” command is “hidden”, not shown by “ip help”.

Configuration saving saves this information.

Example:

```
DSL> ip portname flush
DSL> ip portname add someport 105/tcp
DSL> ip portname list
someport 105/TCP
DSL> ip portname read //isfs/services
DSL> ip portname list
router 520/UDP
snmp 161/UDP
tftp 69/UDP
telnet 23/TCP
someport 105/TCP
```

13.5.17.relay

Syntax:

```
relay
relay all | <i/f> [<i/f>] [forward]
```

Description:

Displays or sets what forwarding TCP/IP will do between interfaces. The combinations of setting forwarding can be a bit confusing; they behave as follows:

Command:	Enables forwarding:
relay all	from every interface to every non-loopback interface
relay if1	from if1 to every non-loopback interface, and from every interface to if1
relay if1 forward	from if1 to every non-loopback interface
relay if1 if2	from if1 to if2 and from if2 to if1
relay if1 if2 forward	from if1 to if2

(Don't confuse the “forward” keyword, which indicates one-way relaying, with the term “forwarding”!)

To disable forwarding, use the “norelay” command.

Configuration saving saves this information. By default all forwarding is disabled.

Example:

```
DSL> ip relay
No relaying is being performed
DSL> ip relay ether vlane forward
relay ether vlane forward
DSL> ip relay ether forward
```

```
relay ether ether
relay ether vlane forward
DSL> ip relay ether vlane
relay ether ether
relay ether vlane
DSL> ip relay all
relay ether ether
relay ether vlane
relay vlane vlane
```

13.5.18.rip accept

Syntax:

```
rip accept [all|<i/f>] [none|<version>*]
```

Description:

Controls for which version or versions of RIP (RIP version 1, RFC 1058, or RIP version 2, RFC 1723) TCP/IP will accept incoming information on each interface.

Configuration saving saves this information. By default both RIP versions are accepted on all interfaces (“rip accept all 1 2”).

Example:

```
DSL> ip rip accept all 1 2
DSL> ip rip accept ether 2
DSL> ip rip allowed
rip send ether none
rip send vlane none
rip accept ether 2
rip accept vlane 1 2
```

13.5.19.rip allowed

Syntax:

```
rip allowed
```

Description:

Displays the RIP versions that will be accepted and sent on each interface.

Example:

```
DSL> ip rip allowed
rip send ether 2
rip send vlane 2
rip accept ether 1 2
rip accept vlane 1 2
```

13.5.20.rip boot

Syntax:

```
rip boot
```

Description:

Broadcasts a request for RIP information from other machines. TCP/IP does this automatically when it first starts up, and the routing information should be kept up to date by regular broadcasts from the other machines, so this command is normally of little use.

Example:

```
DSL> ip rip boot
```

13.5.21.rip hostroutes

Syntax:

```
rip hostroutes [off]
```

Description:

Sets or clears the “hostroutes” flag; TCP/IP will accept RIP routes to individual hosts only if this flag is on. If the flag is off, then RIP version 1 routes that appear to be to individual hosts will be treated as if they were to the network containing the host; RIP version 2 routes to individual hosts will be ignored. (The reason for this difference is that RIP version 1 does not allow specification of subnet masks; a RIP version 1 route that appears to be to an individual host might in fact be to a subnet, and treating it as a route to the whole network may be the best way to make use of the information.) To see the state of the flag without changing it, the “config” command must be used.

Configuration saving saves this information. By default the “hostroutes” flag is off.

Example:

```
DSL> ip rip hostroutes off
```

13.5.22.rip killrelay

Syntax:

```
rip killrelay <relay>
```

Description:

Deletes a RIP relay. See “rip relay” for information on RIP relays.

13.5.23.rip poison

Syntax:

```
rip poison [off]
```

Description:

Sets or clears the “poisoned reverse” flag. If this flag is on, TCP/IP performs “poisoned reverse” as defined in RFC 1058; see that RFC for discussion of when this is a good thing. To see the state of the flag without changing it, the “config” command must be used.

Configuration saving saves this information. By default the “poisoned reverse” flag is off.

Example:

```
DSL> ip rip poison
```

13.5.24.rip relay

Syntax:

```
rip relay <RIP version> <name> [/f] [<timeout>]
```

Description:

Configures a RIP relay. RIP relays were designed as a means of using RIP on a non-broadcast medium (currently, only IP-over-ATM); on such an interface, TCP/IP will send RIP information individually to each configured RIP relay, instead

of broadcasting it. However, the RIP relay support has not been recently tested and is not believed to be reliable; furthermore, configuration saving does not save the RIP relay configuration. On a non-broadcast medium, therefore, it is preferable to use static (manually configured) routes.

13.5.25.rip relays

Syntax:

```
rip relays
```

Description:

Displays the configured RIP relays. See “rip relay” for information on RIP relays

13.5.26.rip send

Syntax:

```
rip send [all|<i/f>] [none|<version>*]
```

Description:

Controls which version or versions of RIP (RIP version 1, RFC 1058, or RIP version 2, RFC 1723). TCP/IP will use to broadcast routing information on each interface. If both versions are specified, routing information is broadcast in duplicate, once using each version. Specifying “all” affects all interfaces except the loopback interface (if any).

Configuration saving saves this information. By default RIP version 2 only is used on all non-loopback interfaces (“rip send all 2”).

Example:

```
DSL> ip rip send all 2
DSL> ip rip send ether 1
DSL> ip rip allowed
rip send ether 1
rip send vlane 2
rip accept ether 1 2
rip accept vlane 1 2
```

13.5.27.route

Syntax:

```
route
route add <name> <dest> <relay> [<mask> [<cost> [<timeout>]]]
route delete <name>
route flush
```

Description:

Lists routes; adds or deletes a static route; or deletes all routes.

“<name>” is an arbitrary name specified to “route add” that can be used to delete the route using “route delete”.

“<dest>” is the IP address of the network being routed to (only those bits of “<dest>” corresponding to bits set in “<mask>” are relevant).

“<relay>” is the IP address of the next-hop gateway for the route.

“<mask>” (default ff:ff:ff:00) is the subnet mask of the network being routed to, specified as four hexadecimal numbers separated by colons. For example, 0:0:0:0 is a default route (matches everything without a more specific route), ff:ff:ff:0

would match a Class C network, and ff:ff:ff:ff is a route to a single host. (Note: the default is not always sensible; in particular, if “<dest>” is 0.0.0.0 then it would be better for the mask to default to 0:0:0:0.)

“<cost>” (default 1) is the number of hops counted as the cost of the route, which may affect the choice of route when the route is competing with routes acquired from RIP. (But note that using a mixture of RIP and static routing is not advised.)

“<timeout>” (default 0, meaning that the route does not time out) is the number of seconds that the route will remain in the routing table.

Note that the routing table does not contain routes to the directly connected networks, without going through a gateway. TCP/IP routes packets to such destinations by using the information in the device and subnet tables instead. The “route” command (with no parameters) displays the routing table. It adds a comment to each route with the following information:

- How the route was obtained; one of
 - MAN — configured by the “route” command
 - RIP — obtained from RIP
 - ICMP — obtained from an ICMP redirect message
 - SNMP — configured by SNMP network management;
- The time-out, if the route is not permanent;
- The original time-out, if the route is not permanent;
- The name of the interface (if known) that will be used for the route;
- An asterisk (“*”) if the route was added recently and RIP has not yet processed the change

(the asterisk should disappear within 30 seconds, when RIP next considers broadcasting routing information).

Configuration saving saves this information. (Only the routes configured by the “route” command are saved or displayed by “config”.)

Example:

```
DSL> ip route add default 0.0.0.0 192.168.2.3 0:0:0:0
DSL> ip route add testnet1 192.168.101.0 192.168.2.34
DSL> ip route add testnet2 192.168.102.0 192.168.2.34 ff:ff:ff:0 1 60
DSL> ip route
route add testnet2 192.168.102.0 192.168.2.34 ff:ff:ff:00 1 # MAN 58s/1m via
ether *
route add testnet1 192.168.101.0 192.168.2.34 ff:ff:ff:00 1 # MAN via ether
route add default 0.0.0.0 192.168.2.3 00:00:00:00 1 # MAN via ether
```

13.5.28.routeflush

Syntax:

```
routeflush [<i/f>] [all]
```

Description:

Removes routes from the route table. If “<i/f>” is specified, only routes through the named interface are removed. If “all” is not specified, only host routes (those with a mask of ff:ff:ff:ff) are removed. The “routeflush” command is “hidden”, not shown by “ip help”.

Configuration saving saves this information.

Example:

```
DSL> ip routeflush ether all
DSL> ip routeflush
```


13.5.29.routes

Syntax:

```
routes
```

Description:

Lists routes. (The same as “route”, with no parameters.)

13.5.30.stats

Syntax:

```
stats arp|icmp|ip|tcp|udp [reset]
stats help [<cmd>|all]
```

Description:

Displays or clears a subset of IP statistics.

Example:

```
DSL> ip stats udp
ip: UDP receptions delivered to users: 0
ip: UDP receptions with no users: 170
ip: Otherwise discarded UDP receptions: 0
ip: Transmitted UDP packets: 35
DSL> ip stats udp reset
DSL> ip stats udp
ip: UDP receptions delivered to users: 0
ip: UDP receptions with no users: 0
ip: Otherwise discarded UDP receptions: 0
ip: Transmitted UDP packets: 0
```

13.5.31.subnet

Syntax:

```
subnet
subnet add <name> <i/f> <IP address> <mask>
subnet delete <name>
subnet flush
```

Description:

Lists defined subnets; defines a subnet; deletes a subnet definition; or deletes all subnet definitions.

“<name>” is a label, that can be specified by “subnet add” and later used by “subnet delete” to delete the subnet.

“<i/f>” is not used, but is present for historical reasons and must be specified as either “.” or a valid interface name.

“<IP address>” is the IP address of the subnet being defined (only those bits of “<dest>” corresponding to bits set in “<mask>” are relevant).

“<mask>” is the subnet mask of the subnet being defined, specified as four hexadecimal numbers separated by colons.

A subnet is defined automatically for each interface, with a name formed by appending “.home” to the device name. The only significant use for the “subnet” command is to change the masks for these automatic subnets, if the default masks (see “device” command) are not correct. (Subnet definitions for other subnets *can* also be useful in conjunction with RIP version 1, which does not communicate subnet masks, but this is not very common.)

Configuration saving saves this information.

Example:

```
DSL> ip device
# type dev file IP address
device ether ether //nice mtu 1500 192.168.2.1
device vlane ether //lane mtu 1500 192.168.55.1
DSL> ip subnet
subnet vlane.home . 192.168.55.0 ff:ff:ff:00 vlane
subnet ether.home . 192.168.2.0 ff:ff:ff:00 ether
DSL> ip subnet add vlane.home . 192.168.55.1 ff:ff:fc:0
DSL> ip subnet
subnet vlane.home . 192.168.52.0 ff:ff:fc:00 vlane
subnet ether.home . 192.168.2.0 ff:ff:ff:00 ether
```

13.6. Commands for NAT process

13.6.1. ip nat

Syntax:

```
ip nat add|delete <i/f name>
```

Description:

This command adds or removes NAT functionality from the named interface. The interface name is the name as listed by the `ip device` command. NAT should always be enabled only on the interface connecting to the public network, not the interface connecting to the private network.

Example:

```
> ip nat add ppp_device
```

13.6.2. nat interfaces

Syntax:

```
nat interfaces
```

Description:

The `nat interfaces` command displays the IP router ports on which NAT is currently enabled. For each of these, a status and IP address is listed. The IP address is discovered automatically from the IP stack. The status shows the user whether NAT is currently operational on that interface (“enabled”), or whether NAT is still waiting to find out the interface’s IP address (“not ready”).

Example:

```
> nat interfaces
Name Status IP address
ethernet enabled 194.129.40.2
ppp not ready
```

13.6.3. nat inbound

Syntax:

```
nat inbound list
```

```
nat inbound add <i/f> <port>/<proto> <new IP> [quiet]
nat inbound delete <#>
nat inbound flush
```

Description:

This command enables the user to list or to set up a series of rules, to determine what happens to incoming traffic. By default all incoming packets, other than packets arriving in response to outgoing traffic will be rejected.

The `nat inbound add` command allows packets arriving on a specific port and IP protocol to be forwarded to a machine on the private network. `<i/f>` is an interface name as shown by the `nat interface list` command; `<port>` is the destination UDP or TCP port number to match in the incoming traffic; `<proto>` is the IP protocol, either “udp” or “tcp”; `<new IP>` is the new IP address on the private network which the packet’s destination IP address should be translated to. If a rule is added for an interface on which NAT is not enabled, the rule is added anyway but a warning is printed to alert the user to this fact. `quiet` is a special option which should not normally be issued at the console, and causes this warning to be suppressed. The `quiet` option is automatically added by NAT to when writing its configuration to flash; this is because when a system boots, the NAT process reads in these rules before IP has registered any interfaces

`nat inbound list` shows the current rules for inbound traffic, including all the arguments passed to the `nat inbound add` command.

`nat inbound delete` removes a rule, where `<#>` is the rule number as shown by the `nat inbound list` command.

`nat inbound flush` removes all the rules.

Example:

```
> nat inbound add ppp_device 80/TCP 192.168.219.38
> nat inbound list
# Interface Port/Proto New IP address
1 ppp_device 80/tcp 192.168.219.38
2 r1483 21/tcp 192.168.219.40
> nat inbound delete 2
```

13.6.4. nat info

Syntax:

```
nat info
```

Description:

This command displays the values of various parameters, which are defined in the module file, for example the session table size and the session timeouts. NAT’s current memory usage is also displayed.

Example:

```
> nat info
Interface table size 1 (116 bytes)
Session table size per interface: 128 (6656 bytes)
Total: 6656 bytes
Hash table size per interface: 128 (512 bytes)
Total: 512 bytes
Fragment table size per interface: 32 (640 bytes)
Total: 640 bytes
Max queued buffers: 16
Fragment timeout: 30
Support for incoming fragments: enabled
```

```
Support for outgoing fragments: enabled
Session timeouts:
ICMP query: 10
UDP: 30
TCP (established): 300
TCP (other): 15
Initial port number: 10000
```

13.6.5. nat protocol

Syntax:

```
nat protocols
```

Description:

The `nat protocols` command lists the application level gateways (ALGs) provided in the current image in order to support particular higher-level protocols, and the port or ports, which each ALG monitors

Example

```
> nat protocols
Name Port/IP protocol
ftp 21/tcp
```

13.6.6. nat sessions

Syntax:

```
nat sessions <i/f> [all | summary]
```

Description:

The `nat sessions` command displays a list of currently active NAT sessions on the interface `<i/f>`. In this context, a session is a pair of source IP addresses and port numbers (and corresponding new port number) that NAT regards as one side of an active connection. For each TCP or UDP session active, the source and destination IP address and port number, and the local port number and the age of the session, are printed.

The `all` option causes the `sessions` command to print out information on every session, including sessions, which have timed out. Normally the `sessions` command only shows active sessions (those which have not timed out). The `summary` command does not show detailed information on each session, but only prints out the total number of active, timed out and available sessions.

Example:

```
> nat sessions ppp
Proto Age NAT port Private address/port Public address/port
TCP 34 1024 192.168.219.38/3562 194.129.50.6/21
TCP 10 1025 192.168.219.64/2135 185.45.30.30/80
Total:
2 sessions active
101 sessions timed out
126 sessions available
```

13.6.7. nat stats

Syntax:

```
nat stats <i/f> [reset]
```

Description:

This command displays various statistics gathered by NAT on the interface **<i/f>**. These are cumulative totals since power on, or since the `reset` keyword was given. The `nat stats` command does not provide the total number of packets or bytes transferred, as this information is normally available from the device driver on the interface which NAT is filtering.

Example:

```
> nat stats ppp_device
Outgoing TCP sessions created: 456
Outgoing UDP sessions created: 123
Outgoing ICMP query sessions: 12
Outgoing ICMP errors: 0
Incoming ICMP errors: 6
Incoming connections refused: 2
Sessions deleted early: 0
Fragments currently queued: 0
```

13.7. Commands for PPP process

13.7.1. Console object types

The **ppp** process presents its setup in terms of a number of distinct object types:

The upper limit on the number of each of these objects permitted in a system is configured using the `'config resource'` console command. The current state of each object is saved by `'config save'`.

Channels

The **ppp** process provides a number of PPP connection *channels*. A channel is a single PPP connection. Channels are numbered from 1. Many **ppp** console commands affect only a single channel. The command is prefixed with the channel number.

Users

A *user* is a user name and password. All users must have distinct names. The user console command controls these.

Interfaces

An interface is an internal MAC (Ethernet) device. PPP channels must be associated with an interface to be involved with bridging or routing.

Interface 1 and Channel 1

Interface 1 has some special functions associated with it, allowing dynamic IP address assignment to be performed. Channel 1 is by default associated with Interface 1. These two should be used only for IP dial-out functions, and for this function should be attached to the router interface named `'ppp_device'`.

13.7.2. <channel> clear

Syntax:

```
<channel> clear
```

Description:

Clear all aspects of this channel back to their default settings. If there is an active connection it is torn down.

13.7.3. <channel> disable

Syntax:

```
<channel> disable
```

Description:

Clear the enable flag for a PPP channel. This is the default setting. Disabling does not remove other configured information about this channel. In the PPP state machine, this sets the PPP link to 'closed'. If it is already closed, there is no effect.

Configuration saving saves this information. By default all channels are disabled.

13.7.4. <channel> discard

Syntax:

```
<channel> discard [<size>]
```

Description:

Discard is a PPP LCP packet type, which is like the Echo packet type but does not generate a return. This can be used for more careful tests of data transfer on the link, for instance at sizes near the negotiated MRU. This command sends an LCP Discard packet, of the specified size. If no size is given, a minimal sized packet is sent. Arrival of a Discard packet is logged locally as a level 2 event. The link must be up and operational in order to do the discard test.

13.7.5. <channel> echo

Syntax:

```
<channel> echo [<size>]
```

Description:

Echo is an LCP packet, which is used to test an established PPP link. It solicits a ping-like reply from the far end. This command sends an LCP Echo packet, of the specified size. If no size is given, a minimal sized packet is sent. If a size greater than the remote Maximum Receive Unit size is specified, the value is reduced to the remote MRU before sending. The command waits for 1 second for a reply packet to arrive, and prints whether the reply arrived. If a reply arrives subsequent to this, it is logged as a level 2 event. The link must be up and operational in order to do the echo test. See also the discard test.

13.7.6. <channel> echo every

Syntax:

```
<channel> echo every <seconds>
```

Description:

Echo is an LCP packet, which is used to test an established PPP link. It solicits a ping-like reply from the far end. This command sets a channel to confirm the continued presence of an open PPP connection by sending an LCP echo every few seconds, and requiring an echo reply. The number of seconds between echo requests is specified as a parameter. If 0 is specified, the function is disabled. Use the info all command to read the current state on a channel. Configuration saving saves this information. By default the function is disabled.

13.7.7. <channel> enable

Syntax:

```
<channel> enable
```

Description:

Set the enable flag for a PPP channel. By default this is disabled.

In the PPP state machine, this flag sets the PPP link to 'open'. If it is already open, there is no effect.

Configuration saving saves this information. By default all channels are disabled.

13.7.8. <channel> hdlc

Syntax:

```
<channel> hdlc [1|0]
```

Description:

If 1, use an HDLC header on the front of transmitted packets and require one on received ones. This consists of two bytes, FF-03, and assists in interoperability with some other (non-standard) implementations. If 0, disable this. Call with no argument to find the current setting.

The default value is 0 (disabled). Configuration saving saves this information.

If not set, and a packet is received with an HDLC header, the channel goes into a 'learned HDLC' mode and sends packets with the HDLC header. Thus, interoperation with HDLC-using equipment should not normally require any configuration. Learning occurs in this direction only. Setting `hdlc` to 0 clears this learned state.

Configuration saving does not save the learned state.

13.7.9. <channel> info

Syntax:

```
<channel> info [all]
```

Description:

Provide information about the current settings of this channel. This includes all configured state, and also current protocol information. Specifying 'all' prints out more information. `info` and `status` are synonyms.

13.7.10.<channel> interface

Syntax:

```
<channel> interface <n>
```

Description:

Logically associate the specified channel with the specified interface.

Interface 1 is always the router port. It should be used for any PPP channel over which IPCP communication with the local system's IP router is desired. Other interfaces can be created for bridging. A single PPP channel can only be associated with a single interface, or a single tunnel. Use `info` to find the current setting.

Calling with `n=0` removes any association. This is the default state. Configuration saving saves this information.

13.7.11.<channel> lcpmaxconfigure

Syntax:

```
<channel> lcpmaxconfigure [<n>]
```

Description:

Set the Max-Configure parameter for LCP. This is the maximum number of Configure Requests that will be sent without reply, before assuming that the peer is unable to respond. Call with no argument to find the current setting.

The default value is 10. Configuration saving saves this information

13.7.12.<channel> lcpmaxfailure

Syntax:

```
<channel> lcpmaxfailure [<n>]
```

Description:

Set the Max-Failure parameter for LCP. This is the maximum number of consecutive Configure Naks that will be sent before assuming that parameter negotiation is not converging. Call with no argument to find the current setting.

The default value is 5. Configuration saving saves this information.

13.7.13.<channel> lcpmaxterminate

Syntax:

```
<channel> lcpmaxterminate [<n>]
```

Description:

Set the Max-Terminate parameter for LCP. This is the maximum number of Terminate Requests that will be sent without reply, before assuming that the peer is unable to respond. Call with no argument to find the current setting.

The default value is 2. Configuration saving saves this information.

13.7.14.<channel> llc

Syntax:

```
<channel> llc [1|0]
```

Description:

If 1, use an LLC header on the front of transmitted packets and require one on received ones. This consists of four bytes, FE-FE-03-CF, and is required for PPP Over AAL5 (RFC 2364 p4) when using LLC encapsulated PPP. If 0, disable this. Call with no argument to find the current setting.

The default value is 0 (disabled). Configuration saving saves this information.

If not set, and a packet is received with an LLC header, the channel goes into a 'learned LLC' mode and sends packets with the LLC header. Thus, interoperation with LLC-using equipment should not normally require any configuration. Learning occurs in this direction only. Setting `hdlc` to 0 clears this learned state.

Configuration saving does not save the learned state.

13.7.15.<channel> pvc

Syntax:

```
<channel> pvc [[<port>] <vpi> <vci> [ip|mac] [listen]
```

```
<channel> pvc none
```

Description:

Attach an ATM PVC to the given PPP channel. The port can be specified (only for a multi-port device), and the VPI (default is 0), and the VCI. The allowable range of port, VPI, VCI depends on the ATM driver. Normal limits are 0 only for port, 0 only for VPI, 1..1023 for VCI. If a single argument `none` is supplied, any current connection is torn down. This is equivalent to `svc none` on the channel. In the PPP state machine, providing a link of this form causes the link to be 'up'. Note that `enable` must also be used, to allow the link to become operational. The `ip` or `mac` indicates which form of data is transported over the connection: one of IP data (controlled by the IPCP protocol), or MAC data (for BCP). If neither is provided, `ip` is assumed. If the channel is not linked to an interface, and the channel is for IP data, the channel is linked to interface 1. If the channel is not linked to an interface, and the channel is for MAC data, the channel is linked to interface 2. Providing a PVC setting unsets any SVC setting. See the `svc` command. It is possible for a PVC to become 'down' in the PPP state machine even though the PVC is still there, for instance due to an authentication failure. If in this state, an incoming packet will cause the PPP state machine to go 'up'. If `listen` is specified then this is the server end of a PVC. It will not send out PPP Configure Requests until it first receives a packet over the PVC. When a connection is torn down it goes returns to this state. Use the `info` command to read this information.

Configuration saving saves this information. By default a channel has no connection information.

Example:

```
> ppp 3 pvc 3 32 ; set channel 3 to be (VPI=3,VCI=32)
> ppp 4 pvc ; read PVC settings for channel 4
> ppp 5 pvc 0 ; remove any PVC settings from
channel 5
```

13.7.16.<channel> qos

Syntax:


```
<channel> qos [cbr|ubr] [pcr <pcr-tx> [<pcr-rx>]]
```

Description:

Specify that the VC for a PPP channel should be Constant Bit Rate or Unspecified Bit Rate, and (optionally for UBR) give a Peak Cell Rate for the connection. If two values are specified then they are transmit and receive PCRs respectively. If called while not attached to a VC then the settings are saved for use when a VC is created. If the channel is already attached to a VC then it is closed, and re-opened with the new values. If it cannot be reopened, it remains closed. Configuration saving saves this information. By default channels are established UBR.

Example:

```
> ppp 3 qos cbr pcr 10000 ; set channel 3 to be CBR limited
at 10000 cells/sec
```

13.7.17.<channel> remoteip

Syntax:

```
<channel> remoteip [<ipaddress>]
```

Description:

If a PPP link is established using IPCP, this call causes the channel to provide the given IP address to the remote end of the connection. PPP will refuse to complete the connection if the other end will not accept this. This is normally used for channels on which the remote party dials in, to allocate the IP address to that remote party. Call with no argument to find the current setting.

Call with 0.0.0.0 to remove any setting. This is the default state.

Configuration saving saves this information.

13.7.18.<channel> svc

Syntax:

```
<channel> svc listen [ip|mac]
```

```
<channel> svc addr <addr> [ip|mac]
```

```
<channel> svc none
```

Description:

Specify that the VC for a PPP channel should be an SVC (i.e. created by signalling). This can either be by listening for an incoming call, or by making an outgoing call to a specified ATM address.

The outgoing call or listen occurs immediately. If the call fails it will be retried after a few seconds. In the PPP state machine, providing a connection of this form causes the channel to be 'up' or 'down'. Note that `enable` must also be used, to allow the link to become operational. Outgoing and incoming UNI signalling calls are identified by a BLLI value that identifies PPP. (Aside: A BLLI of length 3 bytes is used, hex values 6B, 78, C0.) If the channel is already attached to an SVC or PVC then it is closed, and re-opened with the new settings. If it cannot, it remains closed. If a single argument `none` is supplied, any current connection is torn down. This is equivalent to `pvc none` on the channel. The `ip` or `mac` indicates which form of data is transported over the connection: one of IP data (controlled by the IPCP protocol), or MAC data (for BCP). If neither is provided, `ip` is assumed. Providing an SVC setting unsets any PVC setting. See the `pvc` command.

Configuration saving saves this information. By default a channel has no connection information.

Example:

```
> ppp 3 svc 47.00.83.01.03.00.00.00.00.00.00.00.00.00.20.2b.00.03.0b.00
> ppp 4 svc listen ; listen for incoming call
> ppp 7 svc none ; tear down connection, remove setting
```

13.7.19.<channel> theylogin

Syntax:

```
<channel> theylogin pap|chap|none
```

Description:

This command describes how we require the far end to log in on this channel. Requiring the other end to log in most frequently happens when they dial us (rather than the other way round), so this is likely to be one of several channels which are set using `svc listen`. Because of this, exact names and passwords are not attached to individual channels but are matched to particular users, as defined using the `user` command. This command specifies that when using this channel, the user must log on using the specified protocol, and that they must provide any name/password combination which has been defined for that protocol, using the `user` command. To remove this information on a channel, call `theylogin` with a single argument of `none`.

Configuration saving saves this information. By default no login is required.

13.7.20.<channel> welogin**Syntax:**

```
<channel> welogin <name> <password> [pap|chap]
```

```
<channel> welogin none
```

Description:

This command describes how we should log in to the far end when a connection is established.

A name and password are supplied, and whether these should be used with the PAP or CHAP authentication protocol. CHAP is the default. To remove this information on a channel, call `welogin` with a single argument of `none`. If `chap` is specified, we will also log in using `pap` if the other end prefers this. If `pap` is specified we will only log in using `pap`.

Configuration saving saves this information. By default no login is performed.

13.7.21.bcp**Syntax:**

```
bcp stp|nostp
```

Description:

This command describes parameters for BCP, the Bridge Control Protocol, which is used to transport MAC (Ethernet) packets over the PPP link. See the protocol conformance section of this spec for BCP option settings which are not controllable. If `stp` is specified, the Spanning Tree Protocol is in use by the Bridges, to control bridge loops. In this case STP frames should be carried over any links using BCP. If `nostp` is specified, STP frames should not be carried. Configuration saving saves this information. By default STP is not supported.

13.7.22.interface <n> localip**Syntax:**

```
interface <n> localip <address>
```

Description:

This command describes parameters for IPCP, the IP Control Protocol, when providing the server end of an IPCP connection. The server knows its own IP address (and may allocate an IP address to the remote end). This command tells the PPP process, for a particular interface, the local IP address to be associated with the local end.

For interface 1, this should be the same IP address as possessed by the device `ppp_device` in the IP stack. See the IP dial-in server console example, at the start of this section. If PPP channels are now associated with this interface, remote users can dial in to those channels and will be connected to the IP stack. They can be allocated IP addresses, see the command `<channel> remoteip`.

Call with 0.0.0.0 to remove any IP address setting. This is the default state.

Configuration saving saves this information

13.7.23.interface <n> stats**Syntax:**

```
interface <n> stats
```

Description:

The interface is regarded by the operating system as an Ethernet-like device like other Ethernet devices. It also provides an Entry to SNMP providing basic information about traffic through the interface. This command shows the basic information about byte and packet traffic through the interface, in SNMP terms.

13.7.24.user

Syntax:

```
user add <name> [pwd <passwd> [pap|chap]]
user [<name>]
user delete <name>|all
```

Description:

This command stores information about a particular login name/password combination. This is referred to as a ‘user’, regardless of whether it represents an individual. When `user` is called on its own, information about all existing users is listed. When `user <name>` is called with no further arguments, details of that user alone are printed. Passwords are not shown.

Use `user delete` to delete an individual user by name, or to delete all users.

Use `user add <name>` to create a new user or update an existing one. The password is stored, and the authentication protocol which must be used for this user.

If a user is deleted or changed, existing sessions are not affected.

Configuration saving saves this information.

13.8. Commands for SNMP configuration

13.8.1. access

Syntax:

```
access [read | write] <community> [<IP addr>]
access delete <community> [<IP addr>]
access flush
access list
```

Description:

The “read” and “write” options configure a community name that can be used for read-only or read-write access, respectively. If an IP address is specified, then the community name is valid only for SNMP requests issued from that IP address. (It should be noted that this can be rather weak security, since it is possible for the source address of IP packets to be forged.) The same community name can be configured several times with different IP addresses, to allow access with the same community name from a number of different machines. The number of access records (community names paired with optional IP addresses) that can be configured is limited only by available memory.

The “delete” option deletes an access record. The IP address must match exactly; if it is not specified, only a matching access record that has no IP address will be deleted. The “flush” option deletes all access records. The “list” option lists the access records.

Configuration saving saves the access records.

By default, if there are no access records in the `snmpinit` file, no SNMP management is allowed.

Example:

```
DSL> snmp access list
access read public
access write password
DSL> snmp access write xyzy 192.168.4.73
DSL> snmp access delete password
DSL> snmp access list
```

```
access read public
access write xyzzy 192.168.4.73
```

13.8.2. config

Syntax:

```
config [save]
```

Description:

Displays the configuration (as from “access list” and “trap list” together), or saves it to flash memory.

Example:

```
DSL> snmp config
access read public
access write xyzzy 192.168.4.73
trap add public 192.168.4.73 162
```

13.8.3. trap

Syntax:

```
trap add <community> <IP addr> [<port>]
trap delete <community> <IP addr> [<port>]
trap flush
trap list
```

Description:

Manipulates the list of destinations to which SNMP traps will be sent. The default UDP port to send traps to is 162, but it may be overridden by specifying <port>.

Configuration saving saves the list of trap destinations.

Example:

```
DSL> snmp trap flush
DSL> snmp trap add public 192.168.4.73
DSL> snmp trap add public 192.168.4.74 999
DSL> snmp trap list
trap add public 192.168.4.73 162
trap add public 192.168.4.74 999
```

13.9. *Commands for ADSL process*

13.9.1. show rate

Syntax:

```
Show rate
```

Description:

This command displays the channel data of the ADSL link. It will not return any message if ADSL link is not established yet.

13.9.2. show defect

Syntax:

```
show defect
```

Description:

This command displays the defects data of the ADSL link. It will not return any message if ADSL link is not established yet.

13.9.3. down

Syntax:

```
down
```

Description:

Disable ADSL link

13.9.4. gasp

Syntax:

```
gasp
```

Description:

Send dying gasp

13.9.5. mode glite

Syntax:

```
mode glite
```

Description:

Set G.Lite mode

13.9.6. mode

Syntax:

```
mode
```

Description:

This command displays the current mode of the ADSL link.

13.9.7. mode multi

Syntax:

```
mode multi
```

Description:

Set multi mode

13.9.8. show error

Syntax:

```
show error
```

Description:

This command displays the line data of the ADSL link. It will not return any message if ADSL link is not established yet.

13.9.9. show perf

Syntax:

```
show perf
```

Description:

This command displays the performance counters data of the ADSL link. It will not return any message if ADSL link is not established yet.

13.9.10.up

Syntax:

up

Description:

Enables ADSL link

13.9.11.show id

Syntax:

show id

Description:

This command displays the vendor id of local equipment and remote equipment. It will not return any message if ADSL link is not established yet.

14. Reset to Factory Defaults

To reset the router to defaults you need to enter a CLI command. Therefore you will have to connect to the router via either Console or Telnet (see above for details). Then at the ‘..703 >’ prompt enter the command:

```
config save default
```

Then wait whilst the default settings are loaded.

Please note: If you’ve changed the LAN IP address of the router then resetting to factory default will reset the router to 192.168.7.1 and hence kill your telnet connection. You will then need to go in via the console to change the IP address.

15. Appendix A Product Specifications

PC interface	10Base-T Ethernet through RJ-45 connector
ADSL interface	ADSL line through RJ-11 connector
Console Port	RS-232
Standard Compliance	ANSI T1.413 issue2 ITU-T G.992.1(Full rate DMT) ITU-T G.992.2(Lite DMT) ITU-T G.994.1(Multimode) RFC 1483 BPDU(Bridge Ethernet over ATM PVC, LLC/SNAP) RFC 1483 RPDU(Routed IP over ATM PVC, LLC/SNAP) RFC 1577(Classic IP over ATM, MTU=1500) RFC 2364(PPP over ATM) ATM Forum INU 3.0, 3.1 and 4.0 signalling*** ATM Forum ILMI 4.0*** ATM Forum LANE 1.0 client, MTU=1516, over SVC only*** ATM supports AAL5, AAL3/4 and AAL0 ATM Traffic shaping supports CBR and UBR OAM F4 and F5 segment end-to-end loopback are supported(F4 on all VPIs, F5 on VIP 0 only)*** Transparent Bridging features conformance to IEEE 802.1d and supports spanning tree protocol and bridge filters

	TCP/IP with RIP version 1(RFC 1058) and version 2(RFC 1723) compatible
	ARP(RFC 1293, supports only one single subnet)
	BOOTP(RFC 2131, RFC 2132)
	SNMP version 1(RFC 1155, RFC 1157, RFC 1213)
	TELNET server(RFC 854, 855, 857, 858)
	NAT server
	TFTP revision 2(RFC 1350)
	PPP (Point-to-Point Protocol) support
	PAP/CHAP user Authentication with PPP
	PPTP tunnelling***
Data rates	Up to 8 Mbps downstream and 640 Kbps upstream. Rate adaptive in 32 Kbps steps
Connect Distance	Up to 18,000 feet
Supported OS	Windows 95, 98, 2000, Me, NT4.0, XP, Mac, Unix & Linux
Power Consumption	6W max through 9V or 12V DC 1000mA power adapter
Product certification	FCC part 15, FCC part 68, and CE marking
Physical Dimension	Approximately 150mm(W) x 135mm(D) x 35mm(H)
Operating Environment	Temperature 0 to 45C ambient Humidity 5% to 95%(non-condensing)

*** Some models does not support

15.1. Power Adapter

The Heritage ADSL modem is powered by a 12V DC 1A power adapter, which included in this package, with positive polarity inside and negative polarity outside. In any case the standard power adapter come with the modem is not available, please find a power adapter meet above specifications.

16. Appendix B Troubleshooting

This chapter is intended to help you troubleshoot problems you may encounter while setting up and using the Modem. It also describes some common hardware and software problems and gives some suggestions to troubleshoot them.

16.1. How to Restore Defaults

To restore factory default settings you must access the Command Line Interface via telnet or console (see above). The CLI command

'config load default'

restores the default settings.

16.2. B.1 Diagnostics with the LEDs

Most hardware problems can be diagnosed and solved by checking the LEDs on the front panel of your router.

- **If the POWER LED is dark**

- Make sure the power cord is firmly plugged into the back panel of the router and the other end into an active AC wall or power strip outlet.

- Make sure the power switch is turned on.

- **If the PC LED is dark**

- Make sure your Ethernet cable is firmly plugged into the back panel of the router and the other end into your computer or HUB.
- Make sure you using the correct Ethernet cable for your application.
- Make sure your Ethernet board is installed properly in your system by ping the IP address of your PC.

16.3. B.2 Problems when configure the Modem via the console port

- **Can't see any message from the configuration screen**

- Make sure the cable connection from the Modem's console port to the computer being used as a console is securely connected.
- Make sure the terminal emulation software is accessing the correct port on the computer that's being used as a console.
- Make sure that flow control on serial connections is turned off.
- Make sure the RS232 device attached to the console is configured as a 'DTE'. If not, a crossover or null modem adapter is required.

- **Junk characters appear on the configuration screen**

- Make sure the terminal emulation software is configured correctly. Check the baud rate and data format is configured to 9600 bps, No parity, 8 data bits, and 1 stop bit.

16.4. B.3 Problems when connecting to the Modem via Ethernet

- **Cannot connect your PC to the Modem for configuration via Ethernet.**

- Make sure the PC LED is light
- Make sure the Modem's IP address matches the IP address previously stored into the Modem's configuration. You must have previously set the Modem's Ethernet IP address and subnet mask, saved the Ethernet configuration changes, and rebooted the Modem for the new IP address to take effect.
- Make sure the PC and the Modem are on the same IP subnetwork or the target router is reachable through a router on your LAN.
- Make sure the TCP/IP properties setting is correct in your PC.
- Make sure if the TX and RX LED on the Modem's front panel blinks when 'pinged'.

16.5. B.4 Problems when accessing the Internet or remote network

- **Cant's access the Internet or remote network**

There are four possibilities to causes this problem

1. The connection between the computer and the Modem
2. The connection between the Modem and your NSP
3. The connection between your NSP and your ISP
4. The connection between your ISP and the Internet

To isolate the problem, you can verify IP connectivity with following steps by running a **ping <IP address>** command. For example, **ping 192.168.254.254**.

1. Ping the IP address of your PC. If you get a response back, proceed to next step directly. If you don't get a response back, check that:
 - The network adapter card is installed.
 - The TCP/IP protocol is installed.

- The TCP/IP protocol is bound to the network adapter.
- 2. Ping the IP address of your Modem. If you get a response back, proceed to next step directly. If you don't get a response back, the problem lies between your PC and your Modem:
 - Check the cables.
 - Check the hub.
 - Make sure that your PC and your Modem belong to the same IP sub network.
 - Observe the TX and RX LEDs to see if data traffic flow appears to be normal
- 3. Ping the DNS server.
 - **If the Modem is configured to bridging mode**
 - Be sure to reboot the Modem if you have made any changes with configuration.
 - All IP addresses must be in the same IP sub network.
 - **If the Modem is configured to routing mode**
 - Check that IP Routing is enabled at the local and the remote end.
 - Make sure the IP addresses of the local and remote networks belong to different IP sub networks.
 - Make sure that there is an existing route to the remote network.
 - Make sure that there is a route back from the remote network.
 - Be sure to reboot the Modem if you have made any changes with configuration.

17. Appendix C Glossary

10Base-T

IEEE 802.3 standard for the use of Ethernet LAN technology over unshielded twisted pair wiring, running at 10Mbps.

ADSL

Asymmetric Digital Subscriber Line - Technology that delivers high-speed data and voice connections over existing phone lines. Up to 8 Mbits/sec can be sent downstream and 640 Kbits/sec upstream.

ANSI (American National Standards Institute)

Devises and proposes recommendations for international communications standards.

ARP

Address Resolution Protocol. An Internet protocol used to bind an IP address to Ethernet/802.3 addresses.

ASCII

American Standard Code for Information Interchange. 8-bit code for character representation.

ATM

Asynchronous Transfer Mode - Cell-relay broadband technology for high-speed transmission of video, audio, data over LAN/WAN, making use of fixed-size cells (53-byte cells).

Bridge

A device that segments network traffic. A bridge maintains a list of each segment's nodes and only traffic destined for a node on the adjacent segment is passed across the bridge. A bridge operates at Layer 2 of the OSI reference model.

CHAP

Challenge Handshake Authentication Protocol. A security protocol supported under Point-to-Point Protocol (PPP) used to prevent unauthorised access to devices and remote networks. Uses encryption of password, device names, and random number generation.

Class A, B, and C networks:

The values assigned to the first few bits in an IP network address determine which class designation the network has. In decimal notation, Class A network addresses range from 1.X.X.X to 126.X.X.X, Class B network addresses range from 128.1.X.X to 191.254.X.X, and Class C addresses range from 192.0.1.X to 223.255.254.X.

Client

An intelligent workstation that makes requests to other computers known as servers. PC computers on a LAN can be clients.

Community strings

Sequences of characters that serve much like passwords for devices using SNMP. Different community strings may be used to allow an SNMP user to gather device information or change device configurations.

Console port

Device used by the network administrator to configure and monitor the Modem. The console port employs an RS232 interface. Command Line Interface are used on the console port.

DHCP

Dynamic Host Configuration Protocol - Service that provides network information (such as IP addresses, masks, domain names) to PCs and other clients automatically.

DNS

Domain Name Service - Transmission Control Protocol/Internet Protocol (TCP/IP) service which translates a name that a person can remember into an IP address that a computer can use.

DTE

Data Terminal Equipment - Term defined by standards committees, that applies to communications equipment, typically personal computers or data terminals, as distinct from other devices that attach to the network, typically modems.

Ethernet address

Sometimes referred to as a hardware address. A 48-bits long number assigned to every Ethernet hardware device. Ethernet addresses are usually expressed as 12-character hexadecimal numbers, where each hexadecimal character (0 through F) represents four binary bits. Do not confuse the Ethernet address of a device with its network address.

Firmware

System software stored in a device's memory that controls the device.

HDLC

High-Level Data Link Control - A generic link-level communications protocol developed by the International Organisation for Standardisation (ISO). HDLC manages synchronous, code-transparent, serial information transfer over a link connection.

Internet

A set of networks connected together by routers. This is a general term, not to be confused with the large, multi-organisational collection of IP networks known as the Internet. An internet is sometimes also known as an internetwork.

Internet address, IP address

Any computing device that uses the Internet Protocol (IP) must be assigned an internet or IP address. This is a 32-bit number assigned by the system administrator, usually written in the form of 4 decimal fields separated by periods, e.g., 192.9.200.1. Part of the internet address is the IP network number (IP network address), and part is the host address (IP host address). All machines on a given IP network use the same IP network number, and each machine has a unique IP host address. The system administrator sets the subnet mask to specify how much of the address is network number and how much is host address.

IP

Internet Protocol - A networking protocol developed for use on computer systems that use the UNIX operating system. Often used with Ethernet cabling systems. In this manual, IP is used as an umbrella term to cover all packets and networking operations that include the use of the Internet Protocol. See also *TCP/IP*.

ISP

Internet service provider - A company that provides Internet-related services. Most importantly, an ISP provides Internet access services and products to other companies and consumers.

ITU

International Telecommunication Union - United Nations specialised agency for telecommunications

LAN

Local area network - A privately owned network that offers high-speed communications channels to connect information processing equipment in a limited geographic area. (usually within a single campus or building).

LED

Light Emitting Diodes - Type of indicator lights on the panel of the router.

MAC layer/address

Media Access Control layer/address defined by the IEEE 802.3 specification which defines media access including framing and error detection. Part of the OSI reference model data link layer.

MIB

Management information base - A standardised structure for SNMP management information.

NAT

Network Address Translation - A feature that allows communication between the LAN connected to the Modem and the Internet using a single IP address, instead of having a separate IP address for each computer on the network.

NSP

Network Service Provider - Company from which you buy your network services.

PAP

PPP Authentication Protocol - A method for ensuring secure network access.

Ping

An echo message, available within the TCP/IP protocol suite, sent to a remote node and returned; used to test the accessibility of the remote node.

Port number

A number that identifies a TCP/IP-based service. Telnet, for example, is identified with TCP port 23.

Protocol

A set of rules for communication, sometimes made up of several smaller sets of rules also called protocols.

PPP

Point-to-Point Protocol - A Data Link layer protocol that provides asynchronous and synchronous connectivity between computer/network nodes. It defines how packets of information are exchanged between computers or network nodes connect via a point-to-point connection (as opposed to multipoint or broadcast). Includes standardisation for security and compression negotiation.

PVC

Permanent Virtual Circuit - Dedicated connection between end stations. The PVC is made up of 2 parts: the VPI and the VCI. In a PVC number of 0,32, 0 represents the Virtual Path Identifier (VPI) and 32 represents the Virtual Circuit Identifier (VCI).

RFC 1483

Protocol that encapsulates ATM cells into logical data link frames.

RFC

Request for Comment - A series of documents used to exchange information and standards about the Internet.

RIP

Routing Information Protocol - A protocol used for the transmission of IP routing information.

RJ-11

A telephone-industry standard connector type, usually containing four pins.

RJ-45

A telephone-industry standard connector type, usually containing eight pins.

Routing

A network layer function that determines the path for transmitting packets through a network from source to destination.

Router

A device that supports network communications. A router can connect identical network types, However—unless a gateway is available—a common protocol, such as TCP/IP, must be used over both networks. Routers may be equipped to provide WAN line support to the LAN devices they serve. They may also provide various management and monitoring functions as well as a variety of configuration capabilities.

Routing table

A list of networks maintained by each router on an internet. Information in the routing table helps the router determine the next router to forward packets to.

Serial port

A connector on the back of the workstation through which data flows to and from a serial device.

Server

A device or system that has been specifically configured to provide a service, usually to a group of clients.

Subnet

A network address created by using a subnet mask to specify that a number of bits in an internet address will be used as a subnet number rather than a host address.

Subnet Address

An extension of the Internet 32-bit addressing scheme which allows the separation of physical or logical networks within the single network number. assigned to an organisation. TCP/IP entities outside this organisation have no knowledge of the internal 'subnetting'.

Subnet mask

A 32-bit number to specify which part of an internet address is the network number, and which part is the host address. When written in binary notation, each bit written as 1 corresponds to 1 bit of network address information. One subnet mask applies to all IP devices on an individual IP network.

RS-232

EIA standard specifying the physical layer interface used to connect a device to communications media.

SNMP

Simple Network Management Protocol - A widely implemented Internet network management protocol that allows status monitoring, getting/setting of parameters for configuration and control of network devices, such as routers and bridges.

TCP/IP

Transmission Control Protocol/Internet Protocol - An open network standard that defines how devices from different manufacturers communicate with each other over one or more interconnected networks. TCP/IP protocols are the foundation of the Internet, a world-wide network of networks connecting businesses, governments, researchers, and educators. TCP provides a connection-oriented transport layer ensuring end-to-end reliability in data transmission. IP provides for network layer connectivity using connectionless datagrams.

TFTP

Trivial File Transfer Protocol - A protocol used to transfer files between IP nodes. TFTP is often used to transfer firmware and configuration information from a UNIX computer acting as a TFTP server to an IP networking device.

TELNET

Internet standard protocol for remote terminal emulation that allows a user to remotely log in to another device and appear as if directly connected.

Transparent Bridging

Bridging technique used in Ethernet networks which allows transfer of frames across intermediate nodes using tables associating end nodes with bridging addresses. Bridges are unknown to the end nodes.

VCI

Virtual Channel Identifier - Number that identifies a channel within a virtual path in a ADSL/ATM environment.

Virtual Channel

Refers to a logical connection between end stations in an ADSL/ATM environment

Virtual Path

Refers to a bundle of virtual channels in a ADSL/ATM environment.

VPI

Virtual Path Identifier - Number that identifies the link formed by the virtual path in a ADSL/ATM environment.

UDP

User Datagram Protocol - A TCP/IP protocol describing how packets reach applications in destination nodes.

Wall jack

A small hardware component used to tap into telephone wall cable. An RJ-11 wall jack usually has four pins; an RJ-45 wall jack usually has eight pins.

WAN

Wide Area Network - A network that consists of nodes connected by long-distance transmission media, such as telephone lines. WANs can span a state, a country, or even the world.

18. Appendix D Government compliance notices

European CTR 21 compliance

The equipment has been approved in accordance with Council Decision 98/482/EC for pan-European single terminal connection to the public switched telephone network (PSTN). However, due to differences between the individual PSTNs provided in different countries, the approval does not, of itself, give an unconditional assurance of successful operation on every PSTN network termination point. In the event of problem, you should contact your equipment supplier in the first instance.

Note: The manufacturer should ensure that the vendor and user of the equipment is clearly informed of the above information by means of package and/or user manuals of the forms of user instructions.