Solwise Ltd.

Extra Set-up instructions for

# Solwise SAR110/130 ADSL Routers Running Vr2.1 Firmware

www.solwiseforum.co.uk

The Solwise Forum is designed to be the first port-of-call for technical support and sales advice for the whole Solwise product range.

Please check the forum for coverage on any technical problems you have. Many people have trodden your path before you, and a quick check on the forum will reduce the pressure on our support staff.

March 30, 2004
Covers Solwise-260304/2.1

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
Support Tel: 0845 1931320
SBV: 1100
Email sales@solwise.co.uk
Http www.solwise.co.uk

### Copyright

Changes are periodically made to the information in this document. They will be incorporated in subsequent editions. The product manufacturer may take improvement and/or changes in the product described in this document at any time.

## FCC compliance

This equipment complies with Part 68 of the FCC Rules. On this equipment is a label that contains, among other information, the FCC registration number and Ringer Equivalence Number (REN) for this equipment. You must, upon request, provide this information to your telephone company.

If your telephone equipment causes harm to the telephone network, the Telephone Company may discontinue your service temporarily. If possible, they will notify in advance. But, if advance notice isn't practical, you will be notified as soon as possible. You will be informed of your right to file a complaint with the FCC.

Your telephone company may make changes in its facilities, equipment, operations, or procedures that could affect proper operation of your equipment. If they do, you will be notified in advance to give you an opportunity to maintain uninterrupted telephone service. The FCC prohibits this equipment to be connected to party lines or coin-telephone service.

In the event that this equipment should fail to operate properly, disconnect the equipment from the phone line to determine if it is causing the problem. If the problem is with the equipment, discontinue use and contact your dealer or vendor.

## DOC compliance information

NOTICE: The Canadian Department of Communications label identifies certified equipment. This certification means that the equipment meets certain telecommunications network protective, operational and safety requirements. The Department does not guarantee the equipment will operate to the user's satisfaction.

Before installing this equipment, users ensure that it is permissible to be connected to the facilities of the local Telecommunications Company. The equipment must also be installed using an acceptable method of connection. The customer should be aware that compliance with the above conditions might not prevent degradation of service in some situations.

Repairs to certified equipment should be made by an authorized Canadian maintenance facility designated by the supplier. Any repairs or alterations made by the user to this equipment, or equipment malfunctions, may give the telecommunications company cause to request the user to disconnect the equipment.

Users should ensure for their own protection that the electrical ground connections of the power utility, telephone lines and internal metallic water pipe system, if present, are connected together. This precaution may be particularly important in rural areas.

**CAUTION:** Users should not attempt to make such connections themselves, but should contact the appropriate electric inspection authority, or electrician, as appropriate.

**NOTICE:** The Load Number (LN) assigned to each terminal device denotes the percentage of the total load to be connected to a telephone loop which is used by the device, to prevent overloading. The termination on a loop may consist of any combination of devices subject only to the requirement that the sum of the Load Numbers of all the devices does not exceed 100.

## 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.

# Table of Contents

# 1 CLI Scripting

## 1.1 Overview - CLI Scripts

CLI provides you with scripting capability. This enables you to write your own CLI scripts and execute those scripts on the command line. CLI recognizes these script files by their file extensions, which have to be either ".cfg" or ".sh".

## 1.2 .cfg scripts

The ".cfg" scripts support sequential execution of CLI commands, where, each script statement must contain a single CLI command. The following example shows a typical "sample.cfg" script file.

**Sample .cfg file**

```
get system
get atm port ifname atm-0
get atm vc intf ifname aal5-0
```

The CLI commands are executed in sequence.

**NOTE:**

You cannot put comments in a ".cfg" script. Unlike ".sh" scripts, these scripts must contain only CLI commands.

## 1.3 .sh scripts

The ".sh" scripts provide enhanced scripting features as discussed later in this chapter. They are backward compatible with ".cfg" scripts. These scripts enable you to execute the commands conditionally rather than sequentially. They also help you perform multiple tasks, with the execution of the same file.

**WARNING:**

Never use the following CLI commands in **.cfg** and **.sh** scripts: **apply, download, reboot, exit, alias, unalias, prompt, verbose**.

## 1.4 Executing CLI scripts

You can execute the script on the command line using the "apply" command. For example, if you want to execute the "sample.sh" (or "sample.cfg") script, you should give the following command on the command line.

- To execute "sample.sh" script, enter:

```
$ apply fname sample.sh
```

**NOTE:**

Please refer to the CLI Manual for

details on the “apply” command.

**NOTE:**

The “sample.sh” (or “sample.cfg”) script file must be in the file system for it to execute at all. You may use the **ftp** or **download** commands for transferring the file to the file system.

You can also pass input parameters to the “.sh” script using the same command. Let us say, “sample.sh” script takes three input parameters - param1, param2 and param3. The following command shall pass on the parameters to the script.

## 1.5 Shell script programming

- To pass on the parameters to the script, enter:

```
$ apply fname sample.sh sparams “param1 param2 param3”
```

**NOTE:**

Input parameters must be separated by white spaces and must be inside “”.

- In the script, you can access these parameters using mnemonics $1, $2 and $3. Please refer to the example below:

*Example* Assume you want to write a script, which modifies the VPI/VCI value of a specific VCC. This script takes three parameters - VCC name, VPI and VCI.

- To execute the script, enter:

```
$ apply fname modvcc.sh sparams “aal5-0 0 35"
```

The script file “modvcc.sh” contains the following code

```
# This script modifies the VPI/VCI value of a given VCC.
# $1, $2 and $3 are replaced by aal5-0, 0 and 35 respectively.
modify atm vc intf ifname $1 vpi $2 vci $3
```

## 1.6 Shell script programming

The scripting feature allows you to achieve much more than parameter substitution discussed in the previous section. This feature provides basic programming constructs such as assignment statements, comparison statements, regular expression based search statements, Goto statements and return statements. These constructs enable you to build logic into your scripts and perform functions intelligently.

**NOTE:**

These are features of “.sh” scripts only.

### 1.6.1 Assignment statement

This is an example of a basic assignment statement, categorized as an explicit statement, in which the return variable is defined explicitly.

```
  Explicit assignment statement
```

- To store the return status of a CLI command using an assignment statement, enter:

```
retval = create atm port ifname atm-0
```

In the example given above, “retval” stores the return status of the “create atm port” command. You can access the value stored in “retval” variable using the “$” mnemonic.

**Implicit assignment statement**

Given below, is an example of an implicit assignment statement.

```
create atm port ifname atm-0
```

In this example, the return variable is missing. But you can still access the value of the return status of this command using “$0”. “$0” stores the return status of the last executed CLI command.

### 1.6.2 Comparison statement

Comparison statements enable you to execute script statements conditionally. Using these statements, you can execute statements based on whether the comparison was successful or not. The example below, contains nested comparison statements.

```
if $1 eq create
create atm vc intf ifname aal5-0 vpi 0 vci 35
else
if $1 eq modify
modify atm vc intf ifname aal5-0 vpi 0 vci 35
fi
fi
```

### 1.6.3 Regular expression statements

This statement enables you to perform regular expression-based searches on the value of a script variable. It is most useful in scenarios where you want to look for a specific pattern in the output of a CLI command.

- To read the output of the last executed CLI command into a variable, use the following statement:

```
# retval variable stores the return status of “get system”
command
retval = get system
# system_info variable stores the output of “get
system” command
> system_info
```

- To search for a specific pattern in the “system_info” variable, use the following statement:

```
# system_name variable stores the pattern, which matches the
regular expression
system_name = search $system_info 'Name[ ]*:[ ]*[a-zA-Z]+'
```

This statement stores the pattern, which matches the regular expression “Name[ ]*:[ ]*[a-zA-Z]+” in “system_name” variable. The regular expression must always be enclosed in ' '. There is a statement, which returns the string after “:” in a pattern.

- In the continuing example, assume you are interested in the system name, which happens to be after “:”. The “name” variable stores the system name after the execution of this statement.

```
# name variable stores the value after ":" in the pattern
name = findval $system_name
```

**NOTE:**

In most scenarios, the three statements discussed in this section, are expected to be executed in sequence.

### 1.6.4 Goto statement

- To Goto to any statement in the script, use this statement:

```
If $1 eq xyz
…
goto label1
else
…
goto label2
fi
label1:
return 0
label2:
return 1
```

This example returns different values, which is achieved using Goto statements.

**NOTE:**

Each label should be terminated with ":" , should not contain any white space characters and should be the only keyword in that line.

### 1.6.5 Return statement

- To return from the script, use the following statement:

```
if $1 eq create
create atm vc intf ifname aal5-0 vpi 0 vci 35
return 0
fi
```

Currently, the return statement only serves the purpose of returning from the script. The value returned is of no consequence and is ignored. Therefore, it is preferable that you consider this statement as a mechanism for returning from the script. Using this statement at the end of the script is optional.

*Example - Scripting* The example here, will help you achieve something useful, using the scripting feature.

Assume you want to write a script, which creates a PPP interface over a VCC, only if the VCC creation is successful. The CLI command you will need to use is:

```
$ apply fname creppp.sh sparams "ppp-1 aal5-0 0 35"
```

The script file "creppp.sh" contains the following code:

```
# Create the VCC
retval = create atm vc intf ifname $2 vpi $3 vci $4
#Create the PPP interface only if VCC creation was successful.
# retval stores the return status of the VCC creation command.
Check this value # against 0, which indicates
success and only if the comparison is successful, the # PPP
interface should be created.
if $retval eq 0
create ppp intf ifname $1 lowif $2
```

```
fi
```

**NOTE:**

Please refer to the Appendix of this document for details on all the programming constructs.

### 1.6.6 Script programming rules

- Each statement must be terminated by a new line. Two statements cannot be on the same line.

```
# wrong script
a = get ethernet intf ifname eth-0 b = get system
# correct script
a = get ethernet intf ifname eth-0
b = get system
```

- Each keyword of the statement must be separated by white spaces.

```
# wrong script
retval=get system
# correct script
retval = get system
```

- Literal strings used in the script must not contain any white spaces and must not be included in “ “.

```
# wrong script
if $1 eq hello world
…
fi
if $1 eq "hello world"
…
```

**Script programming rules**

```
fi
# correct script
if $1 eq helloworld
…
fi
if $1 eq helloworld
…
fi
```

- Each “if” keyword must have a matching “fi” keyword.

```
# wrong script
if $1 eq hello
…
else if $1 eq bye
…
fi
# correct script
if $1 eq hello
…
else
if $1 eq bye
…
fi
fi
```

- Only single line comments are supported.

```
# wrong comment
This is not how to write comments
```

```
#
# correct comment
# This is how to write comments
#
```

- Only those parameters are substituted, which are white space separated.

```
# no parameter substitution will happen in this case
retval = modify system name ‶$1″
# parameter substitution will happen in this case, # as there
is white space after $1
retval = modify system name ‶ $1 ‶
```

# 2 Interfaces and Operating Modes

This chapter briefly discusses the unit's interfaces, and explains how to create and configure the interfaces needed for the bridge and router operating modes, as well as how to select each mode.

## 2.1 Default Interface and Mode Configuration on the SAR130 Reference Board

By default, the SAR130 is configured as a router, with the following interfaces

• Operating mode: Router
• Ethernet (LAN) interfaces
- LAN port: eth-0
· IP address 192.168.7.1, subnet mask 255.255.255.0
- Virtual Ethernet interfaces: None
• WAN interfaces
- ATM port: atm-0
· Maximum number of VCs allowed: 8
- VC: aal5-0
· Lower interface atm-0, VPI 1, VCI 38
- PPPoA interface: ppp-0
· Lower interface aal5-0, CHAP authentication
· User name 'guest', password 'guest' (for login with ISP)
· Default route

The following commands are included in the default configuration file to set this configuration:

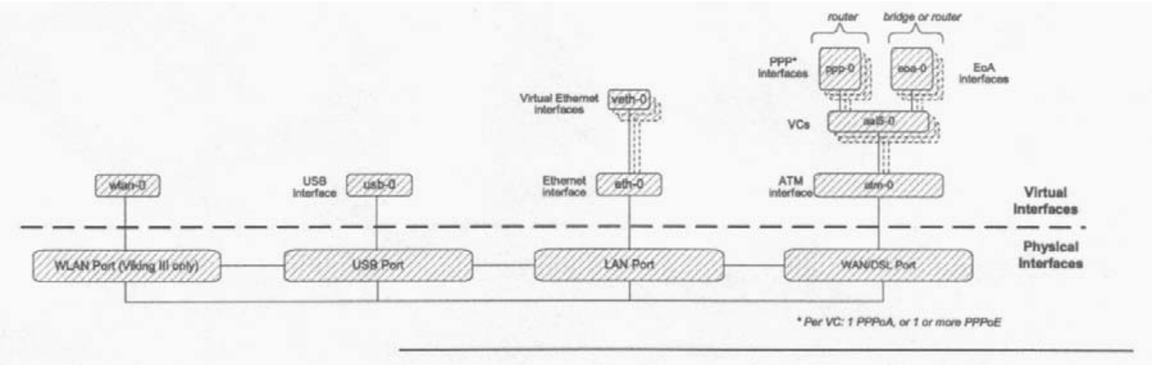
```
create atm port ifname atm-0 maxvc 8

create atm trfdesc trfindex 0

create atm vc intf ifname aal5-0 lowif atm-0 vpi 0 vci 38 vcmux

create ppp security ifname ppp-0 CHAP login guest passwd guest

create ppp intf ifname ppp-0 start lowif aal5-0 droute true PPOA usedhcp
false

create ethernet intf ifname eth-0 ip 192.168.7.1 mask 255.255.255.0
```

The first three lines create the ATM port atm-0, the VC aal5-0, and the PPPoA interface ppp-0. The subsequent command creates the ethernet port.

## 2.2 Interfaces – Overview

At the physical level, the unit provides WAN-LAN connectivity through its physical WAN and LAN ports. At the logical level, the connection can be made in a number of ways, depending on the virtual interfaces configured on top of the physical ports and how these interfaces are connected.

Below shows the virtual interfaces you can define on each physical port.



**Ports and Interfaces**

In order to create an interface, you first create all the interfaces below it, starting at the lowest interface. For instance, to create a PPP interface, you first create the ATM port, then a VC.

## 2.3 Configuring the Ethernet Port

The Ethernet port is a physical port on that enables you to connect the unit to a computer or Ethernet network. You can configure only one physical Ethernet port, *eth-0*; however, you can define multiple virtual ethernet interfaces over this port. This port can be created with or without an IP address (no IP address is required if it is a bridge port).

When creating the Ethernet port, you may need to consider the following:

• IP address and subnet – To connect the unit to an existing LAN whose subnet differs from the Ethernet port's default subnet (192.168.1.1, mask 255.255.255.0), assign the Ethernet port an IP address in the same subnet as your LAN. (Alternatively, you would have to assign to each LAN computer a new IP address and mask that places it in the same subnet as the Ethernet port.)

Commands related to the Ethernet port are briefly described below.

**NOTE:**

For a complete listing of these commands, including parameters and default values, refer to the CLI Manual.

**Creating the Ethernet port**

- To create the Ethernet port *eth-0*, enter:

```
$ create ethernet intf ifname eth-0 ip 192.168.1.1 mask
255.255.255.0
```

- To display information on the Ethernet port, enter:

```
$ get ethernet intf
```

### Setting Interface security type

You can set the interface security type to either pvt, pub, dmz, while creating the Ethernet interface.

```
$ create ethernet intf ifname eth-0 ip 192.168.1.1 mask
255.255.255.0 ifsectype private
```

### Changing the Ethernet port’s IP address

- To change the Ethernet port’s IP address to 10.1.1.1 with mask 255.0.0.0, enter:

```
$ modify ethernet intf ifname eth-0 ip 10.1.1.1 mask 255.0.0.0
```

If you are connecting the unit to an existing LAN, and if the Ethernet port’s default subnet—IP address 192.168.1.1, mask 255.255.255.0—is different from the LAN’s subnet, change the Ethernet port’s IP address, as follows:

**1.** Set any LAN host’s IP address to 192.168.1.3, mask 255.255.255.0.

**2.** Using this host, Telnet to 192.168.1.1 and log in to the system.

**3.** Enter the `modify ethernet intf` command (described above) to change the IP address and/or mask of the eth-0 interface.

**4.** Enter `commit` to save the changes.

**5.** Change the host’s IP address and/or mask to the original value(s).

**6.** Reboot the host.

If you are connecting the unit to a new LAN, i.e., one whose subnet is not yet determined, you do not need to change the Ethernet port’s IP address. Instead, assign each LAN host an IP address from the Ethernet port’s default subnet, i.e., 192.168.7.2, 192.168.7.3, etc. Or, configure each PC as a DHCP client so that it will be assigned an appropriate address from the unit’s default DHCP pool (assuming that this pool has been configured).

### Using a LAN DHCP server to assign the port’s IP address

- To reconfigure the unit to get its LAN IP address from a DHCP server running on a LAN host, enter:

```
$ modify ethernet intf ifname eth-0 ip 0.0.0.0 mask 0.0.0.0
usedhcp remote
```

Both the IP address and mask must be set to 0.0.0.0. Setting **usedhcp** to **remote** (default=**false**) invokes a DHCP client to obtain an IP address for this interface from a DHCP server.

The `get ethernet intf` command will show the IP address as *0.0.0.0* , while the `get ip address` command will show the address obtained from the dhcp server.

**NOTE:**

*If you are changing the IP address of the Etherne*t address over a telnet or HTTP connection, the connection will be lost once the address is modified.

You can also configure ethernet to get its address from the unit's own DHCP server. To do so, set **usedhcp** to **local**, using the ommand, `modify ethernet intf ifname eth-0 usedhcp local` The **usedhcp local** option assumes that the DHCP server is enabled and a DHCP pool is created. If the wireless LAN interface has also been created using the same option, then it is necessary to have two DHCP pools configured, and these must belong to different subnets. If one pool has been used by one interface, the unit automatically chooses the other pool for the other interface.

### Displaying the Ethernet port's *IP address*

- To see the current configuration of the Ethernet interface, enter:

```
$ get ethernet intf ifname eth-0
```

If the displayed IP address is 0.0.0.0, the unit has been configured to get its LAN IP address from a LAN DHCP server. To see the actual IP address, use the `get ip address` command.

- To see the IP address obtained from a DHCP server (plus the IP addresses for all configured IP-enabled interfaces), enter:

```
$ get ip address
```

### Deleting an Ethernet interface

- To delete an Ethernet interface, enter:

```
$ delete ethernet intf ifname eth-0
```

### Configuring the interface MTU

- To change the MTU for the interface to, say, 1400 enter:

```
modify ethernet intf ifname eth-0 mtu 1400
```

## 2.4 Configuring Virtual Ethernet Interfaces

Virtual Ethernet interfaces give the impression of multiple subnets on a single physical subnet, by dividing your LAN hosts into groups, each with its own subnet mask. You can up to two virtual Ethernet interfaces, named veth-0 and veth-1, over the single physical Ethernet interface.

- To create a virtual interface, enter:

```
$ create ethernet intf ifname veth-0 ip 172.25.1.1 mask
255.255.255.0 phyif eth-0
```

The `phyif` parameter indicates that the virtual interface *veth-0* actually sits on the physical interface *eth-0*. Unlike the physical Ethernet interface, the virtual Ethernet interfaces can be deleted using the `delete ethernet intf` command.

- To list the virtual Ethernet interfaces (as well as physical Ethernet interfaces), enter:

```
$ get ethernet intf
```

## 2.5 Configuring the WAN ATM Port

Data traffic is carried over the DSL cable in ATM cells. To enable the DSL port (i.e., the WAN port) to carry ATM cells, you need to configure an ATM port on the unit. You can configure only one ATM port, *atm-0*.

When creating the ATM port, consider the following:

• ATM priority scheduling – The relative priorities of the ATM service categories (described in Section 5.9.2). By default, the priorities are in this order - CBR, RTVBR, NRTVBR, GFR, UBR.

Commands related to creating the ATM port are briefly described below.

**NOTE:**

For a complete listing of these commands, including parameters and default values, refer to the CLI Manual.

Creating the ATM port • To create the ATM port atm-0, enter:

```
$ create atm port ifname atm-0
```

• To display information on the ATM port, enter:

```
$ get atm port
```

**Setting ATM service category priorities**

The `create atm port` command is also used to assign relative priorities to ATM service categories.

• To give the UBR service category priority over GFR (GFR has higher priority by default), enter:

```
$ create atm port ifname atm-0 ubrpriority 2 gfrpriority 1
nrtvbrpriority 3 rtvbrpriority 4 cbrpriority 5
```

## 2.6 Configuring Permanent Virtual Circuits

Virtual Circuits (VCs), named *aal5-0*, *aal5-1*, etc., sit on top of the ATM port. Each VC has an associated Virtual Path Identifier (VPI) and Virtual Circuit Identifier (VCI) that identify a data path through the ATM network. Besides the VPI and VCI, you should also consider the following when creating a VC:

• AAL5 data encapsulation – VC-muxing, LLC-muxing (default), or none.

• Service category – Unspecified Bit Rate (UBR) (default) or Guaranteed Frame Rate (GFR), Non Real-Time Variable Bit Rate (NRTVBR), Real- Time Variable Bit Rate (RTVBR), or Constant Bit Rate (CBR). - A UBR traffic descriptor usually exists as part of the default configuration. So a UBR VC can be created right away. For any other type of VC - GFR, NRTVBR, RTVBR, or CBR, you must also create a traffic descriptor of the same category if you have not yet done so.

• Priority – The relative transmission priority of the VC vs. other VCs in the same service category.

The commands used to create VCs statically are briefly described below.

**NOTE:**

For a complete listing of these commands, including parameters and default values, refer to the CLI Manual.

*Creating a VC* • To create a VC-muxed VC named aal5-0 with VPI 0 and VCI 35, enter:

```
$ create atm vc intf ifname aal5-0 vpi 0 vci 35 vcmux lowif
atm-0
```

This creates VC aal5-0, with VPI 0 and VCI 35, on top of ATM port atm-0. Since the default values for all other parameters are used, the traffic descriptor (described in Section 5.9.2.3) is 0, and thus the ATM service category is UBR.

**NOTE:**

The number of VCs you can create is limited by the **`maxvc`** parameter in the **`create atm port`** command and the **`maxvc`** and **`max1483vc`** parameters in the **`size`** command (described in section 4.1.1). All three parameters are typically set to the same value.

• To see a list of all currently configured VCs, enter:

```
$ get atm vc intf
```

### 2.6.1 AAL5 Data Encapsulation Method

The unit supports two data encapsulation methods: *VC mux* and *LLC mux*. Each allows you to create different types of interfaces on the VC. A third mode with no encapsulation is also supported.

**VC-muxed VC.** The allowed interfaces are:

- EoA
- PPPoA
- IPoA
- EoA + PPPoE
- EoA + bridge port over EoA
- EoA + bridge port over EoA + PPPoE

**LLC-muxed VC.** The allowed interfaces are:

- EoA 1
- PPPoE 1
- PPPoA 1
- IPoA1
- EoA + PPPoE1
- EoA + PPPoE + PPPoA 2
- EoA + PPPoE + IPoA 2
- PPPoA + IPoA 2
- EoA + IPoA 2
- EoA + bridge port over EoA 2

- EoA + bridge port over EoA + PPPoE 2
- EoA + PPPoE + PPPoA + bridge port over EoA 2
- EoA + bridge port over EoA + PPPoE + IPoA 3
- EoA + PPPoA + IPoA 3
- EoA + PPPoE + PPPoA + IPoA 3

### 2.6.2 ATM Service Categories: UBR, CBR, GFR, NRTVBR and RTVBR

Every VC has an associated ATM *service category*. The following service categories can be defined, based on the Quality of Service (QoS) provided:

- *Unspecified Bit Rate (UBR)* – ATM provides no rate guarantee; data is transmitted on the VC only as and when bandwidth is available.

- *Guaranteed Frame Rate (GFR)* – ATM guarantees a minimum bandwidth, called the *Minimum Cell Rate (MCR)*, for the VC. Depending on available bandwidth, GFR also provides a maximum bandwidth, called the *Peak Cell Rate (PCR)*.

1. if the a5maxproto parameter in create atm vc command is >= 1
2. if the a5maxproto parameter in create atm vc command is >= 2
3. if the a5maxproto parameter in create atm vc command is >= 3

- Non Real-Time Variable Bit Rate (NRTVBR) - ATM guarantees a Sustained Cell Rate (SCR) and allows the user to go up to a Peak Cell Rate (PCR) for a duration derived from the Maximum Burst Size (MBS).

This category is used by non-real time applications.

- Real-Time Variable Bit Rate (RTVBR) - ATM guarantees a Sustained Cell Rate (SCR) and allows the user to go up to a Peak Cell Rate (PCR) for a duration derived from the Maximum Burst Size (MBS). This category is used by real time applications like voice and video.

- Constant Bit Rate (CBR) - ATM guarantees bandwidth up to a Peak Cell Rate (PCR).

You specify a VC's service category when you create the VC, using a traffic descriptor.

#### 2.6.2.1 UBR, GFR, and CBR, NRTVBR and RTVBR Transmission Priorities

Each service category's transmission priority can be set using the **`create atm port`** command's **`ubrpriority`**, **`gfrpriority`**, **`nrtvbrpriority`**, **`rtvbrpriority`**, and **`cbrpriority`** parameters. The three parameters must have different values (by default, **`cbrpriority`** is 5 (highest), **`rtvbrpriority is 4, nrtvbrpriority is 3, gfrpriority`** is 2, and **`ubrpriority`** is 1).

When creating the ATM port, the relative priorities of the ATM service categories are, by default: CBR, RTVBR, NRTVBR, GFR, UBR.

#### 2.6.2.2 Transmission Priorities of VCs

You can also assign relative priorities to the VCs within each service category, using the **vcweight** parameter in the **create atm vc intf** command (for details, refer to the *CLI Manual*). The *Weighted Fair Queuing (WFQ)* algorithm is used to ensure fair and efficient bandwidth allocation for both service categories.

#### 2.6.2.3 Traffic Descriptors

A VC's service category is assigned indirectly, using a *traffic descriptor*. A traffic descriptor defines a set of ATM traffic-related properties, the most important property being the service category, i.e., UBR, GFR, , NRTVBR, RTVBR or CBR.

When you create a VC using the **create atm vc intf** command, you define its service category using the **trfdesc** parameter. The default value of this parameter is 0, corresponding to the default traffic descriptor.

The default configuration provides an initial traffic descriptor with index 0. This *default traffic descriptor* specifies the UBR service category.

To create a UBR VC, omit the **trfdesc** parameter when creating the VC. To create a GFR, NRTVBR, VBR or CBR VC, you must create a traffic descriptor of the same category.

**Creating a GFR traffic descriptor**

- To create traffic descriptor 1, for GFR VCs with MCR=50 and PCR=150:

```
$ create atm trfdesc trfindx 1 GFR CLP_NOTAG_MCR mcr 50 pcr 150
```

The **CLP_NOTAG_MCR** flag indicates that if PCR is exceeded, the VC will drop extra cells without tagging the Cell Loss Priority (CLP) bit.

- To create a VC using the preceding traffic descriptor:

```
$ create atm vc intf ifname aal5-0 trfindx 2 vpi 5 vci 50 lowif
atm-0
```

**Creating a VBR Traffic Descriptor**

- To create traffic descriptor 3, for RTVBR VCs with PCR=150, SCR=75 and MBS=15:

```
$ create atm trfdesc trfindx 3 RTVBR NOCLP_SCR pcr 150 scr 75
mbs 15
```

The NOCLP_SCR flag indicates that the traffic parameters are valid for the aggregate flow and that an SCR is required.

- To create a VC using the preceding traffic descriptor:

```
$ create atm vc intf ifname aal5-2 trfindx 3 vpi 5 vci 52 lowif
atm-0
```

**Creating a CBR Traffic Descriptor**

- To create traffic descriptor 2, for CBR VCs with PCR=150:

```
$ create atm trfdesc trfindx 2 CBR NOCLP_NOSCR pcr 150
```

The NOCLP_NOSCR flag indicates that the traffic parameters are valid for the aggregate flow and that no Sustained Cell Rate is required.

- To create a VC using the preceding traffic descriptor:

```
$ create atm vc intf ifname aal5-1 trfindx 2 vpi 5 vci 51 lowif
atm-0
```

- To display all currently defined traffic descriptors, enter:

```
$ get atm trfdesc
```

```
  Creating a RTVBR traffic descriptor:
```

- To create traffic descriptor 3, for RTVBR VCs with PCR=150, SCR=75 and MBS=15:

```
$ create atm trfdesc trfindx 3 RTVBR NOCLP_SCR pcr 150 scr 75
mbs 15
```

The NOCLP_SCR flag indicates that the traffic parameters are valid for the aggregate flow and that an SCR is required.

- To create a VC using the preceding traffic descriptor:

```
$ create atm vc intf ifname aal5-2 trfindx 3 vpi 5 vci 52 lowif
atm-0
```

## 2.7 Configuring Switched Virtual Circuits (SVCs)

The modem supports Switched Virtual Circuits (SVCs) created through UNI version 3.1 or 4.0 signalling. To create an SVC, first create a signaling channel for UNI. This is simply a PVC which usually has the VPI = 0 and VCI = 5.

```
  Create PVC for UNI signaling

$ create atm vc intf ifname aal5-0 vpi 0 vci 5 none
```

Here, **none** specifies the encapsulation as **none**.

- To configure UNI signaling to run on this VC, give the following command:

```
  Configuring UNI

$ create atm uni ifname aal5-0 nplan atmes saddr
0x47000580ffde0000000000105000000000000000 version uni40
```

The parameter **saddr** is the ATM address of the modem, while **nplan** specifies this address to be an ATM End System type of address. With ATMES, the address must be specified as a string of hex bytes. Conversely, the **nplan** could be specified as ISDN, in which case the address should be given as a string of decimal digits. The **version** parameter specifies the UNI signaling version, which here, is 4.0. The default version is 3.1.

Signaling ATM Adaptation Layer (SAAL) is a layer in the SVC signaling stack that provides reliable transfer of signaling messages between peer UNI entities. If the signaling channel with the remote host is established, the SAAL status is set to UP, and the following trap is generated.

STATUS ALARM : SAAL UP

Otherwise, the SAAL status is DOWN. SAAL may come up later when the signaling channel gets established with the remote host.

The following trap is generated when SAAL goes down:

STATUS ALARM : SAAL DOWN

• You can check the SAAL status at any time, using the command:

get atm uni ifname aal5-0

• With UNI configured, you can now initiate the creation of an SVC by giving the following command:

**Creating an SVC**

```
$ create atm svccfg ifname aal5-1 nplan atmes daddr
0x39000760ff890000000000119000000000000000
```

This tells the modem to establish an SVC with the host having the ATMES address specified by **`daddr`**. The **`ifname`** parameter indicates that the created SVC should be identified by the name *aal5-1*. Other parameters in the command (assumed default here) specify what characteristics you want for the SVC: the traffic descriptor, multiplexing type and so on, as with a PVC. After the command is executed, establishing the SVC with the remote host depends on the Signaling ATM Adaptation Layer (SAAL) status.

If SAAL status is **UP** the modem negotiates SVC parameters with the remote host by exchanging signaling messages. Once the VC is established the following trap is generated:

STATUS ALARM : ATM VC Up : Interface - aal5-1, PortId = 7, Vpi

= 0, Vci = 33

This indicates that the negotiated SVC has the VPI = 0 and VCI = 33 and has been created with the interface name *aal5-1* on the modem. Giving the get atm vc intf command will now show this new VC as well. The allocated VPI and VCI values can also be seen using the get atm svccfg command.

If SAAL status is DOWN, the modem does not exchange signaling messages with the remote host. So, SVC is not established at this point in time. In future, whenever SAAL comes up, the SVC gets established on its own.

To check out, at any time, if an SVC is established or not, its VPI and VCI value should be checked by issuing the “get atm svccfg” command. If it is not established, then, you see the printed value as “-” . Otherwise, the valid numerical value is printed.

**NOTE:**

All SVCs are disconnected when SAAL goes down. So, VPI and VCI value become unassigned for these VCs. Whenever SAAL comes up, the SVCs get established on their own.

• To delete an SVC, use the **`delete atm svccfg`** command.

**NOTE:**

SVC configuration can be specified in the **tefacs.txt** file (default configuration). Also, SVC configuration is committed when the `commit` command is invoked. SVC configuration is retained across boots.

### Starting and Stopping an SVC

You can force SVC establishment or disconnection using the **start** and **stop** commands, discussed below.

- To start/stop an SVC by exchanging appropriate signaling messages with the network side, enter:

```
modify atm svccfg ifname aal5-1 start
```

```
modify atm svccfg ifname aal5-1 stop
```

`Start` is particularly useful when an SVC is disconnected by the network side. If an upper layer protocol such as PPPOE is bound over this VC, and you want to re-establish the SVC, you can do so using the `start` command, without any configuration overheads. If you specify `start` command for an already established SVC, or a `stop` command for an already disconnected SVC, it is ignored.

The trap message “ATM VC Up” displays after the SVC is established. The trap message “ATM VC down” displays when the SVC is disconnected.

### Deleting an SVC

- To delete an SVC, enter

```
delete atm svccfg ifname aal5-1
```

**NOTE:**

SVC deletion fails if an upper layer, such as PPPoE, is bound over the VC.

To verify SVC deletion, use the `get atm svccfg` command. It should not show an entry corresponding to the specified interface name.

Deleting UNI • To delete a configured UNI signaling channel, enter:

```
delete atm uni ifname aal5-0
```

To verify UNI deletion, use the `get atm uni ifname aal5-0` command. It should not show any entry corresponding to the specified interface name.

### Deleting PVC for UNI signaling

- To delete the PVC for UNI signaling, enter:

```
delete atm vc intf ifname aal5-0
```

## 2.8 Configuring PPP Interfaces

The unit supports two types of PPP interfaces—PPPoA and PPPoE. For authentication, both Password Authentication Protocol (PAP) and Challenge Handshake Authentication Protocol (CHAP) are supported. Each PPP interface is IP-enabled, i.e., it has an associated IP address. You may specify this IP address in the **`create ppp intf`** command, if the address is allocated statically by the ISP. If the IP address is obtained dynamically using IPCP, do not specify it as part of the command.

PPP interfaces are named *ppp-0*, *ppp-1*, etc. To create a PPP interface:

**1.** Create a login name and password for the PPP interface.

**2.** Create the PPPoE or PPPoA interface itself.

**NOTE:**

For a complete listing of these commands, including parameters and default values, refer to the CLI Manual.

### 2.8.1 Creating a Login Name and Password for a PPP Interface

- To create the login name and password for the *ppp-0* interface, enter:

```
$ create ppp security ifname ppp-0 pap login user1 passwd
paswd1
```

This creates the login *user1* and password *paswd1* for PPP interface *ppp-0* and configures it to use PAP authentication. Typically, each PPP interface has a unique login and password created by this command.

**NOTE:**

If you create a PPP interface without issuing this command, the interface will use the login and password of the PPP security default

entry. To create this default entry, either include the command **`create ppp security ifname all`** in the factory defaults file, or enter this command at the CLI prompt, specifying the login and password parameters as shown above.

- To show the currently configured PPP user names, enter:

```
$ get ppp security
```

- To change the password for the ppp-0 interface, enter:

```
$ modify ppp security ifname ppp-0 passwd newpwd
```

The new password newpwd will not take effect until a new PPP session is established, either by rebooting the unit, or by stopping and starting the session using the **`modify ppp intf`** command.

## 2.9 Zero Installation PPPoE Bridge (ZIPB) Mode

Configuring the modem in the Zero Installation PPPOE Bridge or ZIPB mode enables service providers to avoid installing a PPPoE client on subscriber PCs as well as avoid running NAT on the modem. ZIPB combines the advantages of routing and bridging modes.

### 2.9.1 Advantages of the ZIPB mode

Configuring the modem in the ZIPB mode:

- does not require you to install any software on subscriber PCs
- does not require you to run NAT on the modem
- allows you to manage modem for both LAN and WAN sides, because the modem has an IP address on both LAN as well as WAN interfaces.
- allows you to run Firewall/filtering feature on the modem.
- allows you to use bandwidth efficient PPPoA, on the modem's WAN interface.

### 2.9.2 ZIPB mode - operation details

LAN PCs get their global addresses through the DHCP server functionality. If a PPP IP address is available to the unit, the LAN PC gets this address on a DHCP request. Initially, when PPP is not yet up, the IP address allocated to the LAN PC comes from the Ethernet pool, and PPP is triggered to come up.

When the LAN PC sends a renewed request for an IP address allocation, the unit checks if any PPP IP address is free to be allocated. If a PPP IP address is free, then, it will send a NACK to the renewal request for the Ethernet pool IP address. This will force the LAN PC to go in to the DHCP discovery state.

Now, when the unit receives a fresh DHCP discovery message, it will allocate the PPP IP address and simultaneously de-allocate the IP address allocated from the Ethernet pool.

For proper functioning of ZIPB, PPP should be configured with the 'startondata' option. This will ensure that PPP comes up only when the LAN PC is up and that PPP goes down when the LAN PC is switched off.

**NOTE:**

The behavior of 'startondata' is different when ZIPB is enabled. With ZIPB enabled, PPP comes up only if the LAN PC sends a DHCP request for an IP address, and not on any other data activity.

The unit remembers the PPP IP address even after PPP goes down. The unit continues to allocate this same PPP IP address to the LAN PC. So, the user can access the Internet the minute PPP is up. He does not need to wait for IP address allocation, provided PPP comes up with the same IP address.

If the PPP IP address is different from the previously allocated address, it will send a ForceRenew message to the LAN PC. The next time the PC tries to get an IP address, it will get the new PPP IP address.

- To enable ZIPB at the modem, enter,

```
modify zipb cfg enable
```

#### 2.9.2.1 Management from LAN end

When the unit is working in ZIPB mode, the LAN side PCs get the PPP IP addresses allocated to the modem using IPCP or DHCP. If the PPP interface is not up, the PC gets an IP address from the DHCP server pool 0. The DHCP pool has a small default lease and clients will keep sending renew requests after 30 seconds.

**NOTE:**

Each request for renewing an IP address from DHCP pool 0, results in writing to NVRAM and more writes to NVRAM. This reduces the NVRAM life.

LAN machines can access the modem in ZIPB mode, as they would, in non-ZIPB mode, using the Ethernet IP address.

**NOTE:**

Instead of trying to access an IP address, the LAN side PC user should use the DNS relay capability of the modem.

#### 2.9.2.2 Management from WAN end

When in ZIPB mode, with the PPP link up, all requests coming to the modem from the WAN end are passed on to the PC behind it, as the PPP IP address of the modem is considered the IP address of the PC behind the modem. The standard Telnet, FTP and HTTP services on the PC behind the modem, run on ports 21, 20 and 80 respectively. However, if you want to access any of the telnet, ftp or HTTP services on the modem, you can configure the ports to be other than the standard ones used on the PC behind the modem. To access the modem from the WAN end, using telnet or http, use the PPP IP address allocated to the modem. The ports you will specify during WAN-end access should be the same as those specified for the modem in the **nbsize** command.

**NOTE:**

While configuring telnet and http ports, above, from both LAN and WAN ends, the user needs to remember to use the same ports as those mentioned in the **get nbsize** command.

#### 2.9.2.3 Use of ForceRenew in ZIPB mode

When configured in the ZIPB mode, the modem is able to detect that the LAN PC is switched off, and it automatically brings down PPP. When the LAN PC comes up again, the modem senses it, and brings up PPP too.

ForceRenew, as defined in RFC 3203 is used in the ZIPB mode in the following scenarios.

If a LAN client is up with an IP address from the Ethernet pool, and the PPP interface comes up, a ForceRenew message is sent to the client. When the client sends a renew, it is sent a NACK by the server. The Client then sends a Discover and now it can be given any of the free PPP IP addresses maintained by the DHCP server.

The DHCP server initiates ForceRenew for the following trigger points:

- ZIPB is enabled

- ZIPB disabled

- PPP Up trigger

### 2.9.3 Preconditions to configuring the modem in ZIPB mode

- An Ethernet interface should be created. You can use the following syntax,

```
create ethernet intf ifname eth-0 ip 192.168.1.1 mask
255.255.0.0
```

- You need to create and enable a DHCP server pool with poolId 0 and an Ethernet subnet with small lease time. For example, you can use the following syntax.

```
create dhcp server pool poolid 0 start-ip 192.168.1.2 end-ip
192.168.1.5 mask 255.255.0.0 lease 60 mlease 120
```

- Enable dhcp server, by entering,

```
dhcp server cfg enable
```

- You should also configure PPP with startondata.

```
$ create ppp intf ifname ppp-0 ppoe sname test lowif aal5-0
droute true startondata
```

- Configure the ftp, telnet and http ports to be different from the standard ports 23, 20 and 80 respectively, if you want to provide these services on the LAN PC as well.

### 2.9.4 Configuring ZIPB

You can either enable or disable the ZIPB mode on the modem. It is disabled by default. You can set the mode by using either the default configuration (factory defaults file) or CLI commands.

You can dynamically configure the modem to work in the ZIPB mode. When disabled, the modem runs either in bridging or routing mode. When enabled, it runs in the ZIPB mode.

**Configuring using the factory defaults file**

- To enable ZIPB enter:

```
$ modify zipb cfg enable
```

Run the **createfi** utility and upload the image to flash.

**Configuring using CLI**

On the command line interface, use this command to enable ZIPB:

```
$ modify zipb cfg enable
```

**NOTE:**

The preconditions to configuring ZIPB, as mentioned in the section above, need to be met.

## 2.10 Unnumbered PPP Interfaces

The modem's PPP interface is typically assigned a unique IP address from the ISP's PPP server. This IP address must be in a different subnet than the IP addresses assigned to the modem's LAN interface eth-0.

The IP Unnumbered feature provides an alternative configuration that enables the PPP interface to be created with an IP address that is the same as that assigned to the modem's Ethernet interface, eth-0. Using this feature, the PPP interface does not need to obtain an IP address from the ISP.

The PPP interface borrows the IP address from eth-0 to facilitate routing.

During IPCP negotiations with the ISP's server, the PPP interface conveys this address to the other side as its own. If the ISP's server is configured to allow IP Unnumbered connections, then it does not provide another IP address to the PPP interface, as it would in normal operation.

**NOTE:**

If the ISP's PPP server is not configured to allow IP Unnumbered connections, then the server would respond with an IPCP negative acknowledgement (NAK) and instead assign a new IP address to the interface, as it would in normal operation.

The IP Unnumbered feature can be useful in environments in which conserving IP addresses is a priority.

### 2.10.1 Configuration

To configure a PPP interface as IP Unnumbered interface, the PPP interface must be created without an IP address and must specify the interface from which to borrow an IP address (only eth-0 is supported):

**Creating an IP Unnumbered interface**

- The following command creates a PPPoA unnumbered interface that borrows the IP address of eth-0 and specifies this interface as the default route.

```
create ppp intf ifname ppp-0 ppoa lowif aal5-0 numif eth-0
droute true
```

A gateway IP address can also be specified using the `gwy` parameter, or can be learned during the IPCP handshake. A specified gateway IP address will override any address learned via IPCP.

### 2.10.2 Limitations

The following limitations apply when implementing an IP Unnumbered interface:

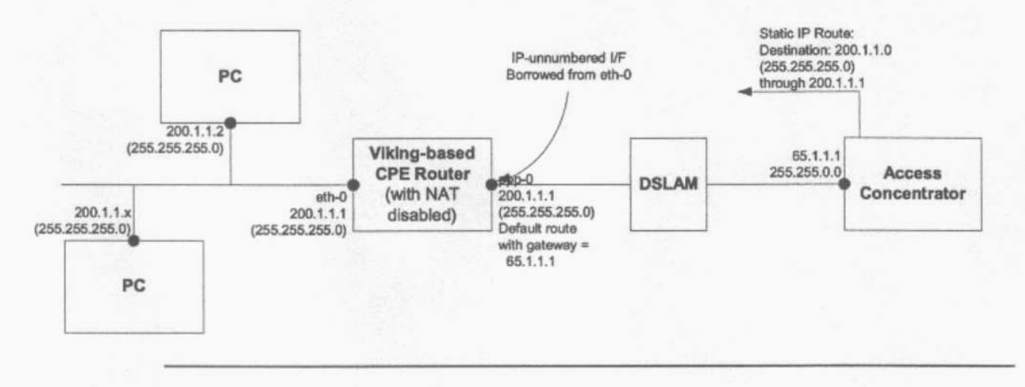
- Only point-to-point interfaces can be IP Unnumbered.
- The interface from which the PPP interface borrows the IP address must be the modem's Ethernet interface, eth-0 ; it cannot be any other LAN interface.

### 2.10.3 IP Unnumbered with NAT

- The interface eth-0 cannot be configured to receive its IP address through DHCP, and the IP address cannot be modified during an active PPP connection.

- The ISP's access server must be configured with an IP route that specifies the LAN's network address as the destination and the interface associated with that user's VPI/VCI as the gateway.

Below provides an illustration of IP Unnumbered configuration.



## 2.10.4 Unnumbered with NAT

The configuration shown above requires each LAN PC to have a public IP address (within a range given by the ISP) and does not make use of Network Address Translation (NAT). However, because each public IP address is normally available only at a cost to the user, there may be cases where the customer has more LAN PCs than available public IP addresses.

For example, a customer may obtain four public IP address from the ISP for use with servers on the LAN (web server, mail server, etc.), but may have 10 additional PCs that use private IP addresses in the subnet 192.168.1.x, mask 255.255.255.0.

The user can configure NAT to enable these 10 PCs to access the internet.

This can be achieved by creating a virtual IP (VIP) LAN interface on the modem with private IP address (say, 192.168.1.1, mask 255.255.255.0). The user would then create a NAT rule (NAPT flavor) to translate the PCs' local IP addresses to the VIP IP address. The following CLI commands create a rule of this type and enable the NAT service:

```
create nat rule entry ruleid 1 napt lcladdrfrom 192.168.1.2
lcladdrto 192.168.1.254
modify nat global enable
```

# 3 Viewing and Modifying DSL Information

This chapter provides details about modifying DSL configuration, and viewing DSL parameters and statistics. The CLI enables you to configure various parameters that control how data is transmitted on the DSL line. You can also view statistics relating to the DSL line performance.

## 3.1 6.1 Modifying the DSL Configuration

You may need to modify various DSL parameters to ensure proper operation of the reference design with your test equipment, or to prepare your customer units for deployment in the user's environment. DSL-related information can be modified using the following command:

```
$ modify dsl config <parameters>
```

The command parameters enable you to change a variety of properties, including the DSL standard to which the firmware complies and the DSL annex type. You can also start and stop operation of the DSL loop and set various operating characteristics, such as the coding gain due to Reed-Solomon or trellis coding, the level of framing overhead, and the power attenuation in dB.

Several examples follow.

**Modifying the DSL *configuration***

- To change the DSL standard to G.dmt (G.992.1), enter:

```
$ modify dsl config gdmt
```

- To change the DSL annex to Annex C, enter:

```
$ modify dsl config annexc
```

- To modify the maximum downstream data rate to ensure interoperability with a service provider's CO equipment, enter:

```
$ modify dsl config maxdownrate <rate>
```

The value you specify (in hexadecimal) is multiplied by 32 Kbits/sec to determine the rate. The default value is *FF*, which results in a maximum data rate of 8 Mbits/sec.

- To enable (default) or disable the operation of the DSL loop, enter:

```
$ modify dsl config loop start
$ modify dsl config loop stop
```

**Viewing the DSL configuration**

- To view current DSL configuration information, enter:

```
$ get dsl config
```

## 3.2 Viewing DSL Parameters and Statistics

You can use the following commands to view a variety of non-modifiable DSL parameters and performance statistics. For a complete list of all parameter values for all the following commands, see the *CLI Manual*.

### Viewing DSL parameters

- To view DSL parameters, enter the following command:

```
$ get dsl params
```

The output displays static DSL information such as the vendor ID and serial number, and calculated values such as far- and near-end RS errors, the signal-to-noise ratio, the calculated line attenuation, and other statistics.

### Viewing DSL statistics

- To view the number of errored, severely errored, and unavailable seconds in the past 15-minute interval and in the past 24 hours, type the following command:

```
$ get dsl stats curr
```

- To view the number of errored, severely errored, and unavailable seconds for eight 15-minute intervals, starting with four intervals ago (i.e., statistics for the intervals from 1 hour ago to 3 hours ago), enter:

```
$ get dsl stats hist 8 4
```

The display also shows the number of intervals in which valid data was transmitted. You can specify up to 96 past intervals to display.

- To view near- and far-end errors counts relating to Reed-Solomon, CRC, and other errors types accumulated since the last reboot, enter:

```
$ get dsl stats cntrs
```

- To view local and remote transmission failures accumulated since the last reboot, enter:

```
$ get dsl stats flrs
```

The output displays loss-of-signal defects (LOS), severely errored frame defects (SEF), no-cell delineation errors, and loss-of-cell delineation errors for the data stream.

### Resetting DSL statistics

- The DSL counters and failure statistics accumulate starting from the last reboot. You can use the following commands to reset these statistics to zero without rebooting:

```
$ reset dsl stats flrs
$ reset dsl stats cntrs
```

# 4 Configuring DNS Relay

This chapter describes the commands for configuring the unit as a DNS relay server.

## 4.1 Overview of DNS Relay

PCs on a LAN can set the IP address of the unit as the DNS server. The SAR130 unit will thus act as a DNS relay server and forward the requests received from the PCs on the LAN to the actual DNS servers, whose addresses have been learned from PPP. If the PPP connection is not available (because of inactivity) at the time a DNS request is received from the LAN, the PPP link will start automatically. All the responses received from the DNS server will be forwarded to the LAN PCs.

When the unit is configured as a DNS relay server, a user will not need to change the DNS server IP address on their PC whenever their ISP changes DNS servers, or when the user connects to a different ISP.

## 4.2 Configuration Details

**Enabling/Disabling DNS Relay**

- To enable or disable DNS relay, enter:

```
modify dns relay cfg [enable|disable]
```

**Displaying Current Status**

- To display the current DNS relay status, enter:

```
get dns relay cfg
```

### 4.2.1 Polling DNS Servers

The DNS polling feature on the modem enables you to continuously monitor the availability of DNS servers. If none of the DNS servers are available, the modem generates a trap “DNS servers not responding”.

- To enable this DNS polling feature, enter:

```
modify dns relay cfg pollstatus enable
```

- To change the polling interval time to 5 minutes, enter:

```
modify dns relay cfg pollinterval 5
```

# 5 Configuring Dynamic DNS

This chapter provides you an overview of the Dynamic DNS feature of the modem, and configuration details related to it.

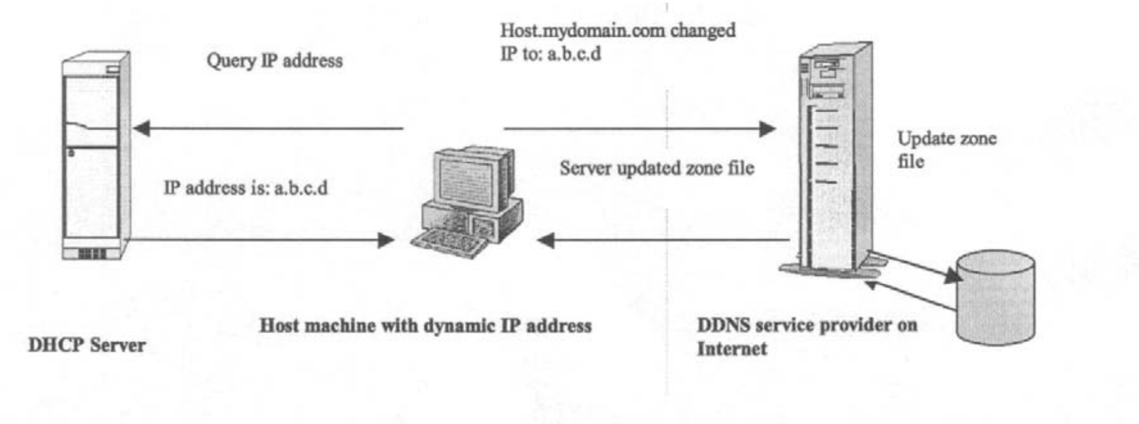
## 5.1 Overview

### Why do we need Dynamic *DNS?*

If some host has a dynamic IP address that keeps changing frequently then it is difficult to keep updating the IP record that is associated with the domain name of this host in the zone files. This will result in non-accessibility of this host on the Internet. Dynamic DNS service allows to keep mapping of a dynamic IP address of such host to a static hostname. Dynamic DNS services are provided by many websites. The host needs to register with some website and get a domain name. When the IP address of the host changes it just needs to send a message to the website that's providing dynamic DNS service to this host.

### How does Dynamic DNS *work?*

For this to work, an automated update client needs to be implemented. These update clients send update messages to the servers whenever there is some change in the IP address of that host. Then, the server updates the entries for that host and replies back with some return code.



The above figure explains one such scenario in which a host gets a dynamic IP address for itself from a DHCP server. As the host has registered with one of the dynamic DNS service providers on the Internet, it sends an update message to the service provider with host name and changed IP address. The service provider updates the new IP address of the host in the zone files that have entry for that host name and replies back with some return code. The return code communicates the success or failure of the update message. This process is repeated every time the host's IP address changes.

### Storing the IP address on the *NVRAM*

If the dynamic DNS service provider is notified of the same IP address again and again, then it considers it an abuse and might block the host name. To avoid this scenario, the IP address that was successfully updated to the ISP, is stored on the unit. Whenever we receive an IP address change notification, the new IP address is compared with the IP address that was stored on the last update. If they differ then only an update request is sent. However when the system comes up there is no way of knowing what was the IP address on last successful update before the system went down. So the modem keeps writing this IP address in NVRAM periodically.

## 5.2 Configuration Details

You need to create dynamic DNS per interface and can only create one service on one interface.

You first need to create host names for a public interface and then create a service on that interface. Hostnames can be added or deleted from that interface dynamically.

### Creating host names

- To create a Dynamic DNS Host Name Table, use the CLI command:

```
create ddns hostname ifname ifname name name
```

In this command, you can specify the interface name of the public interface for which this entry defines the Dynamic DNS Host Name. You can create host names for any of the following interfaces - PPP, IPoA, EoA, Ethernet, and WLAN.

### Deleting host names

- To delete Dynamic DNS Host Name Table, use the CLI command:

```
delete ddns hostname ifname ifname name name
```

### Creating DDNS Interfaces

- To create Dynamic DNS Service Configuration, enter the CLI command:

```
create ddns intf ifname ifname srvcname tzo | dyndns username
username passwd passwd [ system dynamic | static | custom ] [
wildcard enable | disable ] [ mailexchger mailexchger ] [
mailbackup enable | disable ] [ offlinesupport enable | disable
]
```

***ifname*** *- This parameter is the* interface name of the public interface for which this entry defines the Dynamic DNS profile.

***srvcname -*** This is the name of the Dynamic DNS service provider where you have registered and have an account. Currently, you can configure it to either, www.tzo.com or www.dyndns.org

***username -*** Username registered at service provider. The value is to be given in double quotes.

***passwd -*** Password provided by service provider.

***system -*** www.dyndns.org provides three kinds of services - Dynamic DNS, Custom DNS and Static DNS. You can create different domains in these systems. Custom DNS service is a full DNS solution for newly purchased domains or domains you already own. A web-based interface provides complete control over resource records and your entire domain, including support for dynamic IPs and automated updates. Static DNS service points a DNS hostname in some domain owned by dyndns.org to the user's ISPassigned static or pseudo-static IP address. DynDNS service points a fixed hostname in some domain owned by dyndns.org to the user's ISP-assigned dynamic IP address. This allows more frequent update of IP addresses, than allowed by Static DNS.

***wildcard -*** It specifies whether Wildcard CNAME are to be resolved or not. If enabled, addresses *.yourhost.ourdomain.ext are aliased to the same address as yourhost.ourdomain.ext.

***mailexchger -*** It specifies a Mail Exchanger (MX) for use with the hostname being modified. The specified MX must resolve to an IP address, or it will be ignored. Providing no MX setting (or an MX that does not resolve properly to an A record) will cause the hostname's MX record(s) to be removed.

***mailbackup -*** It specifies whether mails are to be backed up by the service provider.

***offlinesupport*** – If enabled, the service provider redirects browsers to its own site if the registered host is currently offline.

**NOTE:**

You can effectively configure the parameters system, wildcard, mailexchger, mailbackup and offlinesupport, only if your service is www.dyndns.org. If you have any other service, configuring these parameters will not have any effect.

### Deleting the service from an interface

- To delete Dynamic DNS Service Configuration, use the CLI command:

```
delete ddns intf ifname ifname
```

If the service is not deleted and only one hostname is left, then the hostname cannot be deleted without deleting the service on that interface. You need to ensure that when you want to change the service on a given interface, you need to delete the service. If the hostnames are different for the new service then you should delete all the existing hosts before creating the new service with new hostnames.

# 6 Configuring DHCP Server and DHCP Relay

This chapter describes the commands for configuring the unit as a Dynamic Host Configuration Protocol (DHCP) server, DHCP relay agent, and DHCP client.

• As a DHCP server, the unit maintains a pool of IP addresses and distributes them to LAN hosts whenever they are switched on.

• As a DHCP relay agent, the unit forwards requests for IP addresses from LAN hosts to a DHCP server (often at the ISP's location), and then returns the IP information from the DHCP server to the hosts.

• If the LAN uses its own DHCP server, then the LAN interface on the unit can be configured as a DHCP client of that server.

## 6.1 Default DHCP Configuration on the SAR130 Reference Unit

By default, the SAR130 unit is configured as a DHCP server, with two pools of IP addresses. The following commands are included in the default configuration file to set this configuration:

The following command creates a single pool of IP addresses from 192.168.1.2 through 192.168.1.34 for distribution via the eth-0 interface to LAN hosts:

```
$ create dhcp server pool start-ip 192.168.1.2 end-ip
192.168.1.34 mask 255.255.255.0 gwy 192.168.1.1 enable
```

The following line enables DHCP server mode.

```
$ modify dhcp server cfg enable
```

**NOTE:**

For the unit to operate as a DHCP server, hosts must be configured to *accept IP information dynamically.*

## 6.2 Configuring Unit as DHCP Server

The two commands used in the default configuration provide the basic instructions for configuring the device as a DHCP server. This section explains those commands in detail and describes additional commands and parameters you can use in your own configuration.

### 6.2.1 Creating DHCP Pools

A *DHCP pool* is a range of IP addresses made available on a server for distribution to LAN hosts.

**Creating a Basic DHCP Pool**

• To create a basic DHCP pool, enter:

```
$ create dhcp server pool start-ip 192.168.1.2 end-ip
192.168.1.13 mask 255.255.255.0
```

This command configures a pool of 12 IP addresses, from 192.168.1.2 to 192.168.1.13. The mask indicates that this pool is applicable to the subnet 192.168.1.0.

**NOTE:**

The IP addresses specified in the pool must belong to same subnet as the physical ethernet interface or one of the virtual ethernet interfaces.

### Viewing Pools

- To see the pool(s) you have created, along with other configurable parameters, enter:

```
$ get dhcp server pool
```

In addition to specifying the address ranges and network mask, you can add parameters to the **create** command, to:

• Assign a specific ID number to the pool (if not specified, the next sequential

number will be assigned by default, starting from 0 as first pool).

• Specify a lease period, which identifies how long computers can use a particular IP address before it is returned to the address pool (if not specified, the lease period is set to 1 day by default)

• Specify a low threshold, which determines when the unit will send an alert that the available pool of addresses is getting low.

• Specify a domain name for hosts that receive addresses from this pool (for the administrator's reference).

• Specify IP addresses that hosts should use to access various network servers (such as DNS, WINS, POP3 servers).

• Enable or disable the pool. IP addresses cannot be assigned from a disabled pool, but the configuration remains on the system for future activation.

### Assigning a Pool ID

You can configure multiple pools for assignment to different subnets on the LAN. Each pool is distinguished by a pool id. If the pool id is not specified in the **create** command, the pool is assigned the first available pool id.

- To assign a particular pool id number include it in the command:

```
$ create dhcp server pool poolid 2 start-ip 192.168.1.3 end-ip
192.168.1.34 mask 255.255.255.0
```

### Specifying a Lease Period

The DHCP server allocates IP addresses to clients for a specified duration, called the lease. You can specify a default lease period and a maximum lease period for each DHCP pool. If you do not specify a lease period when you create the pool, the default lease period is set to 2592000 seconds (30 days) and the maximum lease is set to 31536000 seconds (365 days). When the lease period expires, the client may again request an IP address from the DHCP server. The client can request an address for a specific lease duration. The server will grant the request if the duration is less than the maximum lease period configured for the pool. If the client request does not specify a lease duration, the server assigns an IP address for the default lease period.

- To specify a lease period of one day, for example, create pool as follows:

```
$ create dhcp server pool poolid 2 start-ip 192.168.1.2 end-ip
192.168.1.13 mask 255.255.255.0 dlease 86400 mlease 4294967295
```

If you do not want to limit the lease period, you can set the default lease and maximum lease periods to the maximum of 4294967295 seconds (greater than 136 years).

### Specifying a Low Threshold *Value*

Each DHCP pool has a low threshold value associated with it. Whenever the number of available IP addresses in the pool drops below this threshold, the server produces the low threshold hit trap. This trap indicates that available addresses from the pool may soon be exhausted and hosts coming up on the LAN will not be allocated IP addresses dynamically. By default, the low threshold parameter is given a value of 0. Since the number of available addresses can never fall below 0, this means that the trap will be generated only if you have specifically set the low threshold to a non-zero value.

- To specify the low threshold value, type:

```
$ create dhcp server pool poolid 2 start-ip 192.168.1.2 end-ip
192.168.1.13 mask 255.255.255.0 lthres 3
```

### Enabling and Disabling Pools

By default, when you create a new pool it is enabled for use. You can disable a pool if you do not want to use it currently, but want to retain the information for future use.

- To disable a pool, type the command:

```
$ modify dhcp server pool poolid 0 disable
```

- To re-enable the pool, type the command:

```
$ modify dhcp server pool poolid 0 enable
```

### 6.2.2 Excluding Addresses from a Pool

If you do not want particular addresses to be assigned to LAN hosts, you can add these to a pool exclusion table.

- To mark an address as unusable (for example 192.168.1.13) from an existing pool, type the command:

```
$ create dhcp server exclude ip 192.168.1.13
```

- To remove the entry from the pool exclusion table and make it available for use, type the command:

```
$ delete dhcp server exclude ip 192.168.1.13
```

- To view the pool exclusion table entries, type the command:

```
$ get dhcp server exclude
```

### 6.2.3 14.2.3 Modifying and Deleting Pools

You can modify the lease and other configurable parameters using the `modify` command or by giving the relevant parameters directly in the `create` command.

**Modifying Pools**

- To modify a DHCP pool, use the command:

```
modify dhcp server pool poolid 0 [parameter value]
```

For example, to modify the DNS server assigned to DHCP clients, use:

```
$ modify dhcp server pool poolid 0 dns 192.168.1.11
```

The modification will be reflected on the host whenever it reboots next and gets its address and other parameters from the modem.

**Deleting Pools**

- To delete a pool with pool id 2, use the command:

```
$ delete dhcp server pool poolid 2
```

**NOTE:**

If you delete a pool or modify its settings while IP addresses are currently allocated from the pool, the hosts will continue to use the allocated IP addresses with the original settings, till the next renewal.

### 6.2.4 Creating Static DHCP Assignments

The DHCP server will attempt to assign the same address to a host each time the host boots; however, this may not always be possible. A host may be assigned an IP address, different from the one it previously used, depending on the available addresses. In some situations, it may be important that the DHCP server always assigns the same IP address to a particular host.

The unit's static hosts table enables you to create a permanent one-to-one association between a host and an IP address. On the LAN, a particular host is uniquely identified by its MAC address. A static host entry stores the association between the MAC address and a fixed IP address, as in the following example.

**Adding an Entry to the Static *Hosts Table***

- To create an entry in the static hosts table that associates the MAC address 00:80:48:CB: B8: 83 with the fixed IP address 192.168.1.2, type the command:

```
$ create dhcp server host ip 192.168.1.2 mask 255.255.255.0
hwaddr 00:80:48:CB:B8:83 dlease 4294967295 mlease 4294967295
```

The lease periods carry the same meaning as for a pool. You can set them to 4294967295 if you do not want to limit the lease period. The hardware address parameter (hwaddr) refers to the MAC address or the Ethernet address. The specified IP address is reserved for the host regardless of whether the host is currently switched on. The IP address will not be allocated to any other host.

**Viewing the Static Hosts *Table***

- To see the details of all configured static hosts, type the command:

```
$ get dhcp server host
```

**Deleting an Entry from the *Static Hosts Table***

- To delete a static host entry that is no longer required, use the command:

```
$ delete dhcp server host ip 192.168.1.3
```

- To delete all static hosts, type the command without specifying an IP address:

```
$ delete dhcp server host
```

**NOTE:**

After you delete the entry, the client will continue to use the IP address, but only for the next renewal.

### 6.2.5 Enabling the DHCP Server

After configuring DHCP pools, you must enable the DHCP server.

- To enable the server, type the command:

```
$ modify dhcp server cfg enable
```

- To disable the DHCP server, type the command:

```
$ modify dhcp server cfg disable
```

- To see the current state of the server, type the command:

```
$ get dhcp server cfg
```

### 6.2.6 DHCP- DNS Relay Interaction

The DHCP server indicates the DNS Server addresses to DHCP clients in the following manner.

- If the primary/secondary DNS addresses are provided as part of the pool configuration (using the DNS and SDNS parameters in the **create/modify dhcp server pool** commands), then these are indicated to the client.

- If the DNS and SDNS addresses are not specified in the pool configuration, then one of the following cases will arise.

- If the DNS relay is enabled, the DHCP server gives the modem's LAN IP address as the DNS server address to the clients.(The actual DNS servers are learned by the modem, dynamically, via the PPP link. These can be viewed using the **get dhcp server cfg** command.) - If the DNS relay is disabled, the DNS addresses indicated to the client are the ones that are dynamically learned from the PPP link. That is, they are the same as the ones displayed by the **get dhcp server cfg** command.

### 6.2.7 Viewing DHCP Server Address Assignments

Once the server starts assigning IP addresses to clients, you can see the currently allocated addresses.

- To see the currently allocated addresses, type the command:

```
$ get dhcp server address
```

## 6.3 Configuring DHCP Relay

To use the ISP's DHCP server, you can configure the unit to act as a *DHCP relay agent*. As a relay agent, the unit forwards DHCP requests from the LAN hosts on to the ISP. The ISP's DHCP server then sends back IP addresses and other configuration information, which the unit forwards to the LAN hosts.

To configure the unit as a DHCP relay agent, you first specify the interfaces on which the unit will listen for DHCP requests and responses. Then, you enable DHCP relay mode.

### 6.3.1 Configuring the DHCP Relay Interfaces

The unit's LAN interface must be enabled for DHCP relay in order to receive requests from the LAN hosts for IP information. If multiple LAN interfaces are defined on the unit, the DHCP relay service can be enabled on each interface simultaneously.

To receive responses from the ISP, the unit's WAN interface must also be enabled for DHCP relay. The WAN interface could be a PPP, EoA, or an IpoA interface.

**Specifying the DHCP Relay *Interfaces***

- To specify that the unit will receive DHCP requests on the LAN (eth-0) interface and the WAN (ppp-0) interface, enter these commands:

```
$ create dhcp relay intf ifname eth-0
$ create dhcp relay intf ifname ppp-0
```

**Viewing DHCP Relay *Interfaces***

- To see all the interfaces on which DHCP relay is enabled, type the command:

```
$ get dhcp relay intf
```

### 6.3.2 Specifying the DHCP Server IP Address

You can specify the IP address of the DHCP server by modifying the DHCP relay configuration. It is not mandatory to configure this address. The ISP should be able to route the request to the appropriate server. If you assign the DHCP server IP address, you should also define a route in the unit's IP routing table.

**Specifying the DHCP Server *IP address***

- To specify the IP address of the ISP's DHCP server (202.64.23.4 in this example), use the command:

```
$ modify dhcp relay cfg ip 202.64.23.4
```

### 6.3.3 Enabling DHCP Relay Mode

- To enable the DHCP relay use the command:

```
$ modify dhcp relay cfg enable
```

**NOTE:**

You can enable only one DHCP server or relay at a time. To enable DHCP relay, the DHCP server must be disabled.

- To see the current configuration of the DHCP relay agent, type the command:

```
$ get dhcp relay cfg
```

## 6.4 Using a DHCP Server on the LAN

If the unit is connected to a LAN that uses one of its own hosts as the DHCP server, the unit's LAN interface must be configured as a DHCP client so that it also gets its LAN-side IP address from the server.

### Specifying the LAN interface *as a DHCP client*

- To configure the modem's LAN interface as a DHCP client, create an ethernet interface without specifying an IP address. To do so, use the command:

```
$ create ethernet intf ifname eth-0 usedhcp true
```

- To see the state of the DHCP client, type the command:

```
$ get dhcp client info ifname eth-0
```

The Status field will show Bound**,** once the modem has obtained an IP address from the DHCP server.

- To see the actual IP address assigned to the modem, type the command:

```
$ get ip address
```

This command shows the IP addresses assigned to all the modem's interfaces. The entry for eth-0 will show the IP address assigned to the modem by the DHCP server.

## 6.5 DHCP Traps

The DHCP server not only provides automatic configuration for LAN hosts, but also watches for potential errors in configuration and informs you about them via the following traps.

### 6.5.1 Duplicate IP Address Trap

The duplicate IP address trap may occur when the unit is operating as a DHCP server. Before assigning an address to a requesting host, the unit probes the LAN to see if another host on the LAN is already using the address. If so, the server raises a duplicate IP address trap and assigns the next available address to the host.

### 6.5.2 Low Threshold Hit Trap

This trap is generated when the number of available IP addresses in a DHCP pool is below the low threshold assigned to the pool. For instructions on setting the threshold value, please refer to the **`create dhcp server pool`** command.

## 6.6 ForceRenew

ForceRenew is supported by the DHCP server configured at the modem, according to RFC 3203. If DHCP client(s) also support ForceRenew, it is possible to increase the lease time defined in the pool. Authentication, as defined in RFC 3118, should also be sent in a ForceRenew message. At the client end too, there should exist a mechanism to configure authentication information to use ForceRenew procedure effectively.

# 7 Simple Network Time Protocol

## 7.1 Overview

The SAR130 software implements Simple Network Time Protocol (SNTP), Version 4, RFC 2030, to enable it to periodically synchronize its clock with a reference clock on the Internet. The firewall feature of the modem requires synchronized wall clock time. Firewall rules, which trigger a particular instance of time, require the triggering time to be absolute, i.e., hr:min:sec, mm/dd/yyyy, or periodic. Absolute time refers to an exact and particular point in time, while Periodic time indicates the lapse of time with reference to a pre-defined moment in time. This time synchronization protocol (SNTP) enables the wall clock time to be first initialized on the modem. Subsequently, SNTP enables the wall clock time to remain synchronized with an external reference clock on the Internet.

Simple Network Time Protocol (SNTP) is a simplified adaptation of the Network Time Protocol (NTP), that is used to synchronize computer clocks on the Internet. SNTP exchanges timekeeping information between servers and clients via the Internet. Extremely reliable sources, such as radio clocks and GPS satellite timing receivers, typically act as primary servers.

The SNTP client sends a request to a designated server at its unicast address and expects a reply. This reply helps it to determine the time and optionally, the round-trip delay and local clock offset, relative to the server.

SNTP uses User Datagram Protocol (UDP) for the transport and the UDP port number assigned to SNTP is 123.

## 7.2 SNTP implementation details

### Synchronization Request and Response

The SNTP client, at the modem end, sends an SNTP request to the SNTP server for synchronization. The user can configure up to five SNTP servers, and the SNTP client sends an SNTP request to the first SNTP server in the list. If this server stops responding then the server mentioned in the next entry, is contacted. These periodic requests help the modem SNTP clock to be synchronized with the Network Time.

### Polling Interval and Packet *Time-out*

The SNTP Polling Interval, or the time after which an SNTP request is sent, can be between 64 seconds to 1024 seconds, both inclusive. The polling interval adjusts automatically, depending on the clock drift. The maximum number of retries, in case of no response from server, is 2. The wait time for the response, or the packet time-out is 5 seconds.

**Response validation**

Validation of responses from the server occurs at the modem end. A response is rejected if:

- the timestamp stored at the time of sending request does not match with the Originated Timestamp field of SNTP response.

- the deviation (calculated from SNTP response) in local clock is more than the polling interval.

**Amortization**

The very first time synchronization happens, the local clock is simply set to the server time. Very sudden or large changes in time never occur, due to amortization. If the local clock is lagging behind the Network clock, for less than a second, the local clock may jump to cover the lagging time. However, if the gap is more than one second, the time gradually increments at the local clock end, over a period of time, which is divided in to smaller synchronization periods.

If the local clock is leading, it does not go back to get synchronized with the network time.If the local clock is leading by 1 second or less, it will pause for the leading time period. If it is leading for a value greater than 1 second, it will gradually decrement the time. In this scheme whole synchronization period will be divided in to time intervals, and the time change will gradually occur over smaller synchronization periods.

**Alarm Timer**

It is possible to set periodic and one-shot alarms, synchronized with the network time, on applications connected to the modem. It is also possible to set the absolute time alarm system.

**NOTE:**

The system clock cannot be configured using CLI commands, while SNTP is enabled. You must first disable SNTP before modifying the system clock. All the SNTP time-based alarms will be affected by this operation. They will either expire early or late. Also, few may expire simultaneously.

## 7.3 Configuration details

**Enabling or Disabling SNTP service**

- To modify the SNTP configuration, enter:

```
modify sntp cfg [enable | disable]
```

**Configuring SNTP server address**

- To configure the SNTP server address, enter:

```
create sntp servaddr <ip-address> | dname <domain-name>
```

- To delete the SNTP server address you have configured, enter:

```
delete sntp servaddr < ip-address | dname domain-name >
```

### Obtaining SNTP server address information

- To get SNTP server address information, enter:

```
get sntp servaddr [<ip-address> | dname <domain-name>]
```

### Obtaining SNTP configuration information

- To get SNTP configuration information, enter:

```
get sntp cfg
```

This command indicates whether the SNTP service is enabled or disabled.

### Obtaining SNTP statistical information

- To get statistical information about SNTP, enter:

```
get sntp stats
```

This command displays

- the number of SNTP Requests sent to the SNTP server
- the number of valid SNTP responses received from the SNTP server
- the number of invalid SNTP responses received from the SNTP server
- the number of lost responses against the SNTP request
- the time at which the local clock was last set or corrected.

### Resetting SNTP statistics

- To reset SNTP statistics, enter:

```
reset sntp stats
```

# 8 Layer 2 Security

Traffic flowing towards the modem can be blocked or filtered at different levels. Bridge mode (layer 2) traffic does not reach the IP layer (layer 3), and therefore requires its own security settings. This chapter describes methods for filtering traffic at layer 2. These include,

- Configuring raw filters
- Protocol blocking
- Implementing L2 Wall

## 8.1 Raw Filtering – Overview

This section provides details about the SAR130 unit's raw filtering capability and how to configure the rules and subrules for raw filtering.

The SAR130 unit's raw filtering feature allows it to examine each packet traveling in either direction (incoming or outgoing) and to filter out packets based on rules and subrules that you define. Because the raw filter scans packets at the layer 2 level (e.g., Ethernet), it can be used with either operating mode (i.e., bridge or router).

You can specify multiple rules, each with one or more subrules that apply only to that rule. Each rule tells what to do (accept or deny) to a packet that is moving in the specified direction (incoming or outgoing) on the specified interface, if that packet also matches the pattern(s) specified by the rule's subrule(s).

Each rule and subrule is also assigned an ID number. Rule IDs must be unique; subrule IDs must be unique within a rule. Like NAT rules, these ID numbers determine the order in which rules are evaluated—from lowest to highest number. A rule's subrules are evaluated in this manner as well.

**NOTE:**

To allow you to retain full control over the order of rule evaluation, do not number rules/subrules consecutively, e.g., 1, 2, 3, etc., but in increments, e.g., 10, 20, 30, etc. This will allow you to insert more rules/subrules between the existing ones at a later time. If you number your rules/subrules consecutively, you will have to delete and recreate all existing rules that are to follow the new rule.

When raw filtering is enabled, the unit scans the raw filter rules whenever a packet is received; if a rule is found that matches the packet, the packet is accepted or denied as specified by the rule.

A rule is said to match the packet only if all of its subrules match the packet. This is true whether a rule has one or many subrules. If a subrule is found that does not match the packet, that rule is skipped.

If none of the rules matches the packet, the default action is taken for that packet. The default action is specified as part of the raw filter global configuration.

The maximum number of rules and subrules is determined by the maxpfrawrule and maxpfrawsubrule parameters in the size command.

### 8.1.1 Using Raw Filtering Rules and Subrules

Rules specify, at a minimum, the interface and direction to be monitored, and the action to take if a packet matches the rule. Subrules specify the actual pattern to be searched for in each packet.

#### 8.1.1.1 Commands for rules

The basic commands used to create, modify and delete raw filter rules are described below.

**Creating a raw filter rule**

- To create a raw filter rule, enter:

```
$ create pfraw rule entry ruleid 100 ifname ppp-0 dir in enable
log match act deny
```

**NOTE:**

It is also possible to specify an interface type, either private, public, or demilitarized, while specifying a rule for an interface type. For LAN interfaces such as Ethernet, you can specify private interfaces. For WAN interfaces, you can specify public interfaces.

This command creates a rule with rule ID 100 on interface ppp-0, for packets traveling in the incoming direction (dir in). The enable parameter indicates that the rule is enabled; log match indicates that all matching packets will be logged, and act deny indicates that if a packet matches the rule, the action is to deny the packet.

**Modifying a raw filter rule**

- To modify a rule, enter:

```
$ modify pfraw rule entry ruleid 100 act accept
```

**Deleting a raw filter rule**

- To delete a rule, enter:

```
$ delete pfraw rule entry ruleid 1
```

**NOTE:**

In order to delete a rule, you must first delete all of its subrules. (For information on deleting a subrule, refer to the following section.)

#### 8.1.1.2 Commands for subrules

The basic commands used to create, modify and delete raw filter subrules are described below.

**Creating a raw filter subrule**

- To create a subrule, enter:

```
$ create pfraw subrule entry ruleid 100 subruleid 10 start
linkh offset 0 mask 0xffffffffffff cmpt eq 0xffffffffffff
enable
```

This creates subrule 10 of rule 100. This subrule examines packets at an offset of 0 relative to the link layer header. If masking with 0xffffffffffff gives a result of 0xffffffffffff, the match is successful and the packet is accepted or denied as specified by rule 100.

**Modifying a raw filter subrule**

- To modify a subrule, enter:

```
$ modify pfraw subrule entry ruleid 100 subruleid 10 disable
```

**Deleting a raw filter subrule**

- To delete a subrule, enter:

```
$ delete pfraw subrule entry ruleid 2 subruleid 1
```

#### 8.1.1.3 Displaying rule/subrule configuration

The basic commands used to show the current configuration of rules and subrules are described below.

**Viewing raw filter rules**

- To see the configuration of all rules and subrules, enter:

```
$ get pfraw rule info
```

- To see the configuration of rules applicable to incoming packets on a particular interface, such as eth-0, enter:

```
$ get pfraw rule info ifname eth-0 dir in
```

- To see the configuration of rules applicable to outgoing packets on a particular interface, such as eth-0, enter:

```
$ get pfraw rule info ifname eth-0 dir out
```

**Example**

The following rule and subrule can be used to block all incoming Telnet requests on the ppp-0 interface:

```
$ create pfraw rule entry ruleid 200 ifname ppp-0 dir in act
deny log match enable
$ create pfraw subrule entry ruleid 200 subruleid 20 mask
0xffff start tcph offset 2 cmpt eq 0x0017 enable
```

The subrule matches packets with the value 0x0017 (the Telnet port number) at offset 2 in the TCP header; the rule specifies that the action is deny; thus, all incoming Telnet packets will be dropped.

### 8.1.2 Raw Filtering Global Configuration

- To enable or disable raw filtering and to set the default action, enter:

```
$ modify pfraw global enable accept
```

This command enables raw filtering and specifies that the default action (i.e., the action taken if no rules are matched) is to accept the packet.

**NOTE:**

Do not enable raw filtering without first configuring raw filter rules.

## 8.2 Protocol Blocking

The protocol blocking feature is provided for end users to have a simple way of preventing certain protocols from being trashiness on their network (i.e., without having to create IP filter or raw filter rules). In the Web-based interface, users simply click on the protocol by name to enable it to be blocked. This internally takes care of creating the required filtering rules .

- To view the list of protocols that have predefined filter rules, use the following command:

```
$ get pfraw block?
```

- To modify the pfraw blocking status for the PPPoE protocol, for example, enter:

```
$ modify pfraw block protocol ppe enable
```

## 8.3 L2 Wall

The SAR130 software supports the L2 Wall security feature, which allows a LAN host to prevent accesses to it when the user is not using the Internet.

When active, L2 Wall causes all packets incoming to the host to be dropped, except for packets whose protocols have been specified as “transparent” to the L2 Wall. For example, DHCP and ARP can be specified as transparent protocols to allow DHCP renewals and ARP requests.

**NOTE:**

In this chapter, the term “dropped traffic” does not include transparent traffic.

### 8.3.1 Overview

L2 Wall has three modes, which can be set by the end user:

- On
- Off
- Automatic

When L2 Wall is On, no traffic may pass to or from the host. When L2 Wall is Off, all traffic may pass in both directions.

In Automatic mode, L2 Wall is activated and deactivated automatically, depending on whether there has been recent non-transparent traffic in the outgoing direction. Whenever such traffic is transmitted, a timer is set. If the timer expires without further outgoing traffic, L2 Wall is activated and no further incoming traffic is allowed. When the host again sends outgoing nontransparent traffic, L2 Wall is deactivated and the timer is reset.

The timer counts down an interval called the activation time, which is set by the user and can vary from 1 minute up to 1 day. During this interval, traffic can pass in both directions.

### 8.3.2 Configuration Files

L2 Wall filtering is controlled by three configuration files that are merged into the software image when you create it using the Createfi utility:

- In the factory defaults file *TEFacs.txt*, you set L2 Wall to automatic mode (unless already set by default in the software), and then add CLI commands that create raw filtering rules to be activated when the L2 Wall is on. These rules allow certain types of traffic (i.e., transparent traffic) to be passed even though the L2 Wall is on.

- The text file *l2wall_on.cfg* contains a CLI command that configures the raw filtering global setting to deny all traffic. You add commands that enable the raw filtering rules defined in TEFacs.txt that configure transparent traffic.

- The text file *l2wall_off.cfg* contains a CLI command that configures the raw filtering global setting to accept all traffic. You add CLI commands that disable the same raw filtering rules that were enabled in l2wall_on.cfg.

Whenever the L2 Wall mode is set to On, the system executes `l2wall_on.cfg`, and whenever the mode is set to Off, the system executes `l2wall_off.cfg`.

The file `l2wall_on.cfg` turns on the raw filter, specifying that packets not matching any raw filtering rules should be dropped, and defines the raw filter rules to be applied, while `l2wall_off.cfg` deletes the raw filter rules created by `l2wall_on.cfg`. The rules in `l2wall_on.cfg` thus define the “transparent” protocols, i.e., those protocols that can pass through while L2 Wall is active. L2 Wall can be controlled by the following CLI commands (described in the CLI Manual):

- modify L2wall cfg

- get L2wall cfg

### 8.3.3 L2Wall Algorithm

The basic algorithm for the L2 Wall feature is as follows.

```
If (L2Wall is OFF) OR (L2Wall is AUTO) AND (time since last activity <
activation time) Continue operating normally (i.e. bridged traffic)
Elseif (L2Wall is AUTO) AND (time since last activity > activation
time)
If the packet’s protocol is transparent to L2Wall
//i.e. the protocol is enabled in l2wall_on.cfg
Forward the packet //from LAN to WAN or vice versa
Else //not a transparent protocol
If packet originated from LAN Reset the time since last activity
Apply l2wall_off.cfg to disable filter rules
Forward the packet to the WAN
Else //packet originated from WAN
Packet is dropped due to filter in l2wall_on.cfg
Endif
Endif
Else //L2Wall is ON
If the packet’s protocol is transparent to L2Wall
//i.e. the protocol is enabled in l2wall_on.cfg
Forward the packet //from LAN to WAN or vice versa
Else //not a transparent protocol
Packet is dropped due to filter enabled in l2wall_on.cfg
Endif
Endif
```

### 8.3.4 Assumptions

L2 Wall assumes the following to be true:

- The SAR130 unit is operating in bridging mode.

# 9 Layer 3 Security

Layer 3 filtering at the IP layer enables easier configuration, as it allows working with various fields in the IP header. Also, as more information about the traffic flow is available at this layer, it allows you to provide increased protection.

To enable this protection, you need to configure NAT, Firewall and IP Filter.

This chapter provides you the required configuration details.

## 9.1 NAT

This section describes how to create and use Network Address Translation (NAT) rules and application level gateways (ALGs).

A *NAT rule* specifies when and how to translate IP addresses. As data packets are received on the unit's interfaces, data in their protocol headers is compared to criteria established in the NAT rules. The criteria includes ranges of source or destination addresses. If the packet meets the criteria of one of the rules, the packet header undergoes the translation specified by the rule and the revised packet is forwarded. If the packet does not match any rule criteria, it is forwarded without translation.

Six types, or flavors, of NAT rules are supported. They are: *basic*, *napt*, *filter*, *rdr*, *bimap*, and *pass*. The most commonly used flavors are napt and rdr. NAPT rules are used to translate multiple private addresses on a LAN to a single public IP address for external communication. An rdr rule can be used to allow external access to a privately addressed LAN computer.

**Application Level Gateways**

Creating rules is sufficient to handle most applications. However, applications such as FTP, H.323, Real Audio, CUseeMe, and others require additional configuration in the form of application level gateways (ALGs). These applications require ALGs because their payloads—not just the packet headers—contain IP addresses. When an ALG is configured, NAT translations occur not only on data in a packet's header, but also on data in the packet's payload.

### 9.1.1 Default NAT Configuration on the SAR130 Unit

By default, NAT is enabled on the SAR130 unit, with an napt rule that translates all LAN side addresses to the public IP address assigned to the PPP-0 interface.

The following commands are included in the default configuration file to set this configuration:

```
$ create nat rule entry ruleid 1 napt
$ modify nat global enable
```

The first line creates a rule of type **napt** (Network Address Port Translation) and assigns it a rule ID of 1. The second line enables the NAT service.

### 9.1.2 Configuring NAT Direction

NAT distinguishes between inside interfaces and outside interfaces. An inside interface is one on which you can use private IP addresses. An outside interface is one on which you can use only public IP addresses. Usually, the Ethernet interfacesis the inside interface and the PPP interface is the outside interface.

The direction of an interface is used to determine how to apply NAT rules. The basic, napt, filter, and pass rules manage the address translation for connections initiated from the inside interface and destined for the outside interface. These rules also translate the responses coming from outside to inside. The *rdr* rules manage address translations for connections initiated from the outside and destined for the inside interface. The *bimap* rule is unique because it works in both directions. You can use it for inside to outside as well as outside to inside connections. The *pass* rule is helpful when you want specific inside to outside connections passed through the unit without any change.

A connection, as used here, is simply a network access from one side of the unit to the other. A web browser on the LAN that accesses a web site on the WAN would be an inside to outside connection.

**Configuring the NAT *Direction***

You can specify the NAT direction of these interfaces directly, as part of the corresponding **create** command. By default, the LAN interfaces (Ethernet) have the direction as **inside** while the WAN interfaces (PPP) have the direction as **outside**.

For example, to configure the NAT direction, while creating the Ethernet interface, specify the direction as **inside**.

```
$ create ethernet intf ifname eth-0 ip 192.168.1.1 mask 255.255.255.0
inside
```

### 9.1.3 The napt rule

A Network Address and Port Translation (napt) rule translates the source address as well as the port number for inside to outside connections.

**Creating an napt Rule with a Rule ID**

- To create a napt rule, type the command:

```
$ create nat rule entry napt ruleid 1
```

Although the command takes quite a few parameters, the default values suffice in most cases. Note that the rule is assigned a rule ID. Rules are scanned in order of their rule IDs, lowest to highest. When the packet data does not match a rule, the next rule is tested for a match. You should always create NAT rules with gaps in the rule IDs so that you can add a new rule that you want to be applied between two existing rules. For instance, if you initially create two rules with rule IDs 10 and 20, then later create another with rule ID 15, this new rule would be applied after 10 but before 20. But if you had created the first two rules with rule IDs 10 and 11, then this sequence would not be possible without first deleting one of the rules.

### Viewing NAT Rules

- To see the existing NAT rules, type the command:

```
$ get nat rule entry
```

You can specify a rule ID to view a specific rule:

```
$ get nat rule entry ruleid 1
```

### Creating an napt Rule with All Parameters

- The following command includes all parameters for creating an napt rule:

```
$ create nat rule entry ruleid 1 napt ifname ppp-0 lcladdrfrom
0.0.0.0 lcladdrto 255.255.255.255 glbaddrfrom 0.0.0.0
glbaddrto 0.0.0.0
```

The **ifname** parameter says that the rule is applicable to the **ppp-0** interface. If you do not specify the interface, NAT assumes that the rule is applicable on all outside interfaces.

The **lcladdrfrom** and **lcladdrto** parameters tell NAT which outgoing packets to translate. If the source address in a packet lies within the range specified by these two parameters, the match is considered successful and the packet is translated using this rule. Setting **lcladdrfrom** to 0.0.0.0 and **lcladdrto** to 255.255.255.255 indicates that rule will be matched for all packets going out on the interface.

Suppose you have two subnets on the LAN: 192.168.1 and 172.25, and you want NAT to work for only one of them - 192.168.1 (if for example, the other subnet never needs to access the Internet). Then, setting **lcladdrfrom** to 192.168.1.0 and **lcladdrto** to 192.168.1.255 will indicate that the translation is required only if the first three numbers in the source IP address are 192.168.1.

The **glbaddrfrom** and **glbaddrto** parameters tell NAT what the source addresses should be translated into. If the ISP provides a fixed IP address, such as 202.1.1.1, then setting both **glbaddrfrom** and **glbaddrto** to 202.1.1.1 will specify that the source address should be translated into 202.1.1.1. Setting both to 0.0.0.0 indicates that you want the translated address to be the address of the outgoing interface. This is used when the IP address of the interface is not fixed. If **glbaddrfrom** and **glbaddrto** are not the same, i.e., if they indicate a range of IP addresses, then the translated address is chosen from among this range on a per connection basis.

### Deleting a Rule

- To delete a rule use the delete command and specify the rule ID:

```
$ delete nat rule entry ruleid 1
```

#### 9.1.3.1 Configuring Port Numbers for napt

A napt rule translates the source IP address as well as the source port number. The IP address to be used for translation is picked up from the NAT rule parameters. The port numbers for translations are specified using the **portstart** and **portend** parameters in the **modify nat global** command. These parameters can be modified only when NAT is disabled.

### 9.1.4 The rdr Rule

The rdr, or redirect, rule enables you to route incoming connection requests referencing your public IP address and a specific port number to a private IP addresses and port number on your LAN.

Suppose there is a web server installed on one of the LAN hosts. If someone from outside tries to connect to this server, the connection request will arrive on the unit with the same destination address as the public IP address, 202.1.1.1. However, the web server on the LAN has the address, say 192.168.1.3. NAT allows you to forward such connections to the correct internal destination using the rdr rule.

**Creating an RDR Rule**

```
$ create nat rule entry ruleid 2 rdr destportfrom num 80
destportto num 80 lcladdrfrom 192.168.1.3 lcladdrto
192.168.1.3
```

This command indicates that in all connection requests from the WAN side for the port number 80, which is the well known port number for a web server, the destination address should be substituted by 192.168.1.3. The incoming connection thus gets forwarded to the correct server. Suppose multiple web servers exist on the LAN with IP addresses ranging from 192.168.1.3 to 192.168.1.6 and you need to distribute the load of incoming connections between these. The rdr rule allows you to do this by specifying the range using the **lcladdr** parameters. Setting **lcladdrfrom** to 192.168.1.3 and **lcladdrto** to 192.168.1.6 will distribute the load in a round-robin fashion.

Similarly, you can add rdr rules for other servers on the LAN, which are to be exposed to the outside world. Each server will typically have a well-known port number, which you can specify as the **destportfrom** and **destportto** parameters. These two parameters need not be the same. They can also be a range.

The rdr rule also enables you to substitute the port number in the incoming request. If the web server is running on some non-standard port number, say 8080, create the rule as:

```
$ create nat rule entry ruleid 2 rdr destportfrom num 80
destportto num 80 lcladdrfrom 192.168.1.3 lcladdrto
192.168.1.3 lclport num 8080
```

This will not only forward the request to 192.168.1.3 but also modify the port number to what the server expects.

As with the napt rule, you can specify an interface name, say ppp-0.

- To specify an interface name, type the command:

```
$ create nat rule entry ruleid 2 rdr ifname ppp-0 destportfrom
num 80 destportto num 80 lcladdrfrom 192.168.1.3 lcladdrto
192.168.1.3 lclport num 8080
```

This will make the rule operate only on requests received on the ppp-0 interface. Not specifying the interface makes the rule applicable on all outside interfaces. Further, if you have a fixed IP address, you can specify that as well.

- To specify an interface and a fixed IP address, type the command:

```
$ create nat rule entry ruleid 2 rdr ifname ppp-0 destaddrfrom
202.1.1.1 destaddrto 202.1.1.1 destportfrom num 80 destportto
num 80 lcladdrfrom 192.168.1.3 lcladdrto 192.168.1.3 lclport
num 8080
```

This will translate the request only if it contains the destination address 202.1.1.1.

### 9.1.5 The basic and filter Rules

The napt rule discussed earlier translates not only the source addresses but also the port numbers in outgoing connections. The unit also provides a more basic functionality in which only the addresses are translated and the port numbers remain unchanged. The basic and filter rules fall in this category.

With the basic rule you can translate a range of IP addresses, specified by **lcladdrfrom** and **lcladdrto**, into another group of addresses, specified by **glbaddrfrom** and **glbaddrto**.

**Creating a basic Rule**

- To translate a group of IP addresses, use the command:

```
$ create nat rule entry ruleid 3 basic lcladdrfrom 192.168.1.10
lcladdrto 192.168.1.20 glbaddrfrom 202.1.1.1 glbaddrto
202.1.1.5
```

This will translate source addresses in the range 192.168.1.10 to 192.168.1.20 into the range 202.1.1.1 to 202.1.1.5.

The filter rule extends your control over NAT as it allows translation based on the source address and the destination address and port numbers.

**Creating a filter Rule**

- For instance, with the following command, all accesses from the given local addresses to the web server on 202.64.2.5 will appear to originate from 202.1.1.2:

```
$ create nat rule entry ruleid 4 filter lcladdrfrom 192.168.1.0
lcladdrto 192.168.1.10 destaddrfrom 202.64.2.5 destaddrto
202.64.2.5 destportfrom num 80 destportto num 80 glbaddrfrom
202.1.1.2 glbaddrto 202.1.1.2
```

### 9.1.6 The bimap Rule

Suppose you want to provide a one-to-one mapping between one of the public IP addresses and one of the LAN hosts. All accesses to the public IP address should be forwarded to the particular LAN host, and all accesses from the host should appear to go out from that public IP address only. If you were to use the rules discussed so far, this kind of a situation would require you to create two rules, one to handle the inside to outside translations, and another to handle the outside to inside translations. The bimap rule simplifies this kind of a situation by enabling a two-way translation with a single rule.

**Creating a bimap Rule**

- To enable a two-way translation with a single rule, type the command:

```
$ create nat rule entry ruleid 6 bimap lcladdrfrom 192.168.1.3
lcladdrto 192.168.1.3 glbaddrfrom 202.1.1.1 glbaddrto
202.1.1.1
```

This one rule will suffice to translate the source address in all outgoing accesses from 192.168.1.3 to 202.1.1.1 and also to translate the destination address in all incoming accesses on 202.1.1.1 to 192.168.1.3.

### 9.1.7 The pass Rule

The pass rule allows you to let connections from a range of inside addresses go through without getting translated. For instance the following rule lets all connections from the given range go through to the WAN without any translation:

**Creating a pass Rule**

```
$ create nat rule entry ruleid 7 pass lcladdrfrom 192.168.1.10
lcladdrto 192.168.1.20
```

### 9.1.8 Configuring ALGs

You will need to configure an *Application Level Gateway (ALG)* if you want to use certain applications such as FTP, SNMP, Real Audio, and a few others across the unit. For instance, if you want to ftp a file from some outside host, or you are providing an ftp server that has to be accessible to outside users, you will need to configure an ALG for FTP.

An ALG enables the unit to carry out address translations in the entire packet instead of just the packet headers. The mentioned applications need ALGs since they use IP addresses in their payloads also. Most other applications do not.

- To be able to access external ftp servers from the LAN, type the command:

```
$ create alg port portno 21 algtype ftp
```

This enables the FTP ALG on all connections having the port number 21. The port number would, in most cases be the well-known port number of the application. For instance, 21 is the well known port number for an ftp server.

The command also provides a protocol parameter using which you can enable the ALG selectively on packets carrying a particular protocol. By default, this parameter it is set to **any**, so that all protocols undergo translation (this setting usually suffices in normal usage).

**NOTE:**

The actual translation occurs in accordance with the corresponding NAT rules. The ALG just enables NAT to translate inside the payloads of these specific applications.

**Deleting ALGs**

- To delete a configured ALG, type the command:

```
$ delete alg port portno 21 prot any
```

This will disable the additional processing required for accessing external ftp servers.

**Viewing ALGs**

- To see the list of supported ALGs, type the command:

```
$ get alg type
```

- To see all the configured ALGs, type the command:

```
$ get alg port
```

#### 9.1.8.1 Configuring unit for IPSec traffic pass through

IPSec is a mechanism to provide various security services for traffic at the IP layer. IPSec protects IP datagram by defining a method of specifying the traffic to protect, and how that traffic is to be protected. The method of protecting IP datagrams or upper layer protocols is by using one of the IPSec protocols, the Encapsulating Security Payload (ESP) or Authentication Header (AH). To properly encapsulate and de-capsulate IPSec packets it is necessary for the communicating peers to agree on the security parameters (e.g. keys to use for encrypting/decrypting etc.). Such a construct is called a Security Association (SA).

When NAT is enabled on the unit, IPSec traffic pass through is supported under certain conditions. IPSec pass through does not mean the unit can originate/terminate IPSec sessions (from/to the unit). This means that two more NAT ALGs called IKE ALG and ESP ALG are enabled on the unit, to allow the IPSec traffic to pass through transparently.

##### 9.1.8.1.1 Configuration details

IPSec Pass through is required when the unit is running in routing mode with NAT enabled. The normal customer setup includes VPN client running on LAN PCs behind the unit, trying to connect to some VPN server on the Internet. For example, a telecommuter accessing the corporate Network through VPN from home.

**NOTE:**

Note: When the SAR130 unit is in bridging or ZIPB mode, no extra configuration is required for any kind IPSec traffic to pass through (as NAT is not running on the unit).

By default, the IKE and ESP ALGs are created.

- To confirm if the IKE and ESP ALGs are configured on the unit, enter:

```
$ get alg port
```

You should see IKE and ESP ALGs configured.

- Normally, only one NAPT rule is sufficient to access the internet. But with IPSec pass through, customers will be accessing a VPN server as well as Internet in parallel. So following NAT rule configuration is recommended :

```
$ create nat rule entry ruleid 1 filter destportfrom num 500
destportto num 500 prot udp
$ create nat rule entry ruleid 2 napt
```

In the above command set, the first rule indicates if the protocol is UDP and destination port number is 500 (that is, IKE port number). Then, the IP address will be translated, but not the port numbers (similar to basic rule). This rule will be used for IKE ALG to translate IKE packets (when automatic keying is used by VPN clients).

The second rule indicates NAPT should be used if the first rule is not hit, which will be used for accessing internet.

#### 9.1.8.1.2 Restrictions

IPSec pass through only takes care of passing ESP tunnel mode traffic. The pass through of the following, is not supported:

- Authentication Header (AH) in tunnel mode or transport mode, or with UDP encapsulated. The reason this traffic cannot pass through is because AH header authenticates IP address in IP header also. So, intermediate NAT routers cannot translate the IP address.
- Encapsulating Security Payload (ESP) in transport mode with TCP packets. The reason TCP traffic with ESP header in transport mode cannot pass through, is TCP header pseudo checksum is computed using IP address and as the TCP header is encrypted when it reaches intermediate NAT router. NAT cannot translate the IP address.

**NOTE:**

Normally L2TP over IPSec transport is used by Window-based VPN client (and as L2TP uses UDP), and it can pass through.

- Although multiple VPN clients on the LAN side are supported, only one VPN client should be involved in IKE negotiations, at a time. This limitation exists because, only the **basic** NAT flavor can be enabled for IKE traffic. **NAPT** would change the port, whereas, IKE requires the source and destination ports to be only 500.
- If the VPN traffic carries some protocols, which require ALGs (because they carry IP addresses in their payload), then, those protocols will not pass through properly. This is because, VPN traffic is encrypted. Hence, no ALGs can be applied to them.

### 9.1.9 Enabling NAT

After you have configured the required NAT rules and ALGs, you need to enable NAT.

- NAT can be enabled or disabled using the following command.

```
$ modify nat global [enable/disable]
```

**NOTE:**

NAT works only on the new session which gets created after enabling it, and will not affect the ongoing sessions which have been initiated *before hand.*

## 9.2 Firewall

This chapter describes the firewall feature of the product. This feature protects the modem and the LAN hosts behind the modem from malicious attacks originating from the "unfriendly outside world" WAN hosts. Various kinds of attacks are known that can cause disruption in regular service for hosts behind the modem, or cause harm to internal LAN hosts. This feature detects and protects against such common attacks and reports such attacks to the network administrator, for appropriate action.

### 9.2.1 Attack protection

Typically, attacks from the WAN side exploit deficiencies in the OS and implementation problems.

• Usage of malicious IP address - These attacks exploit usage of source IP address that are illegal on a given interface. Some examples of illegal packets are, sending packets with source IP address as

- internal addresses on a public interface
- loopback address on any interface
- network broadcast address on any interface
- IP broadcast address on any interface
- destination address in the same packet on any interface
- Multicast address on any interface.

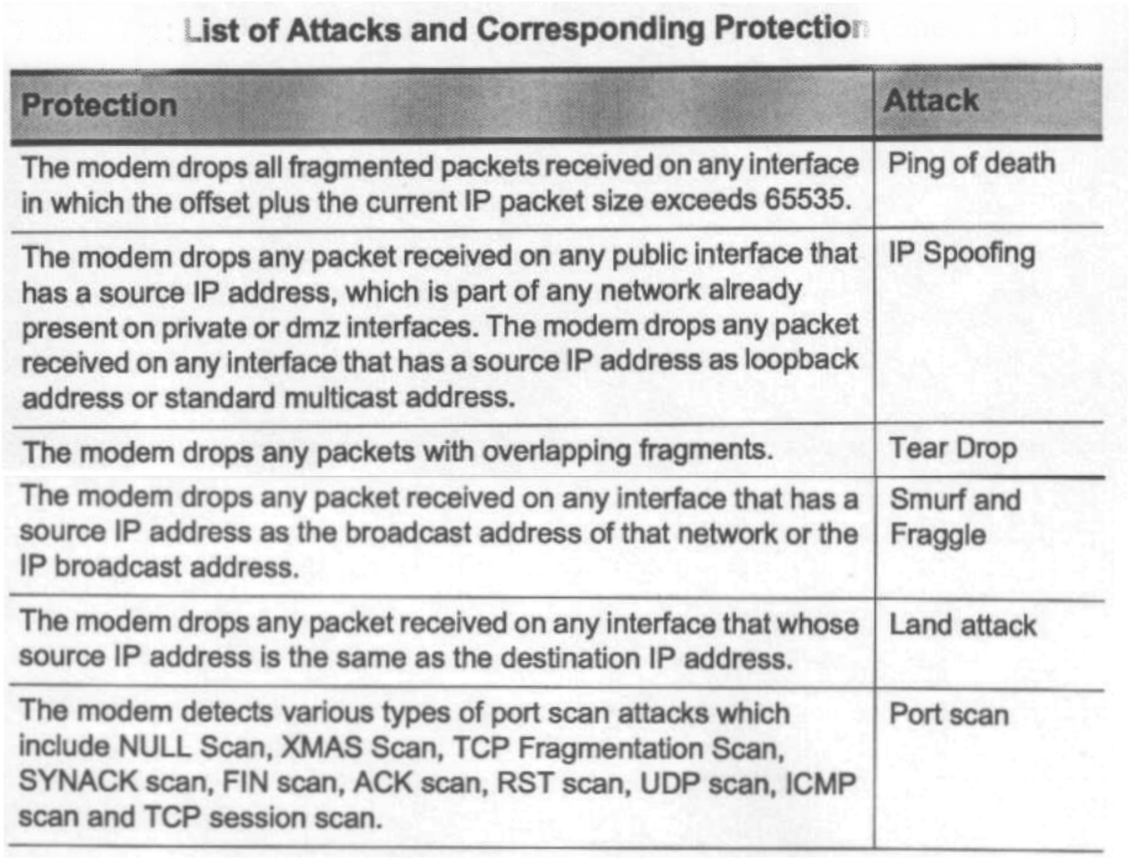
• The network can also be potentially flooded if packets are being sent to

- network broadcast address on any interface
- IP broadcast address on any interface.

• Hackers exploit OS vulnerabilities by sending packets with

- IP length greater than the one specified by the standards
- overlapping fragments.

### 9.2.1.1 List of Attacks



List of Attacks and Corresponding Protectio

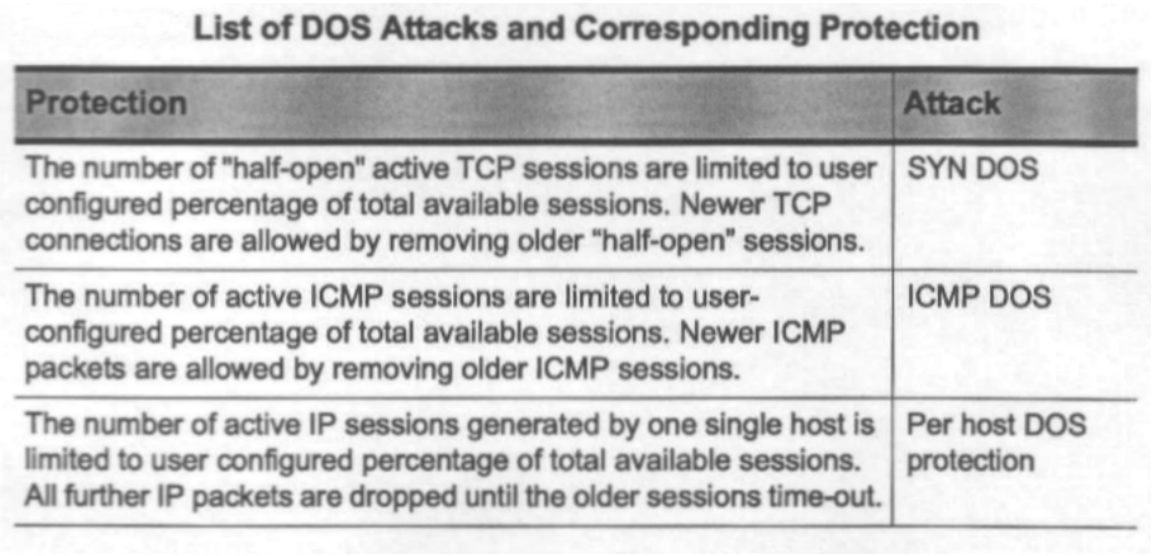
| Protection | Attack |
|---|---|
| The modem drops all fragmented packets received on any interface in which the offset plus the current IP packet size exceeds 65535. | Ping of death |
| The modem drops any packet received on any public interface that has a source IP address, which is part of any network already present on private or dmz interfaces. The modem drops any packet received on any interface that has a source IP address as loopback address or standard multicast address. | IP Spoofing |
| The modem drops any packets with overlapping fragments. | Tear Drop |
| The modem drops any packet received on any interface that has a source IP address as the broadcast address of that network or the IP broadcast address. | Smurf and Fraggle |
| The modem drops any packet received on any interface that whose source IP address is the same as the destination IP address. | Land attack |
| The modem detects various types of port scan attacks which include NULL Scan, XMAS Scan, TCP Fragmentation Scan, SYNACK scan, FIN scan, ACK scan, RST scan, UDP scan, ICMP scan and TCP session scan. | Port scan |

Table above provides a the list of attacks against which the modem provides protection, through its firewall feature.

### 9.2.1.2 Denial of service (DOS) protection

Flooding the modem with large number of packets, causing all the resources to be utilized causes denial of service to genuine connections. DOS protection works by enforcing limits on various types of IP sessions that can pass through the modem. They are,

- Half open TCP connections
- ICMP sessions
- Number of connections from a single host.



List of DOS Attacks and Corresponding Protection

| Protection | Attack |
|---|---|
| The number of "half-open" active TCP sessions are limited to user configured percentage of total available sessions. Newer TCP connections are allowed by removing older "half-open" sessions. | SYN DOS |
| The number of active ICMP sessions are limited to user-configured percentage of total available sessions. Newer ICMP packets are allowed by removing older ICMP sessions. | ICMP DOS |
| The number of active IP sessions generated by one single host is limited to user configured percentage of total available sessions. All further IP packets are dropped until the older sessions time-out. | Per host DOS protection |

Table above provides a the list of DOS attacks against which the modem provides protection, through its firewall feature.

#### 9.2.1.3 Service protection

The firewall provides a mechanism to block certain services or protocols that may be misused by hackers. Using IP Filter rules, the modem restricts access to services for only a set of WAN hosts.The modem uses basic IP filter rules to provide this functionality. These IP filter rules also use the type of interface in determining whether a packet has to be blocked or not. The firewall provides a mechanism in which individual IP filter rules can be marked as whether they are part of a high, medium or low security level. The level of firewall security policy can be configured by the user at runtime. This makes user configuration simple and easy.

The modem drops any packet received on any interface that has a source IP address as the broadcast address of that network or the IP broadcast address.

**Smurf and Fraggle**

The modem drops any packet received on any interface that whose source IP address is the same as the destination IP address.

**Land attack**

The modem detects various types of port scan attacks which include NULL Scan, XMAS Scan, TCP Fragmentation Scan, SYNACK scan, FIN scan, ACK scan, RST scan, UDP scan, ICMP scan and TCP session scan.

**Port scan**

An IP filter rule can be configured to be active at one or more security levels. So the set of all rules configured to be active at the High security level determine the filtering support being provided at High. The same holds for Medium and Low levels. When the security level is set to None, no IP filter rules are active, implying zero protection.

The default configuration provided with the modem contains IP filter rules to cater to typical user requirements. The private side of the network is the most secure. Most accesses from private to other interfaces are allowed, but accesses to the private side are restricted.

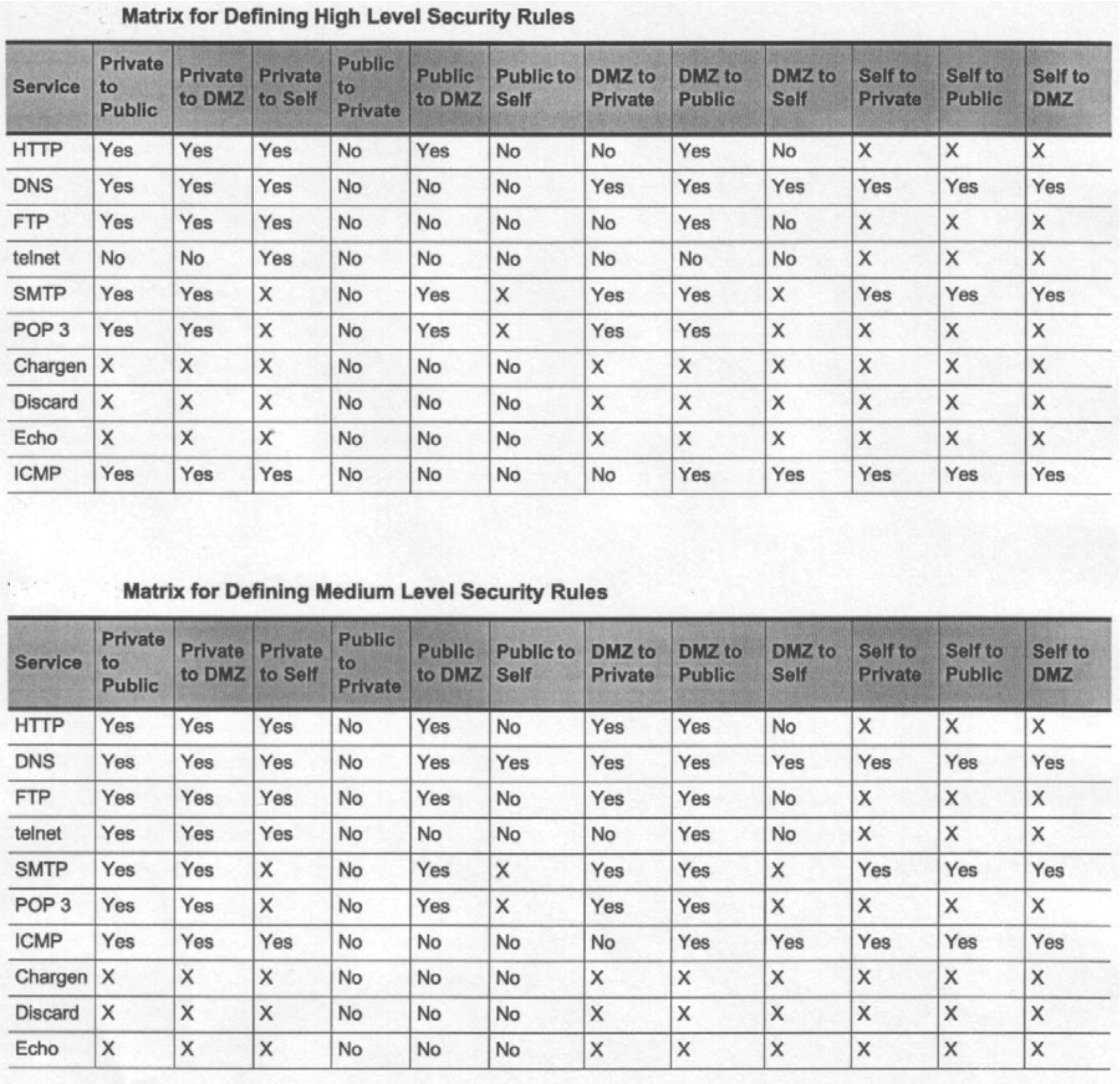
The demilitarized (dmz) side of the network is more accessible, since the publicly visible services, such as web servers, are expected to be hosted on the dmz side. Checks on accesses from the dmz side to the private side are less stringent as compared to those from the public side, since the dmz side is more "trusted".

Also, as a general rule, as the security level decreases from high to low, more services are made accessible. At High, only the most essential services are accessible. Services such as ICMP, which could be easily misused by intruders, are typically not allowed at High security level. For instance, HTTP access from the public to dmz side is allowed at all levels, but ICMP access is allowed only at Low. All accesses from self to any side are always allowed since the modem is expected to be a trusted host on all sides.

The following matrices are used to define default IP filter rules for high, medium and low security.

**NOTE:**

X indicates that no rules are configured to take care of the particular case, since that service does not exist .



**Matrix for Defining High Level Security Rules**

| Service | Private to Public | Private to DMZ | Private to Self | Public to Private | Public to DMZ | Public to Self | DMZ to Private | DMZ to Public | DMZ to Self | Self to Private | Self to Public | Self to DMZ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| HTTP | Yes | Yes | Yes | No | Yes | No | No | Yes | No | X | X | X |
| DNS | Yes | Yes | Yes | No | No | No | Yes | Yes | Yes | Yes | Yes | Yes |
| FTP | Yes | Yes | Yes | No | No | No | No | Yes | No | X | X | X |
| telnet | No | No | Yes | No | No | No | No | No | No | X | X | X |
| SMTP | Yes | Yes | X | No | Yes | X | Yes | Yes | X | Yes | Yes | Yes |
| POP 3 | Yes | Yes | X | No | Yes | X | Yes | Yes | X | X | X | X |
| Chargen | X | X | X | No | No | No | X | X | X | X | X | X |
| Discard | X | X | X | No | No | No | X | X | X | X | X | X |
| Echo | X | X | X | No | No | No | X | X | X | X | X | X |
| ICMP | Yes | Yes | Yes | No | No | No | No | Yes | Yes | Yes | Yes | Yes |

**Matrix for Defining Medium Level Security Rules**

| Service | Private to Public | Private to DMZ | Private to Self | Public to Private | Public to DMZ | Public to Self | DMZ to Private | DMZ to Public | DMZ to Self | Self to Private | Self to Public | Self to DMZ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| HTTP | Yes | Yes | Yes | No | Yes | No | Yes | Yes | No | X | X | X |
| DNS | Yes | Yes | Yes | No | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| FTP | Yes | Yes | Yes | No | Yes | No | Yes | Yes | No | X | X | X |
| telnet | Yes | Yes | Yes | No | No | No | No | Yes | No | X | X | X |
| SMTP | Yes | Yes | X | No | Yes | X | Yes | Yes | X | Yes | Yes | Yes |
| POP 3 | Yes | Yes | X | No | Yes | X | Yes | Yes | X | X | X | X |
| ICMP | Yes | Yes | Yes | No | No | No | No | Yes | Yes | Yes | Yes | Yes |
| Chargen | X | X | X | No | No | No | X | X | X | X | X | X |
| Discard | X | X | X | No | No | No | X | X | X | X | X | X |
| Echo | X | X | X | No | No | No | X | X | X | X | X | X |

**NOTE:**

In addition to the above, network and broadcast accesses from Public are denied at High security.

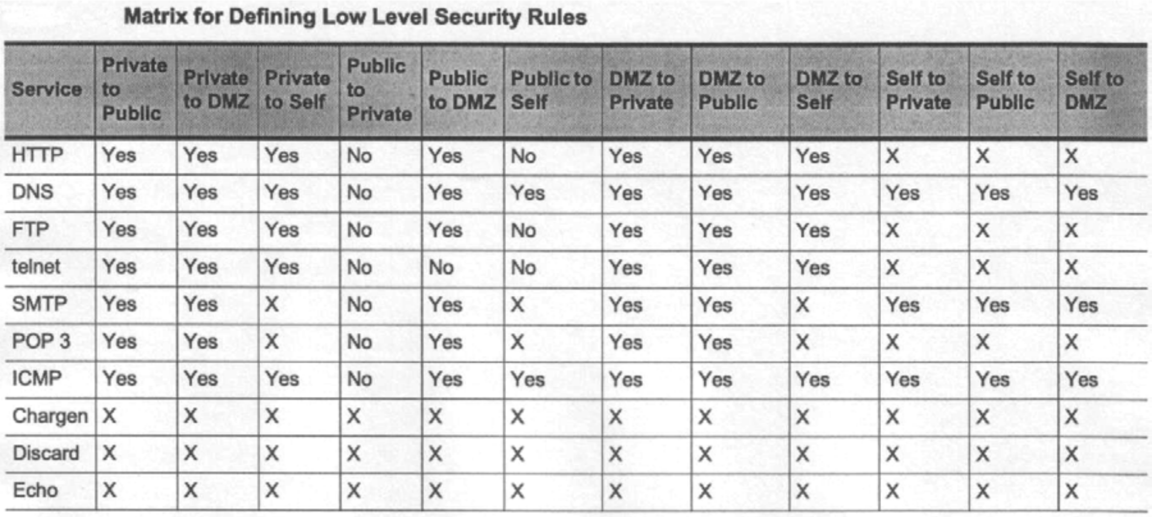
### 9.2.2 Firewall features

The powerful features of Logging and Blacklisting enable the modem to provide greater protection.

**IP session - overview**

The modem's IP filtering feature allows it to examine each packet traveling in either direction (incoming or outgoing) on an interface and to filter out packets based on rules that you define. On finding a matching rule, the modem remembers the matching rule on the particular interface in the particular direction, along with the unique combination of protocol and source/destination addresses and ports. This unique combination of protocol and source/destination addresses and ports is defined as an **IP session**.

An IP session is kept alive by packets flowing in either direction. A session can time out if no packets belonging to the session are encountered for some time. Thereafter, any packets with the same session parameters cause a new session to be formed. Rule lookups for the new session begin afresh. The session time outs depend on particular protocols and, in case of TCP, on the state of the TCP connection as well. The modem keeps track of TCP connections flowing from or through it. It also keeps track of any UDP, ICMP or other protocol messages exchanged between peers for traffic flowing through it. The network administrator is therefore also able to view all active sessions passing through the modem.



**Matrix for Defining Low Level Security Rules**

| Service | Private to Public | Private to DMZ | Private to Self | Public to Private | Public to DMZ | Public to Self | DMZ to Private | DMZ to Public | DMZ to Self | Self to Private | Self to Public | Self to DMZ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| HTTP | Yes | Yes | Yes | No | Yes | No | Yes | Yes | Yes | X | X | X |
| DNS | Yes | Yes | Yes | No | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| FTP | Yes | Yes | Yes | No | Yes | No | Yes | Yes | Yes | X | X | X |
| telnet | Yes | Yes | Yes | No | No | No | Yes | Yes | Yes | X | X | X |
| SMTP | Yes | Yes | X | No | Yes | X | Yes | Yes | X | Yes | Yes | Yes |
| POP 3 | Yes | Yes | X | No | Yes | X | Yes | Yes | X | X | X | X |
| ICMP | Yes | Yes | Yes | No | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| Chargen | X | X | X | X | X | X | X | X | X | X | X | X |
| Discard | X | X | X | X | X | X | X | X | X | X | X | X |
| Echo | X | X | X | X | X | X | X | X | X | X | X | X |

#### 9.2.2.1 Logging

The firewall provides alerts to the network administrator through mails, system traces. These alerts inform the network administrator of detection of any of the attacks mentioned above.

The system does not generate a log every time an attack is detected. This holds good for some port scan attacks such as SYNACK scans, FIN scans, ACK scans, RST scans, UDP scans, ICMP scans and TCP session scan attacks. The logs are generated at a certain periodicity, which can be configured (default 30 minutes). For such type of scans, the system maintains counters on how many times a particular attack has been detected during the last logging period. It also reports when the log is generated.

### 9.2.3 Blacklisting

The network administrator may not always be available to read the mails sent as warning. To protect users during this time, firewall provides blacklisting, a mechanism by which the initiator of an attack is blacklisted so that, for some time, no traffic is allowed from that malicious host. No packets from a host from which a violation has been detected is allowed to pass through, until a user configured time.

Blacklisting is provided against the following attacks:

- tear drop
- ping of death

- port scan.

Immediate blacklisting is done for attacks when the system is sure, after examining the packet, that it has not arrived under normal circumstances. This holds good in case of single packet scans such as NULL Scan, XMAS Scan and TCP Fragmentation Scan.

### 9.2.4 Configuration details

The firewall feature of the modem provides the user with the ability to be alerted of violations, over e-mail. To avail of this benefit, the user needs to configure the IP address or fully qualified domain name (FQDN) of his mail server. This mail server can be on the LAN or can be provided by the ISP.

**NOTE:**

You need to check the IP Filter rules to ensure that the connectivity to the configured SMTP server exists.

- Use the **`modify smtp servaddr`** command to configure the IP address of the SMTP server.
- To modify the global configuration of the IP firewall, enter:

```
modify fwl global [attackprotect enable|disable][dosprotect
enable|disable][blistprotect enable|disable][blistperiod
<decvalue>][maxtcpconn <decvalue>] [maxicmpconn
<decvalue>][maxsinglehostconn <decvalue>][logdest
email|trace|both|none] [email1 email-id][email2 emailid][
email3 email-id] [minlogtime <decvalue>]
```

- Use the **`emailid`** parameter of the **`modify fwl global`** command to specify up to three email ids to which the alerts will be mailed.
- Use the **`attackprotect enable|disable`** parameter of the **`modify fwl global`** command to disable or enable attack protection.
- Use the **`dosprotect enable|disable`** parameter of the **`modify fwl global`** command to disable or enable DOS protection.
- Use the **`get fwl stats`** command to get firewall statistics such as the number of ICMP sessions, the number of Half open TCP Sessions, the type of attack, the time stamp, and other elated details. Use the **`reset firewall stats`** command to reset firewall statistics.
- Use the **`minlogtime`** parameter to specify the minimum logging time between the mailing of logs that inform of attacks.The next log is generated only after the minimum log time, specified in minutes, elapses, and the system detects another attack.

**`Getting information on blacklisted host`**

- To get information on a blacklisted host, enter:

```
get fwl blacklist [ip <ddd.ddd.ddd.ddd>]
```

This command displays

- the IP address of the blacklisted host

- the reason for blacklisting the host

- the IP Filter rule id which caused the blacklisting (valid only if blacklisted due to service protection violation)

- the time duration in seconds after which the IP address entry will be removed from this table.

**Deleting a Blacklisted host**

- To delete a blacklisted host, enter:

```
delete fwl blacklist ip <ddd.ddd.ddd.ddd>
```

**Enabling or disabling Blacklisting**

Using relevant parameters of the **modify fwl global** command described above, you can enable or disable blacklisting and configure the duration for blacklisting an attacking host. You can also configure the percentage of total connections that can be in a TCP half open state, the percentage of total connections that can be ICMP connections and the maximum percentage of connections from a single host, using the relevant parameters of the **modify fwl global** command.

**NOTE:**

Firewall works only on the new session which gets created after enabling it, and will not affect the ongoing sessions which have been initiated before the firewall was enabled.

## 9.3 Filtering and IP Sessions

This section provides details about the SAR130 unit’s IP filtering capability and how to configure the rules for IP filtering.

The SAR130 unit's IP filtering feature allows it to examine each packet traveling in either direction (incoming or outgoing) on an interface and to filter out packets based on rules that you define.

**NOTE:**

Because the IP filter scans packets at the IP layer, it can be used only in the routing mode.

A rule can be configured to be applicable on a specific interface or on all interfaces but it applies only in one direction (in or out).

However, this does not hold good for rules with "storestate" feature as they get applied in both directions

Each rule is assigned an ID. Rule IDs must be unique. These IDs determine the order in which rules are matched - from lowest to highest number.

**NOTE:**

To allow you to retain full control over the order of rule evaluation, do not number rules/subrules consecutively, e.g., 1, 2, 3, etc., but in increments, e.g., 10, 20, 30, etc. This will allow you to insert more rules between the existing ones at a later time. If you number your rules consecutively, you will have to delete and recreate all existing rules that are to follow the new rule.

With IP filtering enabled, when a packet is received on an interface, the unit scans the rules applicable on that interface for the incoming direction. When a packet is sent out from an interface, all rules applicable on that interface for the outgoing direction are scanned. If a rule is found that matches the packet, the packet is accepted or denied as specified by the rule. A packet in transit from the LAN to the WAN, or from the WAN to the LAN thus needs to be cleared by two sets of rules - one on the incoming interface and one on the outgoing interface.

A rule is said to match the packet only if all the selectors specified in the rule match the fields in the packet. The selectors supported are - source and destination IP addresses, transport protocol, source and destination port numbers, ICMP type and code values, the TCP syn flag, filtering based on packet length, filtering based on presence of any IP options and filtering of all fragmented packets. If a rule does not match the packet, that rule is skipped and a match is tried with the next rule.

If none of the rules match the packet, the default action is taken for that packet.

The default action is specified as part of the IP filter global configuration.

On finding a matching rule, the unit remembers the matching rule on the particular interface in the particular direction, along with the unique combination of protocol and source/destination addresses and ports. This unique combination of protocol and source/destination addresses and ports is defined as an IP session.

Now, when another packet with the same session parameters is encountered, the unit does not need to look up its rules once again It simply applies the remembered rule for the particular interface and direction.

A single IP session has four rules associated with it - one each for the incoming/outgoing directions on the incoming interface and one for each direction on the outgoing interface.

All four rules are remembered as part of the IP session.

At least one packet must flow in either direction before all four rules are determined. For instance, if a LAN client does a telnet to a WAN host, the first packet from the client to the server would enable the unit to determine two rules applicable for the session - one for the incoming direction on, say eth-0, and one for the outgoing direction on, say ppp-0.

The first response packet that comes from the server to the client will enable the unit to determine the remaining two rules for the session - one for the incoming direction on ppp-0 and one for the outgoing direction on eth-0.

If no matching rules are found for a given direction on a given interface, the unit also remembers to apply the default action for subsequent packets for the particular interface and direction.

An IP session is kept alive by packets flowing in either direction. A session can time out if no packets belonging to the session are encountered for some time.

Thereafter, any packets with the same session parameters cause a new session to be formed. rule lookups for the new session begin afresh. The session time outs depend on particular protocols and, in case of TCP, on the state of the TCP connection as well. The time outs can be viewed using the **get nat global** command.

The maximum number of rules is determined by the **maxipfrule** parameter in the **size** command.

### 9.3.1 Using IP Filtering Rules

Rules specify, at a minimum, the interface and direction to be monitored, and the action to take if a packet matches the rule.

**Commands for rules.** The basic commands used to create, modify and delete IP filter rules are described below.

#### Creating an IP filter rule

- To create an IP filter rule, enter:

```
$ create ipf rule entry ruleid 10 ifname ppp-0 dir in transprot
eq icmp act deny enable
```

This command creates a rule with rule ID 10 on interface ppp-0, for packets traveling in the incoming direction. Omitting the ifname parameter makes the rule applicable on all interfaces. If the protocol field in the IP packet equals ICMP, this rule drops the packet (since the action is deny). Hence, the rule drops all incoming ICMP packets on ppp-0. The enable indicates that the rule is to be created in the enabled state (the default being disable).

If the direction is set to out, you can also specify an In Interface name using the **inifname** parameter. Thus, the following rule will match only those packets that arrived on the unit on eth-0 and are leaving via ppp-0 -

```
$ create ipf rule entry ruleid 10 ifname ppp-0 dir out inifname
eth-0 transprot eq icmp act deny enable
```

Various options for matching allow you to look for addresses equal to a value, not equal to a value, within a given range of values, outside a given range of values and so on. Similar options are also supported for ports. Following is an example.

```
$ create ipf rule entry ruleid 20 ifname ppp-0 dir out transprot
eq tcp srcaddr range 192.168.1.2 192.168.1.10 destaddr erange
202.1.1.20 202.1.1.30 act deny enable
```

This discards all TCP packets originating from addresses 192.168.1.2 to 192.168.1.10 (both inclusive) and destined for addresses other than those lying in the range 202.1.1.20 to 202.1.1.30 (both inclusive). This means, packets destined for 202.1.1.20 and 202.1.1.30 will be allowed by this rule. Only destinations less than 202.1.1.20 or greater than 202.1.1.30 will be denied.

You can also log the packets matching a given rule using the log option:

```
$ create ipf rule entry ruleid 10 ifname ppp-0 dir in transprot
eq icmp act deny enable log enable
```

The IP filter logs are governed by firewall logging. The logs can be directed to trace-based destinations or can be sent as e-mails using the **modify fwl global** commands.

**Modifying an IP filter rule**

- To modify a rule, enter:

```
$ modify ipf rule entry ruleid 10 accept
```

**Deleting an IP filter rule**

- To delete a rule, enter:

```
$ delete ipf rule entry ruleid 10
```

**Viewing all rule configurations**

- To see the configuration of all rules, enter:

```
$ get ipf rule entry
```

#### 9.3.1.1 IP filter rule configuration for enhanced security

- Predefined IP Filter rules enable you to set the levels of security as High, medium, Low or None, for the modem.

- IP Filter rules are configurable such that they are enabled or disabled depending upon time of the day.

- You can use the interface type as a selector, while configuring IP Filter rules The interface types you can choose between are, "public", "private" or "dmz".

- You can configure IP Filter rules to filter out all fragmented packets and packets with IP options.

- Each IP Filter rule has a 16 character long string as parameter. This string can be used for logging. Typically, the string is the name of the attack detected.

- IP Filter rules can take port ranges as service names (like HTTP etc.) Some standard port numbers, as mentioned in the list below, are used for the following service names, irrespective of the transport protocol selected.

- Echo 7
- Discard 9
- CHARGEN 19
- FTP 21
- TELNET 23
- SMTP 25
- DNS 53
- BOOTP 67
- TFTP 69
- HTTP 80
- POP3 110

- SNMP 161

• IP Filter rules can take a keyword "netbcast", with the **destaddr** parameter, to mean the broadcast address of the network on which the packet was received. This can be used to match packets with the destination as the network broadcast address.

• The keyword “self” can be given with **scraddr** and **destaddr** parameters to match packets generated by the modem, or destined for the modem.

• IP Filter rules can be configured to indicate whether the host generating the packet should be blacklisted or not.

### 9.3.2 Configuring Time-of-day based rules

A rule can be configured to be active during a part of the day using the **todfrom**, **todto** and **todstatus** parameters.

For example,

```
create ipf rule entry ruleid 20 transprot eq icmp deny todfrom
"9:30:00" todto "18:30:00" todstatus enable
```

The above rule will be active between 9:30 AM and 6:30 PM. Since it blocks all ICMP packets, it means that ICMP packets will be blocked between the given time values. If the **todstatus** had instead been **disable**, then the rule would be inactive between the given times and active during the rest of the day, hence ICMP packets would be allowed between the given times and denied during the rest of the day.

The current time on the modem can be seen using the **`get system`** command, and can be modified using the **`modify system`** command.

### 9.3.3 IP Sessions – Advanced Configuration Issues

For an IP session, the unit looks up its rules for each of the two interfaces, and for each direction on that interface, only once. The matched rules are then applicable for as long as the session is alive. This optimizes the unit's search efforts.

Suppose you configure a rule with ruleid 20.

Some packets do pass through the unit using this rule so that the rule information in the session is initialized. Now suppose you add another rule with ruleid 10, having the same selectors as rule 20, but the action as **deny**. You might now find that the packets continue to pass through, even though you have configured a rule with a lower ruleid which is supposed to discard these very same packets.

This happens because the subsequent packets continue to use the session information stored in the unit, hence they continue to use the “older” rule 20.

However, once the session has timed out, say because of inactivity, the new rule will apply to packets that are received thereafter.

There is, however, a get around to this situation. After adding the ruleid 10, you should invoke the **reset ipf sessions** command to force the rule lookup. An IP filter rule has parameters such as the TCP syn flag, which can be used along with the protocol, the source/destination addresses and the ports, to identify a packet as belonging to a particular session.

Assume you have the following rule configured:

```
$ create ipf rule entry ruleid 20 dir out ifname ppp-0 transprot
eq tcp destport eq 23 tcptype syn act accept enable
```

This allows all TCP packets, which have TCP syn flags set, to reach a telnet server (port 23) via ppp-0. The first packet that flows out will have the TCP syn flag set, since this is the manner in which TCP initiates a connection formation.

So the first packet passes through successfully.

The subsequent outgoing packets do not have the TCP syn flag set, but they still pass through successfully. This is because the first packet, which passed successfully, initializes the session information to say that the action specified by rule 20 is to be taken for packets belonging to this session going out via ppp-0.

With subsequent packets, the unit decides that they belong to the same session based only on the protocol and the source/destination addresses and ports (which are anyway the same as the first packet). Hence the rule effectively allows all outgoing packets associated with a LAN client doing a telnet to a WAN host.

#### 9.3.3.1 Stateful Filtering

The SAR130 unit's Stateful filtering feature allows you to permit packet flow in one direction only if a session has been initiated from the other direction.

If you want to permit telnet connections from your LAN to the WAN without anyone from outside being able to telnet into your LAN, you will create rules in both directions using the TCP flag option. The rules would resemble the following.

```
$ create ipf rule entry ruleid 20 dir out ifname ppp-0 transprot
eq tcp destport eq num 23 tcptype syn act accept enable
```

The first rule allows all telnet connections initiated from the LAN.

```
$ create ipf rule entry ruleid 21 dir in ifname ppp-0 transprot
eq tcp destport eq num 23 tcptype syn act deny enable
```

This rule denies all telnet connections initiated from the WAN,

```
$ create ipf rule entry ruleid 22 dir in ifname ppp-0 transprot
eq tcp srcport eq 23 act accept enable
```

This rule allows all packets coming from telnet servers, presumably in response to connections initiated from the LAN.

Stateful filtering will allow you to achieve all this by creating a single rule. To use it, you would create just an outgoing rule with the storestate flag set to enable:

```
$ create ipf rule entry ruleid 20 dir out ifname ppp-0 transprot
eq tcp destport num 23 act accept enable storestate enable
```

Now, when a client on the LAN tries to telnet outside, its packets would go out because this rule allows them to. Thus, in the corresponding IP session, the outgoing rule on ppp-0 is marked as **rule 20**. Additionally, since the storestate flag is enabled, the unit also marks the incoming rule on ppp-0 for the session as **rule 20**.

When it detects incoming packets on ppp-0 belonging to the same session, it lets them through.

Thus, the storestate flag allows you to permit the flow of packets in one direction on an interface, provided at least one packet belonging to the session has flown in the other direction first. This holds as long as the session is alive.

The storestate rule may affect the rule action in the other direction as well, and if this rule is matched, then it also updates the session parameters for the other direction, irrespective of the fact that there was some earlier rule present in the other direction.

On the ethernet interface, rule id 10 and rule id 20 are the rules used. If a telnet is originated from a PC to the modem, then, ruleID 10 will be used in IN direction. When the response to PC is sent from the modem, ruleid 20 is used. As this is a ”storestate" rule, it will overwrite the ruleid 10 in the “IN” direction. So, now, even if ruleID 10 is disabled and changed to a DENY rule, it will not have any impact on the packet flow.

### 9.3.4 IP Filtering Global Configuration

The **modify ipf global** command is used to control the default actions on various types of interfaces and to set the modem's security level.

The default actions are controlled by the **pvtdefact**, **pubdefact** and **dmzdefact** parameters. For instance, to allow all packets on public interfaces by default, use pubdefact.

**Configuring Default actions on various interfaces**

```
$ modify ipf global pubdefact accept
```

To deny all packets on private interfaces by default, use

```
$ modify ipf global pvtdefact deny
```

**Setting Security Levels**

The security level of the modem is controlled using the **seclevel** parameter.

• To set the security level to High, use the command -

```
$ modify ipf global seclevel high
```

• To set the security level to Medium, use the command -

```
$ modify ipf global seclevel medium
```

• To set the security level to Low use the command -

```
$ modify ipf global seclevel low
```

At each level, different rules become active depending upon the security level specified with the rules.

**NOTE:**

Whether a given rule is currently active or not is determined by three factors. The first is whether the rule is administratively enabled or not.

The second is whether the rule's security level matches the current security level (as shown by the **get ipf global command**). The third is whether the rule's Time of Day parameters (todfrom, todto and todstatus) and the current system time (as shown by the **get system** command) indicate it to be active. The current status of the rule is shown as the “Rule Oper Status” in the **get ipf rule entry** command.

**WARNING:**

When the modem boots up, the time is set to the last committed time.

Hence, the rules applicable on boot up will depend on this last committed time. This will hold as long as the actual current time is determined by the modem using SNTP or till the user configures the correct time using the **modify system** command.

**NOTE:**

The IP filter that you have configured works only on the new session which gets created after enabling it, and will not affect the ongoing sessions which have been initiated before hand.

# 10 Usage Control and hURL Diagnostics

## 10.1 Overview - Usage Control

The Usage Control feature of the unit provides a user authentication mechanism for allowing LAN to WAN access, only after a login/password have been provided by the LAN user. The mechanism gets activated when a new LAN user tries to connect to the WAN.

**Why is user authentication required?**

A number of emerging scenarios require the unit to be under the control of the service provider, with strict limits set on the number of users connected.

Enforcing this requires that whenever a user tries to connect through the unit, he be prompted for some authentication, and only then allowed through.

*Who is a Data User?* To cater to the above requirements a new kind of user, the concept of Data User, has been introduced. A Data User has WAN access privileges through the unit, but does not have any administration privileges, except that he is allowed to modify his own login/password, or the PPP login/password. The Data User is allowed WAN access only after he has provided this login/password. The data user, additionally, has the facility of "bumping off" another user by giving his login password from another machine. A data user is one who uses the unit to access the WAN but has no need to either view or modify the unit's configuration. Multiple Data users can login to the unit. For authentication, the unit interrupts HTTP packets from the unauthenticated user. Authentication is in the form of a login and password in a Data User login page.

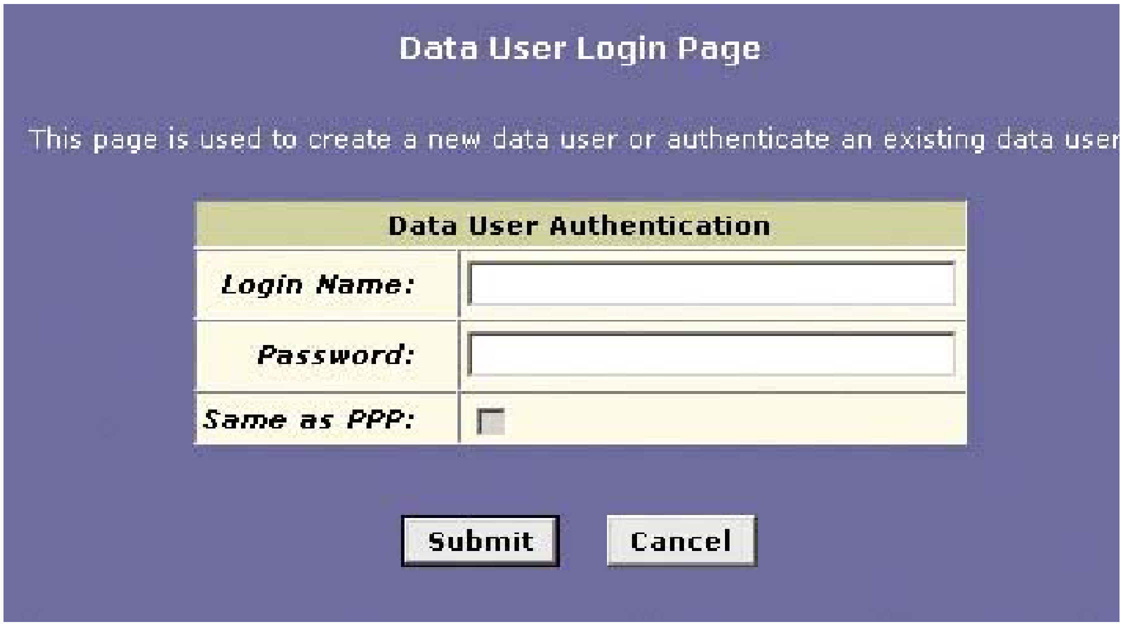
**Who is a Management User?**

A management user can use CLI commands or the HTTP pages for configuring the unit. A management user with root privileges can modify the unit configuration. A management user with user privileges can view, but not modify the unit configuration. Management via HTTP is done by giving the unit's name or IP address, as the URL.

## 10.2 Manual User Authentication Process

When the Usage Control feature is enabled, the unit interrupts WAN-side access by displaying some HTTP pages that force a user to create a new user id or authenticate himself before he is allowed to access the WAN side. For taking this user input and providing appropriate responses to the end user, user-friendly HTTP pages are displayed

**1.** An existing user or a new user provides login name and password using the Data User Login Page. This page comes up whenever any unauthorized LAN to WAN access is detected by the IP layer

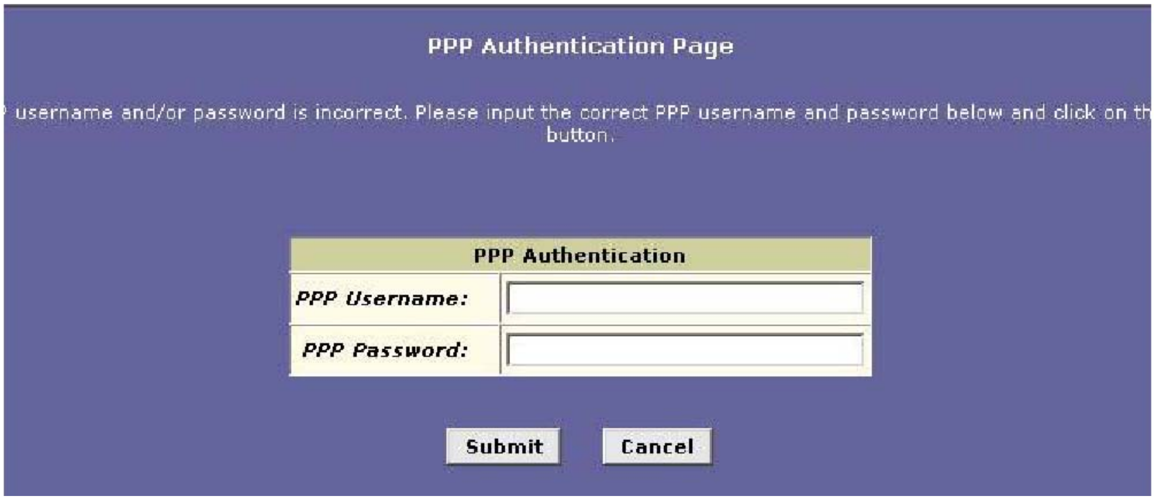


Data User Login Page

This page is used to create a new data user or authenticate an existing data user

| Data User Authentication | |
|---|---|
| Login Name: | |
| Password: | |
| Same as PPP: | □ |

After data user authentication, if the WAN link is down because of PPP authentication failure, the user will see the following page where he can give the correct PPP username and password:



PPP Authentication Page

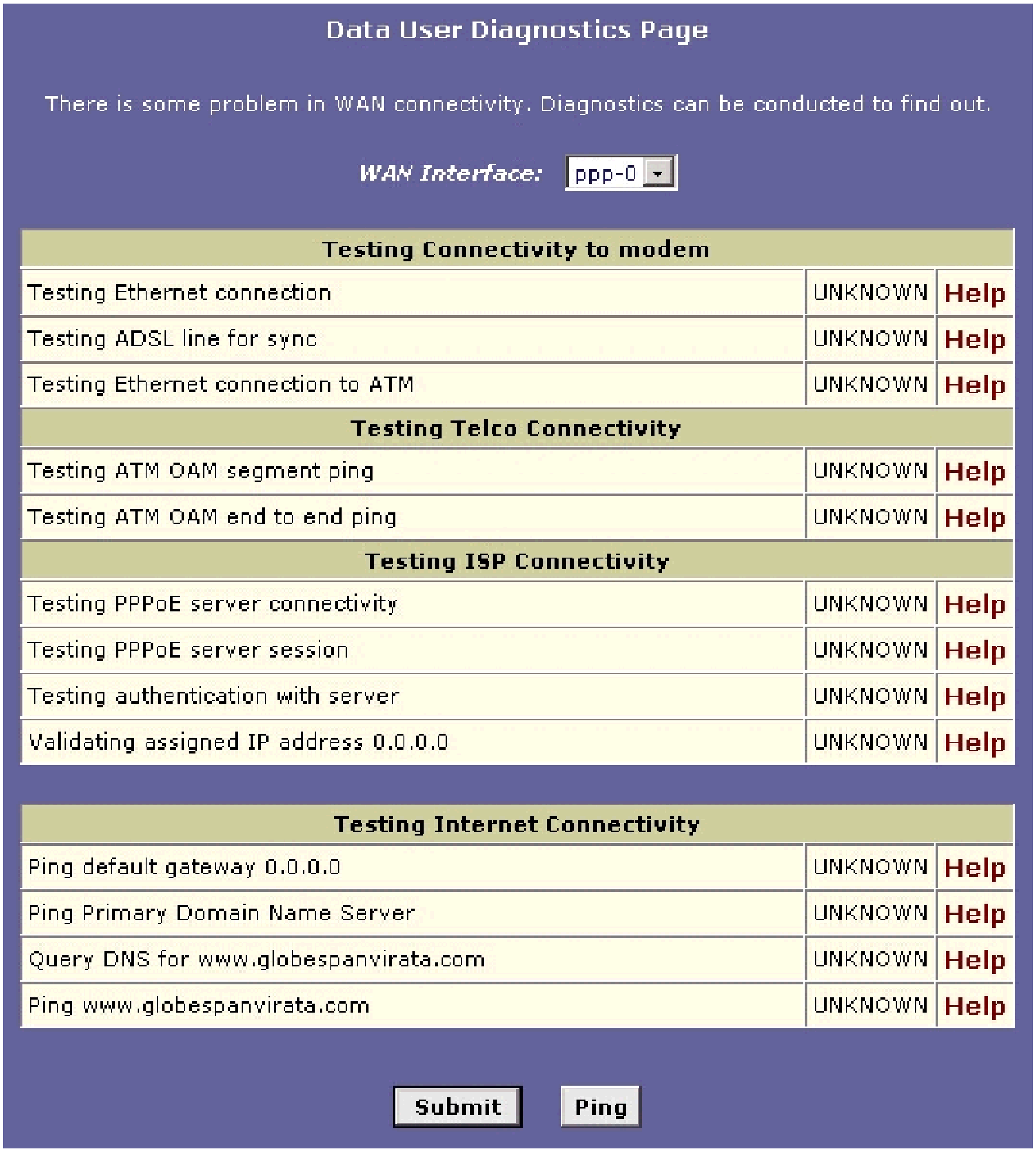
username and/or password is incorrect. Please input the correct PPP username and password below and click on th button.

| PPP Authentication | |
|---|---|
| PPP Username: | |
| PPP Password: | |

**NOTE:**

If **maxauthtries** and **authretrydelay** parameters are configured for the PPP interface then you may see the **"Connection in progress"** page if the PPP authentucation retries are in progress.

However, if the WAN link is down for any other reason then the user see the Diagnostics page:



Data User Diagnostics Page

There is some problem in WAN connectivity. Diagnostics can be conducted to find out.

WAN Interface: ppp-0

| Testing Connectivity to modem | | |
|---|---|---|
| Testing Ethernet connection | UNKNOWN | Help |
| Testing ADSL line for sync | UNKNOWN | Help |
| Testing Ethernet connection to ATM | UNKNOWN | Help |
| **Testing Telco Connectivity** | | |
| Testing ATM OAM segment ping | UNKNOWN | Help |
| Testing ATM OAM end to end ping | UNKNOWN | Help |
| **Testing ISP Connectivity** | | |
| Testing PPPoE server connectivity | UNKNOWN | Help |
| Testing PPPoE server session | UNKNOWN | Help |
| Testing authentication with server | UNKNOWN | Help |
| Validating assigned IP address 0.0.0.0 | UNKNOWN | Help |

| Testing Internet Connectivity | | |
|---|---|---|
| Ping default gateway 0.0.0.0 | UNKNOWN | Help |
| Ping Primary Domain Name Server | UNKNOWN | Help |
| Query DNS for www.globespanvirata.com | UNKNOWN | Help |
| Ping www.globespanvirata.com | UNKNOWN | Help |

**Data User Login Page**

- To begin authentication process, the data user uses the **Login Name** and **Password fields of** the Data User Login Page.

*Login Name* Input Field.

The data user uses this field to provide an existing data user name for authentication purposes, or, create a new data user, subject to the maximum number of data users allowed in the system.

*Password* Input Field.

The input password is provided by the user in this field, to authenticate an existing user or indicate a new data user entry.

*Same as PPP* Check Box.

This checkbox is displayed ONLY when the usage control feature for PPP interface is enabled. Choosing the common login option is allowed only when the PPP security entry is not created. The checkbox is not displayed at all, if the WAN interface used is not a PPP interface. The common login mechanism is provided purely as a convenience, so that the user does not have to configure the PPP login separately. The check box is checked by default, if the PPP security entry is not created. The check box is unchecked and greyed out if the security entry already exists.

**NOTE:**

Once authenticated, the unit remembers the Data User's IP address and all further packets from this data user pass through unhindered.

The IP address is remembered across boots.

**Submit Button.**

Click this button to submit the user login and password information.

**Cancel Button.**

Click this button to cancel the changes and refresh the Data User Login Page.

**2.** On submitting above information, the user is directed to one of the following pages:

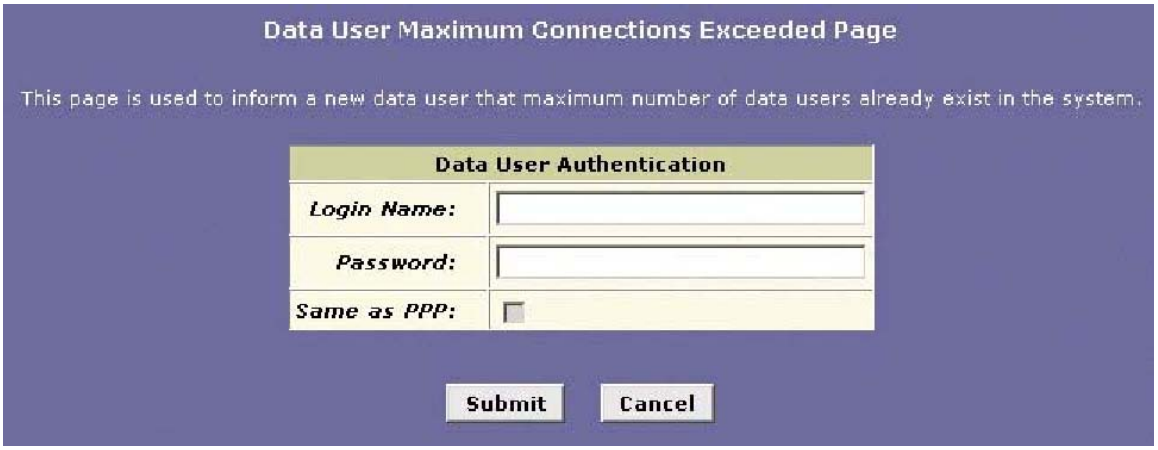
**a.** The original URL on the WAN side, if the data user already exists but is not active from any machine, and the correct password has been submitted. Or, if it is a new data user and the number of data users in the system has NOT exceeded the maximum permissible.

**NOTE:**

Usually, after the data user logs in, his browser gets automatically redirected to the site he was originally trying to access. However, in certain situations, after having logged in, the data user may need to close his existing browser window and open another one to get to the desired site. This limitation is due to the fact that most browsers cache DNS responses locally, instead of using the system DNS cache. This kind of a situation occurs only if the data user logs in while the WAN interface is down. Note that in such cases, it does not suffice to start a new window from the existing one, for instance, by doing a ctrl-N. The data user needs to actually close the existing window and open a new one.

### Data User Maximum Connections Exceeded Page

b. The Data User Maximum Connections Exceeded Page, if the data user is a new user, and the maximum number of data users are already logged in. He will be allowed to input a new login/password combination from this page, very similar to the Data User Login Page.

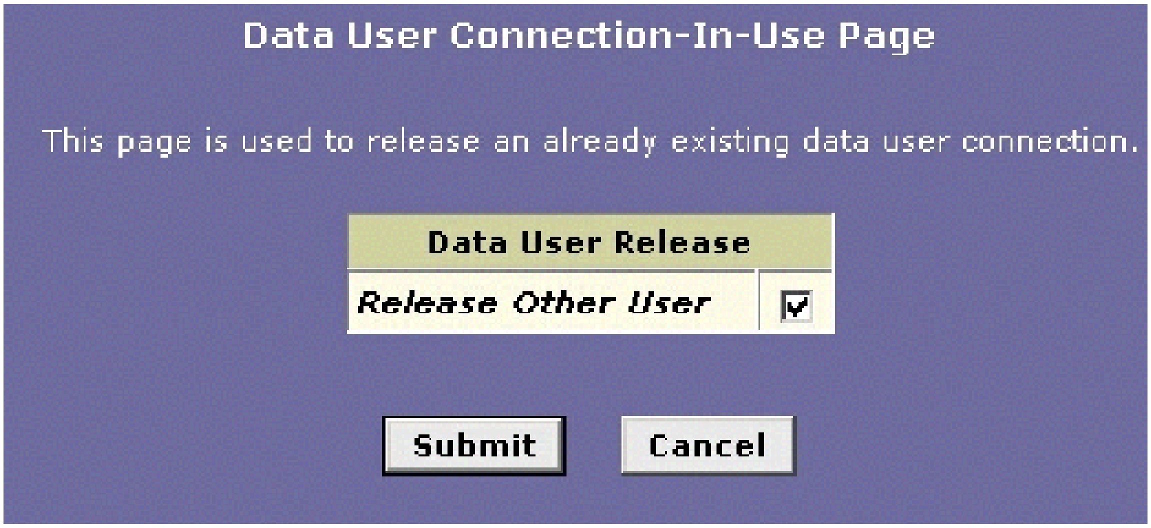


Data User Maximum Connections Exceeded Page

This page is used to inform a new data user that maximum number of data users already exist in the system.

| Data User Authentication | |
|---|---|
| *Login Name:* | |
| *Password:* | |
| *Same as PPP:* | □ |

c. The Data User Connection-in-Use Page appears only after a data user has provided correct input of an existing data user login and password, active from some other machine.



Data User Connection-In-Use Page

This page is used to release an already existing data user connection.

| Data User Release | |
|---|---|
| *Release Other User* | ☑ |

**Data User Connection-in-Use Page**

Release Other User Check Box.

This checkbox is checked by the data user when he wants to bump off the existing user with the same login.

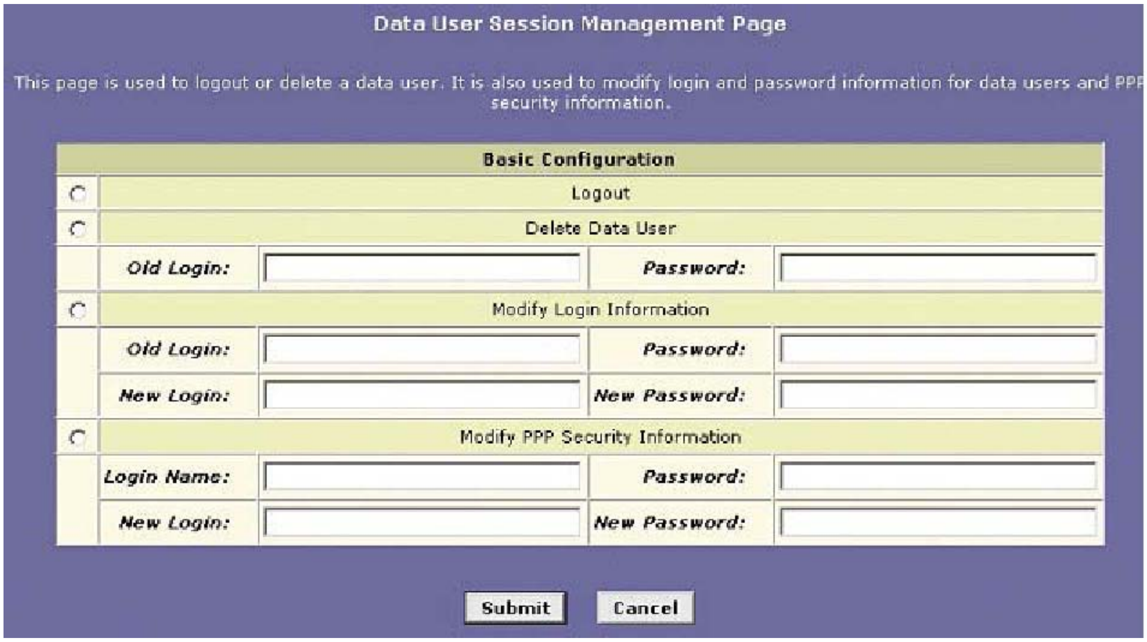
Submit Button.

Clicking this button will "bump" off the existing data user, and the new IP address will be renumbered, for authenticating this particular data user. The data user will automatically be redirected to the original IP address that he had typed in, to access the WAN side.

Cancel Button.

Clicking on this button cancels the changes made in the page by the user, and refreshes the page.

3. The Data User Session Management Page is used by the data user to Logout, Delete, and modify Login and/or PPP Security information for any data user.



Data User Session Management Page

This page is used to logout or delete a data user. It is also used to modify login and password information for data users and PPP security information.

| Basic Configuration | | | | |
|---|---|---|---|---|
| ○ | Logout | | | |
| ○ | Delete Data User | | | |
| | Old Login: | | Password: | |
| ○ | Modify Login Information | | | |
| | Old Login: | | Password: | |
| | New Login: | | New Password: | |
| ○ | Modify PPP Security Information | | | |
| | Login Name: | | Password: | |
| | New Login: | | New Password: | |

**Data User Session Management Page**

*Logout* Option.

If this option is checked, the data user is removed and the information is updated.

*Delete Data User* Option.

If this option is checked, the data user entry will be deleted, provided the IP address for the input data user, and the IP address from where the delete request came, match.

*Modify Login Information* Option.

If this option is checked, the fields are enabled, and the user can modify his login information.

*Modify PPP Security Information* Option.

If this option is checked, the data user can modify his PPP security information.

*Submit* Button.

On clicking this button, appropriate action will take place. On successful completion of the action, a success page is displayed, and on failure, an operation failure page is displayed with the option to go back to the Data User Session Management Page.

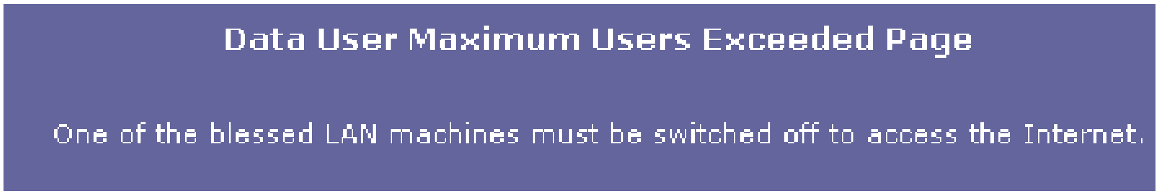
*Cancel* Button.

This will cancel the changes made in the page by the user, and refresh the page.

4. For all other cases, the user will encounter a failure page, with an appropriate error message that will tell him what to do next.

## 10.3 Automatic User Authentication Process

Using the automatic user authentication process, you have the option of allowing the maximum of N users using different LAN machines to access the WAN, without the users having to actually log in using the login page shown in the previous section. In this case, if the (N+1)th user tries to access the WAN, the following page displays.



Data User Maximum Users Exceeded Page

One of the blessed LAN machines must be switched off to access the Internet.

Since you cannot explicitly logout a user in this case, those LAN machines, which are either down or not available on the LAN, are automatically logged out. Now, the new user can access the WAN.

## 10.4 Configuration using CLI

The management user can use the following CLI commands for configuring this feature.

### Enable/Disable Usage Control

- To enable or disable the usage control feature on the unit, enter:

```
$ modify usagectrl [enable|disable]
```

### Configure number of data users

- To modify the maximum number of data users, who can have simultaneous access to the WAN side, enter:

```
$ modify usagectrl [maxusers <decvalue>]
```

### View data user login and IP address

- To view the login name of the data user, and the IP Address of host from which the data user is currently logged in, enter:

```
$ get datauserslist
```

### Delete data users

- To delete all data users, enter:

```
$ reset datauserslist
```

**NOTE:**

It is mandatory to reboot your system after the reset datauserslist command.

**Enabling manual user authentication**

- To enable manual user authentication use the following CLI command:

```
$ modify usagectrl login enable
```

The login parameter specifies whether the login page should display or not.

**Enabling automatic user authentication**

- To enable automatic user authentication use the following CLI command

```
$ modify usagectrl login disable arpcheck 5
```

The **arpcheck** parameter specifies the interval, in minutes, after which the device should check for those LAN machines, which are either down or not available on the LAN, in order to log them out.

**NOTE:**

It is mandatory to specify a non-zero arpcheck parameter if login is disabled.

## 10.5 hURL Diagnostics

The "hURL Diagnostics" feature allows for display of the Diagnostics page when there is a problem with WAN connectivity. This feature is provided with the "Usage Control" feature. However, the "hURL Diagnostics" feature, essentially, is meant for customers who do not need the "Usage Control" feature but want to be able to display Diagnostics pages to LAN users when WAN connectivity is not possible.

### 10.5.1 Configuration using CLI

This feature is implemented as a special mode of "Usage Control".

- To configure:

```
modify usagectrl enable login disable maxusers nolimit
arpcheck 0
```

# 11 Universal Plug and Play (UPnP)

This chapter discusses the benefits of enabling the Universal Plug and Play (UPnP) feature on the unit, the functioning of this feature and configuration details.

## 11.1 Overview

Universal Plug and Play ((UPnP) is an open architecture defined by UPnP forum, for communication between networked devices. It is independent of the network physical layer, can work on any IP enabled interface and makes use of existing protocols. The forum's working groups are standardizing the set of services for particular devices and service types. Currently, separate working groups for Internet Gateways, Audio/Video, Imaging, Mobile Devices, Home Automation, Appliances have been formed. The standard services ensure a level of interoperability between similar types of devices.

On this unit, specifications defined for Internet Gateway are implemented.

## 11.2 Advantages of a UPnP-enabled device

A UPnP-enabled device can autonomously do the following:

- Join a network and obtain an IP address
- Advertise its capabilities
- Learn about the presence and capabilities of other devices
- Communicate directly with other devices.

## 11.3 Components of a UPnP network

The basic building blocks of a UPnP network are, **devices**, **services** and **control points**. The devices and services are built within a **system** and the system is known as a **controlled device**.

### Control Points

A control point (CP) in a UPnP network is a controller capable of discovering and controlling other devices. After discovery, a control point can:

- Retrieve the device description and get a list of associated services
- Retrieve service descriptions for interesting services.
- Invoke actions to control the service.

- Subscribe to the service's event source. Anytime the state of the service changes, the event server will send an event to the control point.
- Inform of any changes that rae taking place.

**Controlled Device**

A controlled device *(CD)* supports multiple devices. Each device contains one or more devices and services. The CD maintains a Device Description Document (DDD) corresponding to each device. This contains the complete information about the device in the XML format and a separate Service Description Document (SDD) for each service embedded in the device and the web interface URL for the box. The CD advertises its devices and services, periodically, on the network, so that all other devices on the network can discover and get to know more details about them. CD also responds to the search requests generated by CPs.

### 11.3.1 How the Controlled Device and Control Points interact

**1.** When a CD is plugged in an IP network it tries to get an IP address for itself unless it is statically configured. It uses either DHCP or Auto-IP protocol to get its own IP address dynamically.

**2.** The CD advertises its DDD URL on a reserved multicast address using the Simple Service Discovery Protocol (SSDP) so that all UPnP capable devices on the network can discover about this device. The CP may also use the SSDP to search the presence of all UPnP capable devices or devices of specific type on the network. The SSDP services are also used to inform other devices whenever they are going down through bye-bye messages.

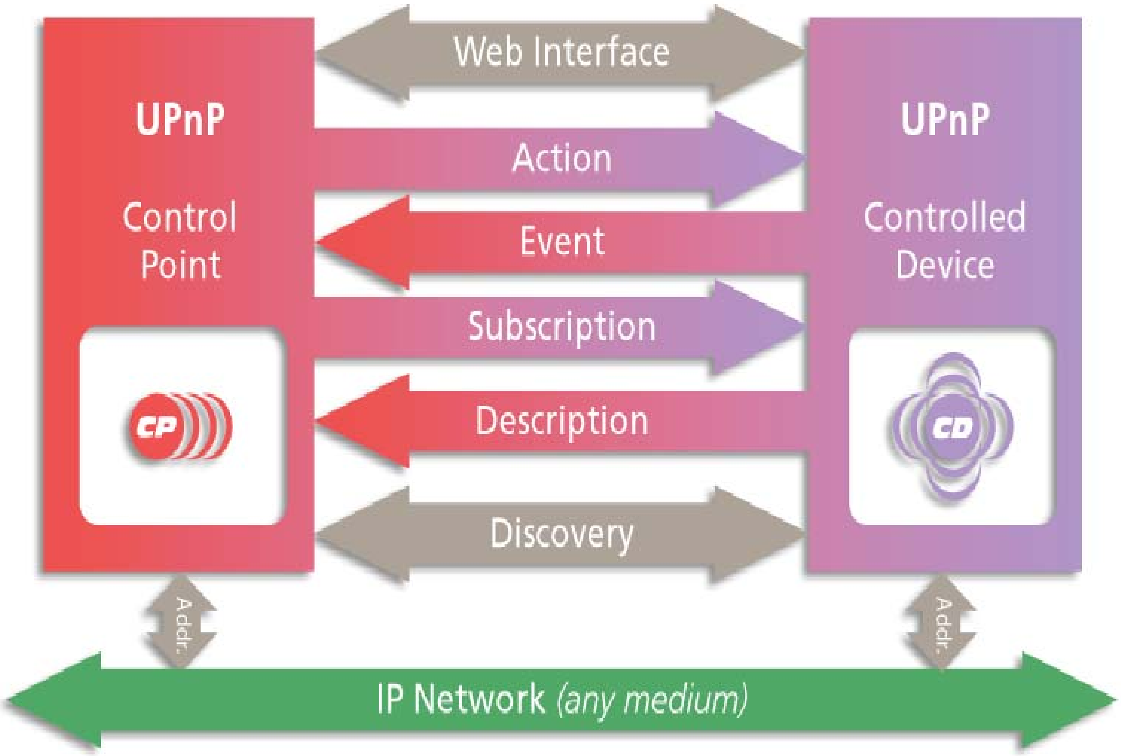
**3.** On receiving the advertisement messages, CP tries to get more information about the device by retrieving its DDD using the device URL present in the advertisement message. Through DDD it gets all SDDs and various action supported within it by the CD.

**4.** The CP may invoke Action request to retrieve or set the values of state variables through various actions supported in the SDDs.

**5.** A CP interested in receiving event notifications subscribes to an event source by sending a request that includes the service of interest, a location to send the events to and a subscription time for the event notification. The CD sends the value of evented variable(s) in the service whenever there is a change to all CP subscribed to the service. A CP can subscribe to more than one service at an instant of time for a specified duration. All CPs are expected to renew the subscription, if they want to continue the subscription, before the timer expiry.

**6.** CD specifies its web interface URL in the presentation page which could be used by CP /Web client to connect to its Web pages. This allows for custom interactions not covered by SOAP.

The following diagram specifies the various interactions between a CP and a CD.



## 11.4 UPnP On SAR130

SAR130 implements the device and services defined by the Internet Gateway Device (IGD) working-group within the UPnP Forum. At present, it supports the following devices and services:

- Internet Gateway Device
- WanDevice
- WanConnectionDevice

**Internet Gateway Device**

The Internet Gateway Device is a root device and supports the Layer 3 Forwarding Service.

- Layer3Forwarding Service - It supports the **get** and **set** actions for setting the default connection state variable, used for Internet connectivity.

***WanDevice***

WanDevice contains the WanCommonInterfaceConfig services and WanConnectionDevices within It. In SAR130, each WanDevice maps to an ATM port while a WanConnectionDevice corresponds to an ATM VC.

• WanCommonInterfaceConfig Service - This is a service in WANDevice that models attributes and actions that are common across all links and all connection instances on a link.This service-type models physical layer properties of a WAN interface on an Internet Gateway used for Internet access. It contains **get** actions to get the link status and statistics information. The only **set** action supported is to enable or disable the Internet connectivity.

**WanConnectionDevice**

The WanConnectionDevice contains the **WanDslLinkConfigService**, **WanPPPConnectionService** and **WANIPConnectionService**. The **WanDSlLinkConfigService** allows you to create or modify ATM VC parameters.

• **WanDslLinkConfigService** - This service-type models physical layer and link layer properties specific to a single physical connection of a Digital Subscriber Line (DSL) modem used for Internet access. This supports various **get** and **set** actions for ATM VC parameters and the layer 3 protocol to be used over AAL5. It also provides an action for getting the modulation type used by the DSL layer. In case AutoConfig is in use, it provides a variable to retrieve this information via CP.

• **WanPPPConnectionService** - This service-type enables an UPnP control point to configure and control PPP connections on the WAN interface. This supports **get** and **set** actions for getting PPP parameters.

You can **create, modify** and **delete** the PPP connection using this service.

In addition to this, on each PPP connection, it allows you to **create** and **delete** NAT redirection rules if NAT is enabled on the PPP interface.

• **WanIPConnectionService -** This service-type enables an UPnP control point to configure and control IP connections on the WAN interface. This supports **get** and **set** actions for getting IP parameters. You can **create**, **modify** and **delete** the IP connection using this service. In addition, on each IP connection, it allows you to **create** and **delete** NAT redirection rules if NAT is enabled on the IP interface.

**NOTE:**

You can have only one device of each type and one service instance in this release of SAR130. Also, you can configure either WANPPPService or WANIPService, at a time.

### 11.4.1 Advantage of having UPnP on SAR130

With the addition of UPnP capability on SAR130, all the UPnP enabled PCs on the Ethernet side can discover the presence or absence of a gateway device, automatically. The CP software available in the PCs can configure SAR130 through the UPnP interface. Also, the UPnP applications, such as games or MSN applications, which are enabled with NAT, can work seamlessly without explicitly creating/deleting the NAT redirection rules required for certain applications such as games, MSN messenger etc. The

NAT-aware application opens only those ports that are required. The ports remain open only till the application is running, thereby reducing security concerns that arise due to permanent opening of a port. With UPnP, you do not require to install the Application-specific Gateways (ALGs) on the router.

## 11.5 Configuration details

- To enable UPnP on your device:

```
modify upnp cfg nbstatus enable
```

When you give this command, UPnP will be enabled next time you boot your device. You get displays for both the current UPnP status and the next boot status.

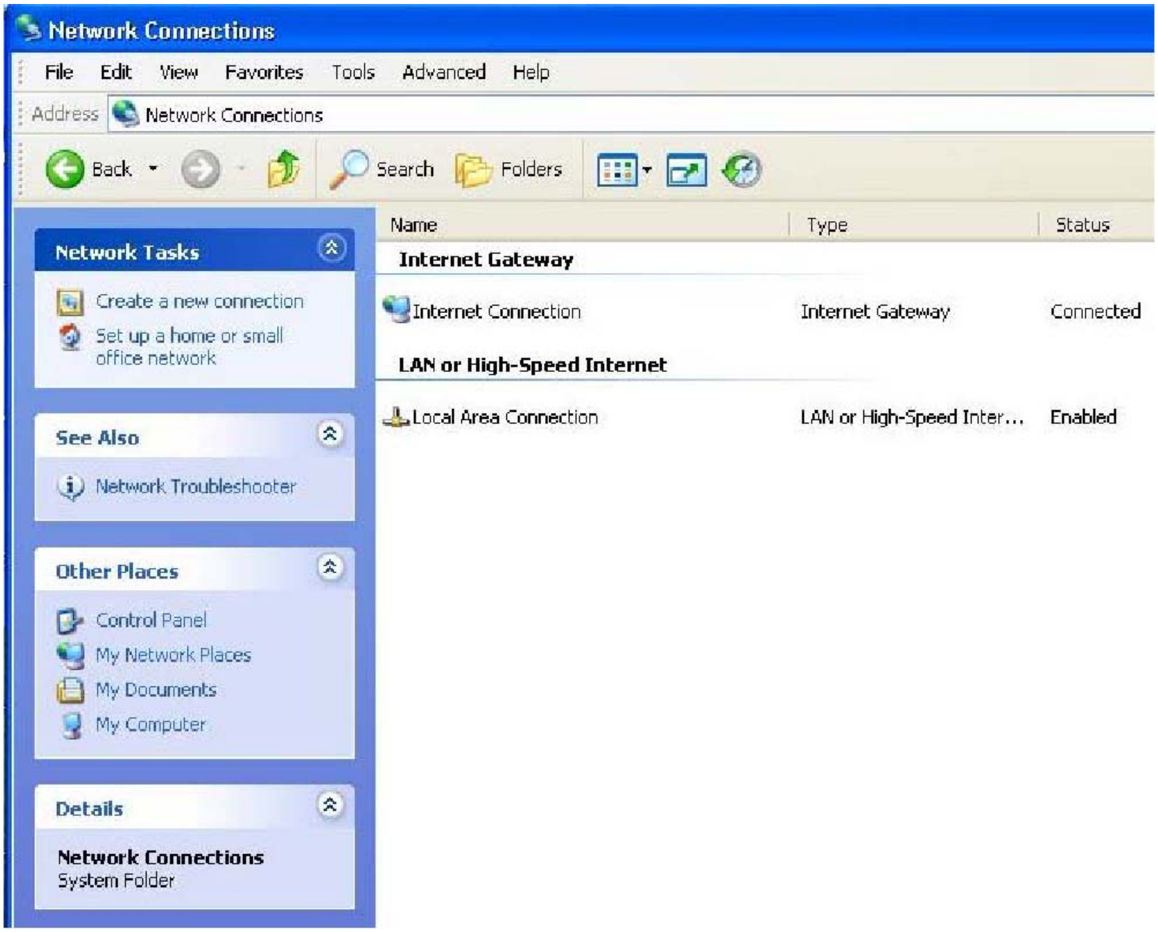
- To verify whether UPnP is enabled on your device:

```
get upnp cfg
```

This returns you the value **enable**, if UpnP is enabled. Or else, it returns **disable**.

**NOTE:**

The unit should be visible as a Internet Gateway icon in the Network Connections folder if IUPnP is enabled on the unit and internet connection is UP. This icon should go away once IUPnP is disabled on the unit, or, internet connection goes down.



| Name | Type | Status |
|---|---|---|
| **Internet Gateway** | | |
| Internet Connection | Internet Gateway | Connected |
| **LAN or High-Speed Internet** | | |
| Local Area Connection | LAN or High-Speed Inter... | Enabled |

### 11.5.1 23.5.1 Configuration and Management through Control Point

SAR130 allows you limited configuration and management of UPnP, via CP applications, such as Device Spy. The following section describes some of the permitted configuration details:

- **To create/modify ATM VC through Control Point:**

From CP, select the **WanDslLinkConfigService** and invoke the action **SetDestinationAddress** specifying the VPI and VCI values as "PVC:VPI/VCI".

For example, if you want to create an ATM VC with VPI 0 and VCI 10 then write it as "PVC:0/10". This command shall create the VC if it does not exist. Or else, it will modify the existing VC, if any, with the VPI/VCI values you have provided.

- **To set the modulation type of an ATM VC:**

From CP, select **WanDslLinkConfigService** and invoke the action **SetATMEncapsulation** (NewModulationType). You can invoke this action at any instant of time, irrespective of whether the VC is created or not. To retrieve the value, you should invoke the action **GetATMEncapsulation**.

- **To set/get the Layer3 protocol information on an ATM VC:**

From CP, select **WanDslLinkConfig Service** and invoke **SetDSLLinkType**, specifying PPPoE/PPPoA/IPoA/CIP/EoA to set the Layer 3 protocol parameter. This information is used to decide the encapsulation type when PPP connection is created over the ATM VC interface. The value of LinkType can be modified as long as no PPP connection is present over the ATM VC.

**• To get DSL layer parameter information including the physical port status:**

From CP, select **WANCommonInterfaceConfig** service and invoke the action **GetCommonLinkProperties**. It returns the data rate used both in upstream and downstream directions, physical port status and the WanAccessType used on the physical interface. On SAR130, this physical interface is DSL.

**• To create PPPoA /PPPoE connection:**

Follow the steps above, to create ATM VC parameters. If you wish to create a PPPoE connection, set the layer 3 protocol to (PPPoE). Or else, set it to PPPoA. From CP, select **WANPPPConnection Service** and invoke the action **SetConnectionType** with the argument IP_routed. This will create an IP routable PPP interface over the ATM VC created earlier.

**• To create IPoA /CIP/EoA connection:**

Follow the steps above, to create ATM VC parameters. If you wish to create an **IPoA connection**, set the **layer 3 protocol to (IPoA)**. If you wish to **create** a **CIP** connection, set the l**ayer 3 protocol to (CIP)**. If you wish to create an **EoA** connection, set the **layer 3 protocol to (EoA)**. From CP, select **WANIPConnectionService** and invoke the action **SetConnectionType** with the argument IP_routed. This will create an IP routable interface over the ATMVC created earlier.

• To set/modify username and password for a PPP Connection: To assign a user and password to a given PPP connection, you need to select the WanPPPConnectionService and invoke ConfigureConnection action with username and password as an argument. This assigns the given username and password to the PPP connection created earlier.

**NOTE:**

If you have already created a user/password combination, it shall modify the existing entry.

• To start a PPP Connection:

Select the **WanPPPConnection** service and invoke the action **RequestConnection**. This initiates the PPP Connection establishment procedure and the connection status changes from Disconnected to connected state. In case the connection already exists, the CD returns only the connection status.

• To stop the PPP Connection:

Select the **WanPPPConnection** service and invoke the action **RequestTermination**. This shall change the PPP Connection status to Disconnected state.

**• To delete a PPP Connection:**

Select the **WanPPPConnection** service and invoke the action **SetConnectionType** with value Unconfigured. This deletes the PPP interface entry corresponding to this service.

• **To create a NAT port mapping on a PPP/IPoA/EoA interface:**

Before you continue with NAT Port mapping, please ensure that NAT is enabled on the unit. You can either use the action **GetNATRSIPStatus** on **WANPPPConnection** service or use the CLI command "get nat global config" to find out if NAT is enabled. To add Port mapping, invoke the action **AddPortMapping** command. This creates/modifies the existing entry with arguments remote host, local host, internal port and protocol for which a port mapping is to be added. While creating the entry, please leave the NewRemoteHost empty to indicate it is a wildcard.

**WARNING:**

The NAT port mappings created through UPnP should not be deleted through other management interfaces such as CLI or HTTP.

• To get a NAT Port mapping entry:

To retrieve the entry, invoke either GetGenericPortMappingEntry or GetSpecificPortmappingActionEntry from the control point. The GetGenericPortMappingEntry action shall return you the entry corresponding to the index value specified in the input. GetSpecificPortMappingEntry action returns the exact entry, if present.

• To Delete the NAT Port mapping entry:

Invoke the action DeletePortMapping, containing remote host, external port and protocol as an input. This shall delete the entry, if it exists. Or else, an error NoSuchEntryInArray(714) shall be returned.

• To create EoA connection in Bridging mode:

Follow the steps above, to create ATM VC parameters. If you wish to create an EoA connection, set the LinkType to EoA. From CP, select WANIPConnection Service and invoke the action SetConnectionType with the argument IP_Bridged. This will create a bridge interface over the ATM VC created earlier.

**WARNING:**

If EoA interface is set up in bridging mode through other management interfaces such as CLI or HTTP, then the connection status of the EoA interface will be displayed as DOWN on the CP, even though the EoA interface is operational.

# 12 Other Device Access Mechanisms

This chapter discusses access mechanisms for the modem, other than CLI.

They include, SNMP, a web-based interface, and the L2 Agent.

## 12.1 Simple Network Management Protocol (SNMP)

Simple Network Management Protocol (SNMP) specifies how information will be exchanged between a Network Management System (NMS) and the network devices managed by it. This information is used to configure and manage the network devices. The SAR130 software provides SNMP access to the following Management Information Base (MIB):

- RFC 1213: *Management Information Base for Network Management of TCP/IP-Based Internet: MIB-II*. Supported groups: system, interfaces, IP, ICMP, UDP, and SNMP groups.

In order to access the unit using an SNMP manager, you need to configure the unit's SNMP agent, as explained in this section.

### 12.1.1 SNMP Communities

Users of SNMP are grouped into categories called **communities**. A community has a name, such as *public*, and access permissions, which give its members read-only or read-write access to the database.

- To create a community called *public* with read-only access, enter:

```
$ create snmp comm community public ro
```

- To see a list of configured communities, enter:

```
$ get snmp comm
```

### 12.1.2 SNMP Hosts

The machines that are part of a community are indicated in the **SNMP Host table**. The SNMP Host table can be configured for managers sending SNMP messages over UDP, EoC or VC channels.

**SNMP over UDP**

If you want to access the unit's SNMP agent from one of the LAN hosts, then you must add the host's IP address to this table.

- To run an SNMP Manager on 192.168.1.3 and access the modem's SNMP Agent, enter:

```
$ create snmp host community public ip 192.168.1.3
```

**SNMP over EoC**

- To access the unit's SNMP agent through EoC you need to add a corresponding entry in SNMP host table. To do so, enter:

```
$ create snmp host hdlc community public
```

·**SNMP over VC**

- To access the unit's SNMP agent through VC, you need to add a corresponding entry in the SNMP host table. To do so, enter:

```
$ create snmp host ilmi community public
```

Since the public community was created for read-only access, this will allow you to read the modem's MIB without allowing you to modify it.

- To see the configured SNMP hosts, enter:

```
$ get snmp host
```

### 12.1.3 SNMP Traps

Traps inform a managing entity of noteworthy or unusual events in the system.

The modem reports some of the events in the system by sending traps to the SNMP manager. In the current release, the modem supports `System Up` and `Authentication failure` traps via the SNMP interface. The `System Up` trap is generated whenever the system comes up successfully. The `Authentication failure` trap is generated whenever an unauthorized SNMP manager tries to access the modem via the SNMP interface.

- To selectively enable or disable the generation of the authentication failure trap, enter:

```
$ modify snmp trap enable
```

or

```
$ modify snmp trap disable
```

By default, generation of the Authentication failure trap is enabled.

- To display the current status, enter:

```
$ get snmp trap
```

### 12.1.4 Providing SNMP Access Across the Modem

To manage one of the LAN hosts from outside using SNMP, and to manage an external machine from one of your LAN hosts using SNMP, you will need to configure SNMP ALGs on the modem's inside and outside interfaces.

SNMP access is usually provided to trusted hosts only. These trusted hosts are configured on the SNMP agent using the IP addresses of the hosts. This means that the local machine that you are providing SNMP access to/from must always get the same public IP address. This is done best by configuring a bimap NAT rule, which provides a one-to-one mapping between one of the modem's public IP addresses and the particular LAN host.

## 12.2 Remote CPE Management

Remote CPE management is about managing a Customer Premise Equipment without establishing IP connectivity with the unit first. Remote CPE management uses either a VC or an EoC channel to provide access to the unit's SNMP agent. For enabling access to the unit's SNMP Agent the VC or EoC channel needs to be enabled and then SNMP packets from the manager can be sent and received for accessing the unit.

### 12.2.1 VC Channel Configuration

In order to access the unit for remote CPE management through a VC, a VC needs to be configured.

- To configure a VC to access the unit for remote CPE management, use the CLI commands below:

```
create ilmi intf ifname atm-0 vpi 0 vci 16
trigger ilmi
```

**NOTE:**

As one VC is reserved for Remote CPE Management, the unit should be configured for one VC more than the maximum number of VCs that will be used for actual data access.

**NOTE:**

Remote Management over VC and TR-37 cannot work simultaneously.

### 12.2.2 EoC Channel Configuration

- To access the unit for remote CPE management through an EoC channel, HDLC over EoC needs to be enabled, using the CLI command:

```
modify hdlceoc cfg status enable
```

**NOTE:**

In addition to configuring VC channels and EoC channels, you also need to configure SNMP community and SNMP host, to access the unit through EoC and VC channels.

## 12.3 Web-based Interface

The SAR130 provides a Web-based interface that enables configuring the unit from a web browser. The Web-based interface includes a subset of the configuration options available in the CLI.

Functions *Not* Enabled in the Web-based Interface

- Creating the Ethernet port
- Creating ATM ports and traffic descriptors
- Configuring SNMP
- Configuring IGMP
- Configuring ILMI

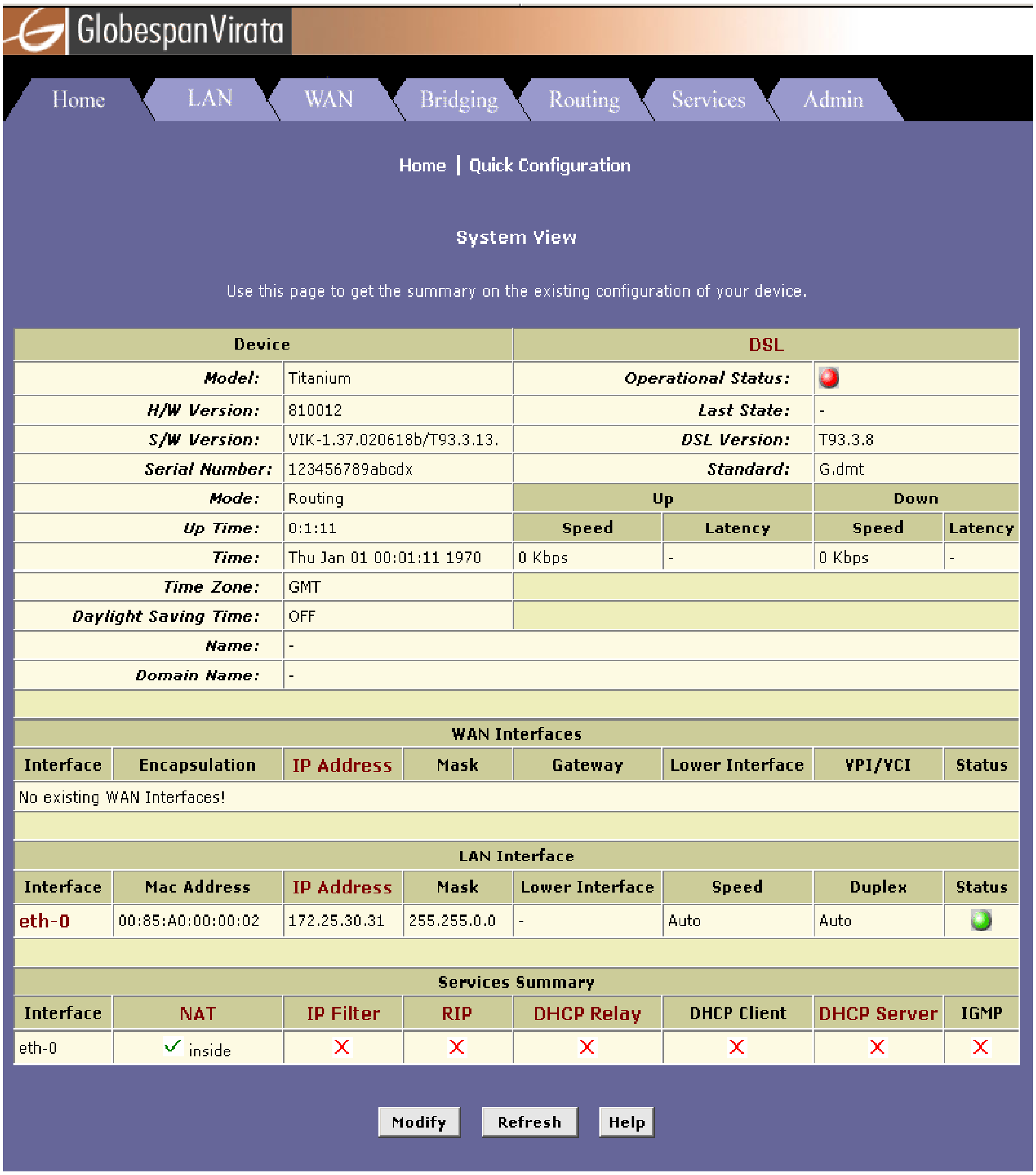
### 12.3.1 Accessing the Web-based Interface

The Web-based interface, like the CLI, is part of the image you load in the system flash. You can access the interface from a computer that has a Ipenabled network connection to the unit via the LAN interface. The PC from which you access the interface *must be in the same subnet* as the device's LAN port IP address.

The Web-based interface is best viewed using Microsoft Internet Explorer® version 5.0 or later versions. Support for Java® and Javascript®should be enabled in the browser.

To access the Web-based interface, simply type the LAN port IP address in your browser's address/location box. The pre-assigned LAN port IP address is 192.168.7.1. You are prompted to log in to the interface. The login name and password are the same as those pre-configured for the CLI interface. You can change the password. The pre-assigned user name and password are DSL and DSL.

After you have logged in, the System View page displays, as shown below.



System View

Use this page to get the summary on the existing configuration of your device.

| Device | | DSL | | | |
|---|---|---|---|---|---|
| Model: | Titanium | Operational Status: | | 🔴 | |
| H/W Version: | 810012 | Last State: | | - | |
| S/W Version: | VIK-1.37.020618b/T93.3.13. | DSL Version: | | T93.3.8 | |
| Serial Number: | 123456789abcdx | Standard: | | G.dmt | |
| Mode: | Routing | Up | | Down | |
| Up Time: | 0:1:11 | Speed | Latency | Speed | Latency |
| Time: | Thu Jan 01 00:01:11 1970 | 0 Kbps | - | 0 Kbps | - |
| Time Zone: | GMT | | | | |
| Daylight Saving Time: | OFF | | | | |
| Name: | - | | | | |
| Domain Name: | - | | | | |

WAN Interfaces

| Interface | Encapsulation | IP Address | Mask | Gateway | Lower Interface | VPI/VCI | Status |
|---|---|---|---|---|---|---|---|
| No existing WAN Interfaces! | | | | | | | |

LAN Interface

| Interface | Mac Address | IP Address | Mask | Lower Interface | Speed | Duplex | Status |
|---|---|---|---|---|---|---|---|
| eth-0 | 00:85:A0:00:00:02 | 172.25.30.31 | 255.255.0.0 | - | Auto | Auto | 🟢 |

Services Summary

| Interface | NAT | IP Filter | RIP | DHCP Relay | DHCP Client | DHCP Server | IGMP |
|---|---|---|---|---|---|---|---|
| eth-0 | ✓ inside | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |

## 12.3.2 Accessing the Quick Configuration Page

The interface also includes a Quick Configuration page. This helps you access  the settings that you may need to configure when you install your own product.

You can also access all Quick Configuration settings under their respective tabs. This page is available under the Home tab and can also be accessed directly by specifying the LAN port IP address followed by “/setup”.

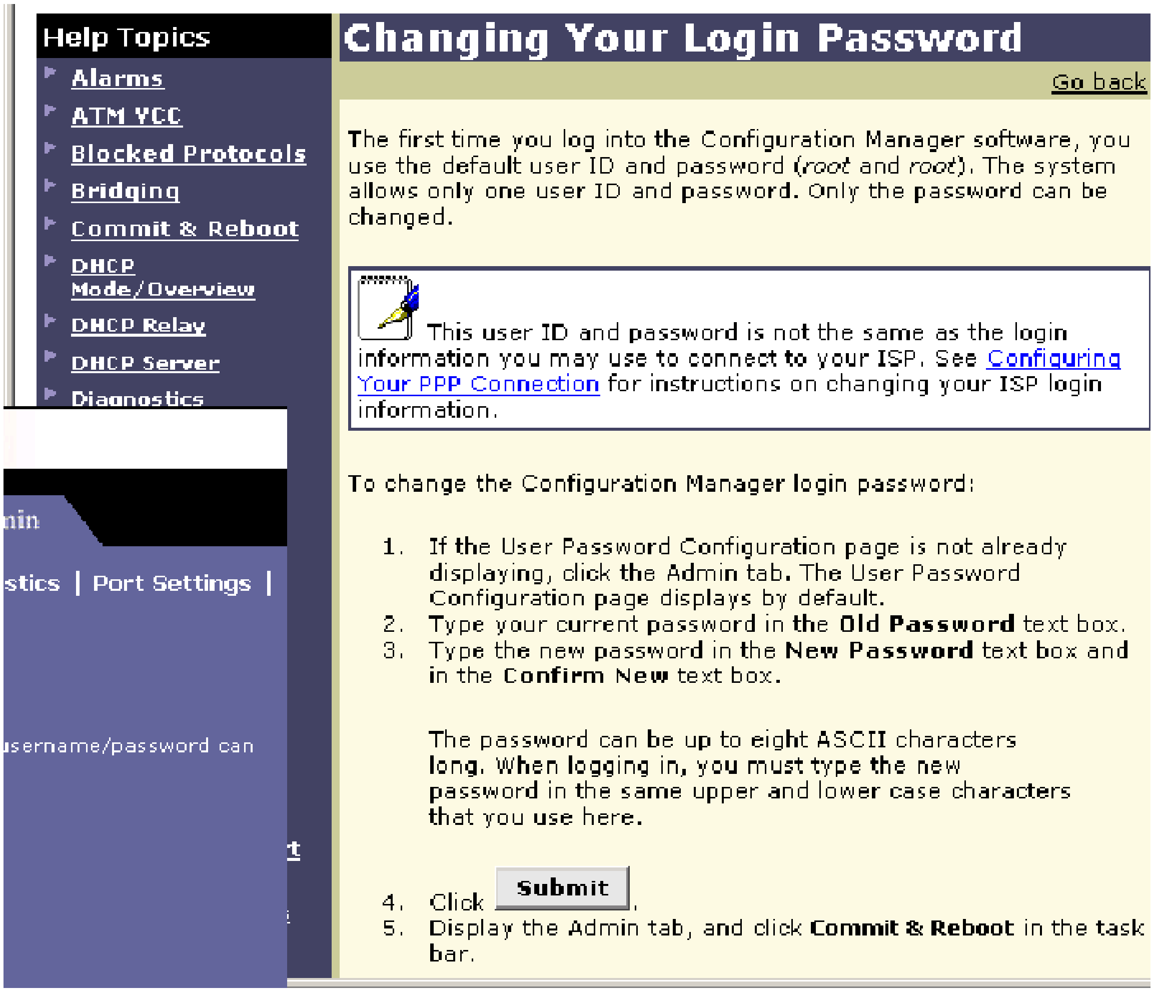
For example:

```
http://192.168.7.1/setup
```

The Quick Configuration page in the Home tab displays. You can also access all other interface functions from the Quick Configuration page.

### 12.3.3 User Instructions

You can access all tasks by clicking the tabs that display horizontally at the top of the page. A menu of related tasks display at the top of each tab. Click these to display the specific tasks. All changes are effective when submitted. However, you must **commit** changes to avoid having them overwritten by the previous configuration when the system reboots. The **commit** and **reboot** feature is located in the Admin tab, as shown below. You can display online help in a separate window by clicking the Help button on any main topic page. A help topic displays relating to the current page, with sidebar navigation to other help topics, as shown.



**Changing Your Login Password**

The first time you log into the Configuration Manager software, you use the default user ID and password (*root* and *root*). The system allows only one user ID and password. Only the password can be changed.

This user ID and password is not the same as the login information you may use to connect to your ISP. See Configuring Your PPP Connection for instructions on changing your ISP login information.

To change the Configuration Manager login password:

1. If the User Password Configuration page is not already displaying, click the Admin tab. The User Password Configuration page displays by default.
2. Type your current password in the **Old Password** text box.
3. Type the new password in the **New Password** text box and in the **Confirm New** text box.

   The password can be up to eight ASCII characters long. When logging in, you must type the new password in the same upper and lower case characters that you use here.

4. Click Submit .
5. Display the Admin tab, and click **Commit & Reboot** in the task bar.

### 12.3.4 Remote Image Upgrade

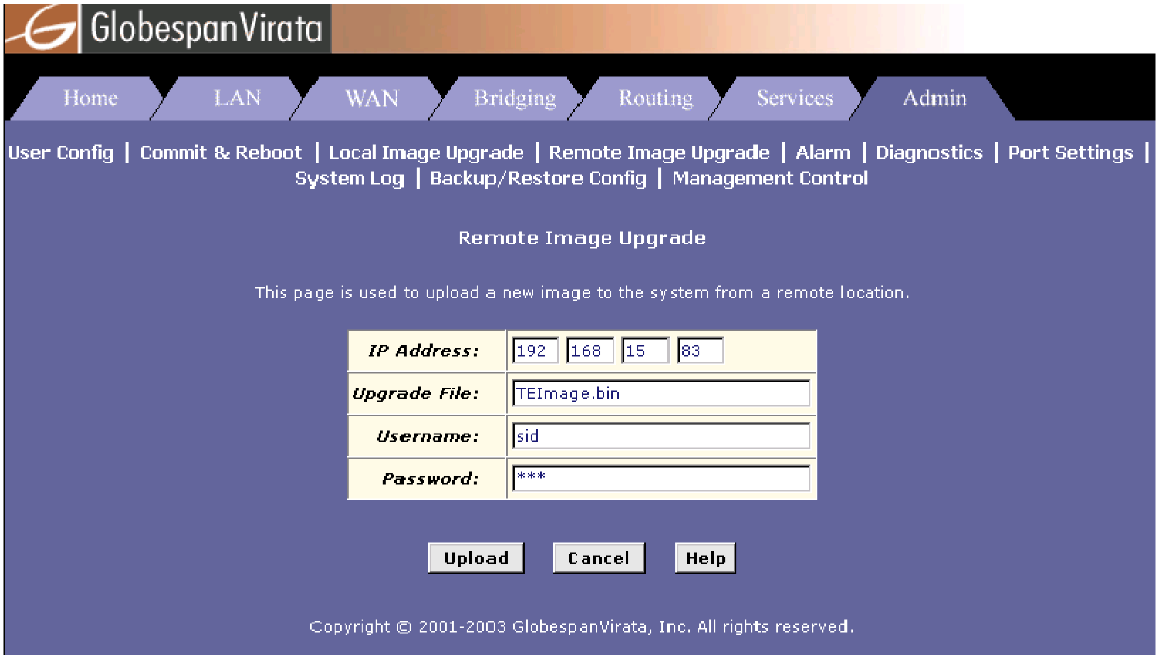
This page allows you to upgrade your device with a new binary image from a remote FTP server on the WAN. You can configure the device to perform remote image upgrades using either of two methods:

- One-click image upgrade: Feature Not Suppoprted
- Non-one-click image upgrade: The user is prompted to specify the FTP server IP address, username, password, and image file name.

### Configuring non-one-click image upgrade

If you do not configure your device for one-click image upgrade (i.e., if you do not add the `modify image upgrade` command to TEFacs.txt), then the nonone- click image upgrade method will be configured.

In this case, the user must enter all the information needed to download the image upgrade file, as shown below. In addition, no version comparison is performed to verify that the remote image is more recent than the current image.


GlobespanVirata
Home
LAN
WAN
Bridging
Routing
Services
Admin
User Config | Commit & Reboot | Local Image Upgrade | Remote Image Upgrade | Alarm | Diagnostics | Port Settings |
System Log | Backup/Restore Config | Management Control
Upload
Cancel
Help

Remote Image Upgrade

This page is used to upload a new image to the system from a remote location.

| | |
|---|---|
| *IP Address:* | 192 . 168 . 15 . 83 |
| *Upgrade File:* | TEImage.bin |
| *Username:* | sid |
| *Password:* | *** |

Copyright © 2001-2003 GlobespanVirata, Inc. All rights reserved.

### 12.3.4.1 Accessing the remote image upgrade page

To access the remote image upgrade page, select Remote Image Upgrade under the Admin tab, then follow the instructions below for the configured image upgrade method.

### Non-one-click image upgrade

The page shown above will display. Enter the requested information, then click Upload.

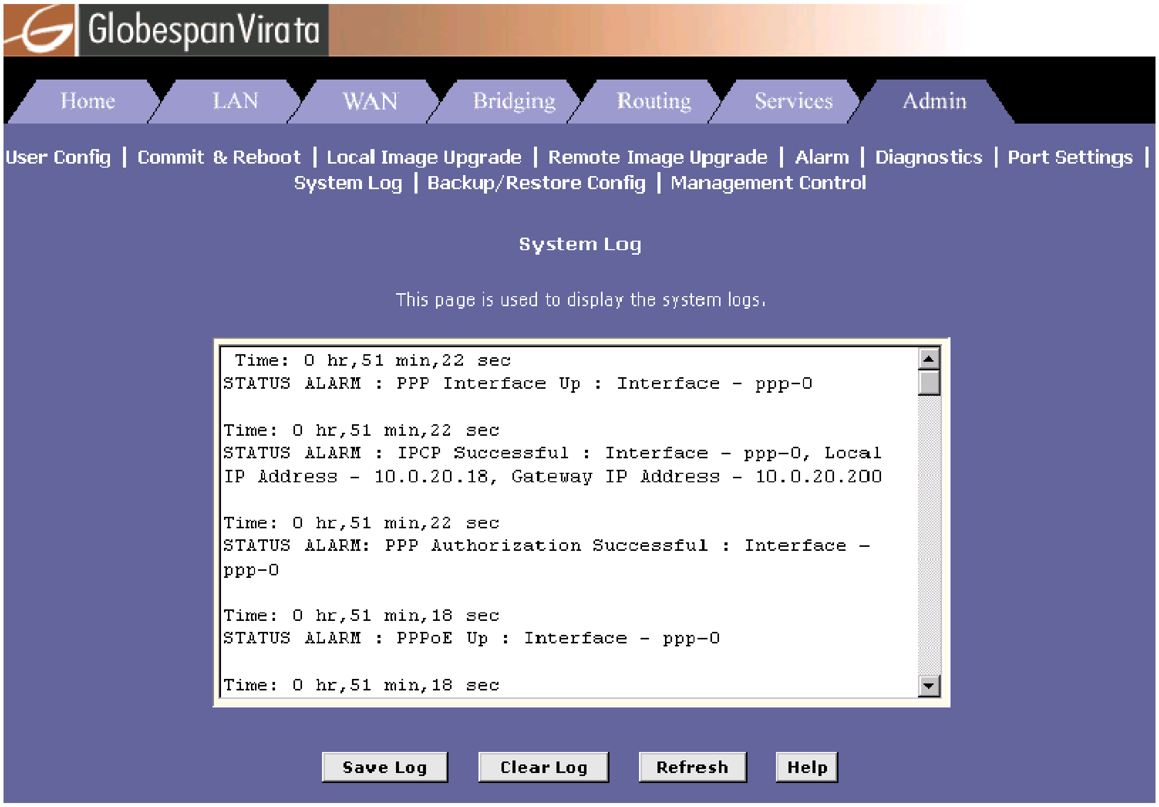
A message is displayed informing that the process will take a few minutes followed by a system reboot. Click OK.

The screen flashes while the upgrade is in progress. When the upgrade is finished, the system automatically reboots and displays a message indicating that the upgrade was successful.

### 12.3.5 System Log

This screen displays firewall-related, PPP-related and DNS relay-related logs.

You can use this interface to save the logs, clear the logs, and refresh the logs to view the latest information.


GlobespanVirata
Home
LAN
WAN
Bridging
Routing
Services
Admin
User Config | Commit & Reboot | Local Image Upgrade | Remote Image Upgrade | Alarm | Diagnostics | Port Settings |
System Log | Backup/Restore Config | Management Control
Save Log
Clear Log
Refresh
Help

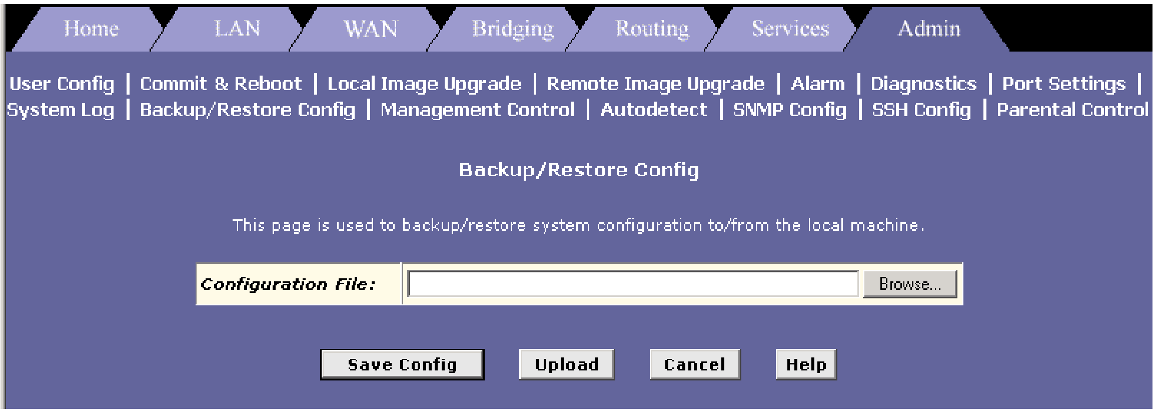
System Log

This page is used to display the system logs.

```
 Time: 0 hr,51 min,22 sec
STATUS ALARM : PPP Interface Up : Interface - ppp-0

Time: 0 hr,51 min,22 sec
STATUS ALARM : IPCP Successful : Interface - ppp-0, Local
IP Address - 10.0.20.18, Gateway IP Address - 10.0.20.200

Time: 0 hr,51 min,22 sec
STATUS ALARM: PPP Authorization Successful : Interface -
ppp-0

Time: 0 hr,51 min,18 sec
STATUS ALARM : PPPoE Up : Interface - ppp-0

Time: 0 hr,51 min,18 sec
```

### 12.3.6 24.3.6 Configuration Save & Restore

The HTTP interface includes a "Backup/Restore" page as shown below, for doing configuration save and restore.


Home
LAN
WAN
Bridging
Routing
Services
Admin
User Config | Commit & Reboot | Local Image Upgrade | Remote Image Upgrade | Alarm | Diagnostics | Port Settings |
System Log | Backup/Restore Config | Management Control | Autodetect | SNMP Config | SSH Config | Parental Control
Browse...
Save Config
Upload
Cancel
Help

Backup/Restore Config

This page is used to backup/restore system configuration to/from the local machine.

Configuration File:

To save the current configuration from SAR130 to your local PC, click on the “Save Config” button.

To apply a configuration saved on your PC to the unit, select the configuration file using “Browse…” button and click the “Upload” button. On clicking the “Upload” button the file will be applied to the current configuration and the unit will reboot and using the new settings.

**NOTE:**

The configuration will be applied only if auto update flag is true.

### 12.3.7 Web-based Diagnostics

The Web-based interface includes a diagnostics program that runs a series of system tests, which examine the system’s physical layer functioning up to the point of connectivity with the Internet. You can access the diagnostics from the Admin tab.

### 12.3.8 Management Control Feature

The management control feature allows you to logout a HTTP, TELNET or FTP session that is accessing the modem, if it is lying idle for a defined duration. It also allows you to block HTTP, TELNET, FTP, SNMP and TFTP access to the modem. This feature provides enhanced security.

This feature has two different types of configuration options available. They are described, below.

**`Inactivity timeout`**

The Inactivity timeout feature forces a user to log in again if he is inactive for a certain period of time. This time period can be configured by the user. The Inactivity timeout feature, when enabled, applies to the following protocols: HTTP, TELNET and FTP.

• To see the current value of the Inactivity timeout issue the following command on the CLI:

```
get mctl inactivity
```

An Inactivity timeout value of zero indicates that the Inactivity Timeout feature is disabled. A non-zero value of the Inactivity timeout indicates that Inactivity Timeout feature is enabled for all supported protocols.

• To set the value of the Inactivity Timeout, issue the following command on the CLI:

```
modify mctl inactivity <time-out>
```

When Inactivity timeout is enabled, then any user staying idle for more than the timeout value will be automatically logged out. After being logged out the user will be required to log in again by providing the username/password pair. This behavior applies to all protocols supported for Inactivity Timeout.

Inactivity timeout can also be configured via the web-based interface. To configure management control inactivity timeout value browse to the "Management Control" page under the "Admin" tab.

**`LAN/WAN access`**

The LAN/WAN access control feature lets you block/unblock access to the modem's various services. The services supported by this feature are HTTP, TELNET, FTP, TFTP and SNMP. Enabling LAN access to a particular service will allow all LAN side machines to access that service. Disabling the LAN access will deny any further usage of the service to LAN side machines.

Similarly, enabling WAN side access to any service will allow all WAN side machines to access that service and vice versa.

The Management Control LAN/WAN access feature also lets you configure a list of IP addresses. When this list is configured, only the IP addresses on the list are allowed access from the LAN or WAN side.

**Configuration Details**

- To view LAN/WAN Access, issue the following command on the CLI:

```
get mctl access
```

- To configure LAN side HTTP access use:

```
modify mctl access httplanaccess <enable|disable>
```

- To configure WAN side HTTP access use:

```
modify mctl access httpwanaccess <enable|disable>
```

LAN/WAN access can also be configured via the web-based interface. To configure management control inactivity timeout value browse to the "Management Control" page under the "Admin" tab.

**NOTE:**

For management control LAN/WAN access to work, IP filters must be enabled and IP filter security level must be set to low, medium or high.

- To configure a list of IP addresses, only from which lan/wan access should be allowed, use the command:

```
create mctl iplist ip <ip address>
```

## 12.4 L2 Agent Module

The L2Agent (L2AG) module is defined to provide access to the modem’s management information base through Ethernet. It provides a proprietary framework for exchange of messages between the L2-Manager and the GenAg module on the modem. This facilitates the L2-Manager to read the existing configuration information from the modem, and to set/modify the configuration on the modem.

L2AG receives Ethernet packets from the Ethernet interface. The Ethernet module hands over the packet to L2AG, when a given pattern is found in the received Ethernet packet. L2AG processes the incoming packet and requests GenAg to either provide the existing configuration information or to set the configuration information received in the packet. The response from GenAg is then, forwarded to the L2-Manager in a pre-defined format.

The communication packet between the L2-Manager and L2AG is through UDP packets. UDP packets are generated by the L2-Manager and L2AG. They do not use the existing IP stacks at either end. L2-Manger uses a pre-defined set of parameters for generating the packet and ignores the Ethernet header, IP header and the UDP header data in the received packet. The messages exchanged must be filled in network byte order. The packet information for both sides is given below:

• From the L2-Manager to the agent, the UDP packet has the following parameters:

- Broadcast address at both MAC level and IP level (Destination MAC is ff:ff:ff:ff:ff:ff; Destination IP is 255.255.255.255)

- Source port is 6001 and destination port is 6001

- Source IP is 192.168.0.235.

• From management agent to the L2-Manager, the UDP packet has the following parameters:

- MAC address of the manager (decoded from the packet received from the manager)

- Destination IP is 192.168.0.235

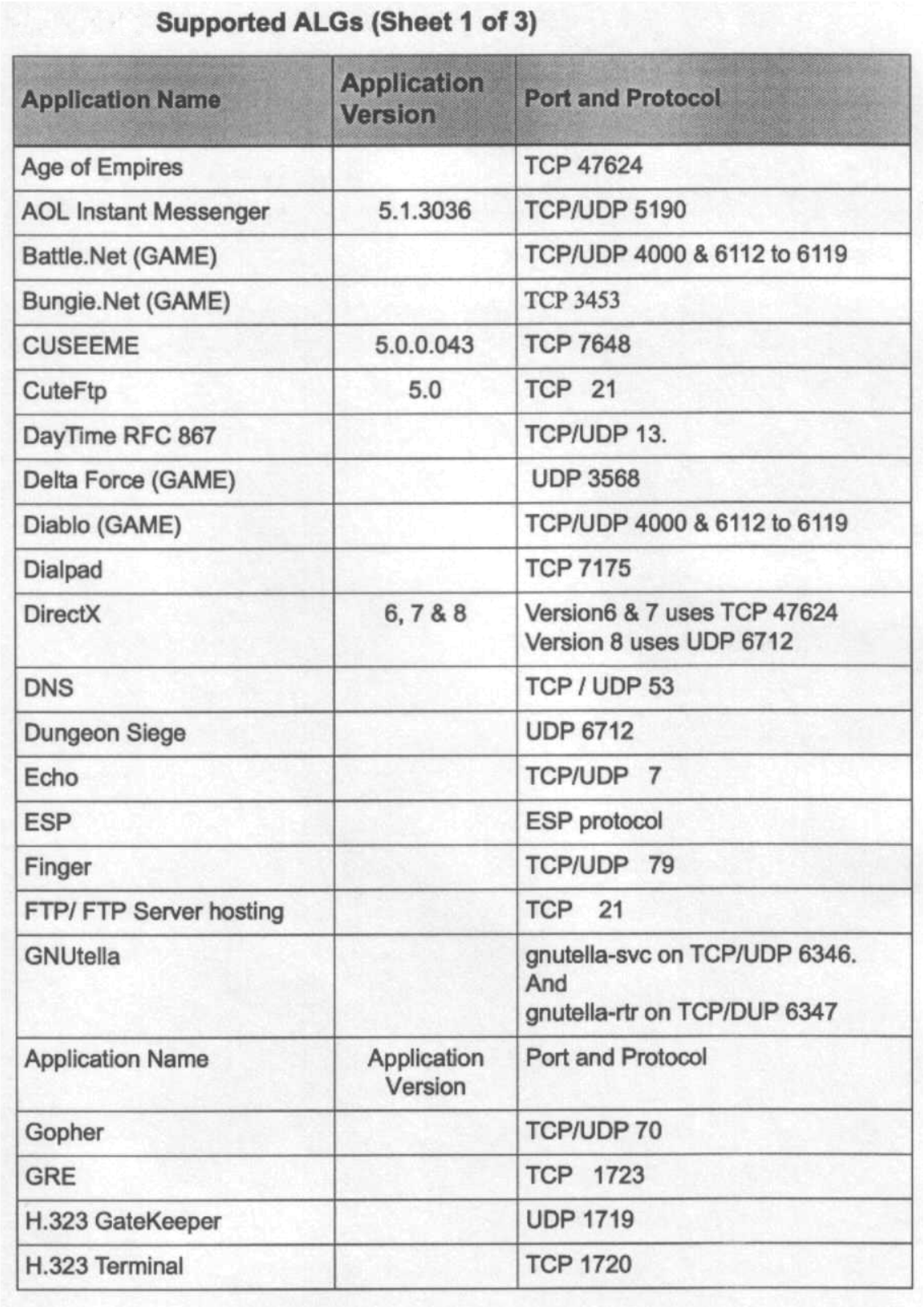
- Source port is 6001 and destination port is 6001

- Source IP is 192.168.0.234

The L2AG is designed as a separate entity over the Ethernet module. It does not use the existing IP/UDP stacks. At the Ethernet module, the packet is sent to the L2AG module, based on a pre-defined pattern, in the packet received. In the transmit direction, the L2AG module builds the entire Ethernet packet and sends the packet directly to the EMac module for transmission.

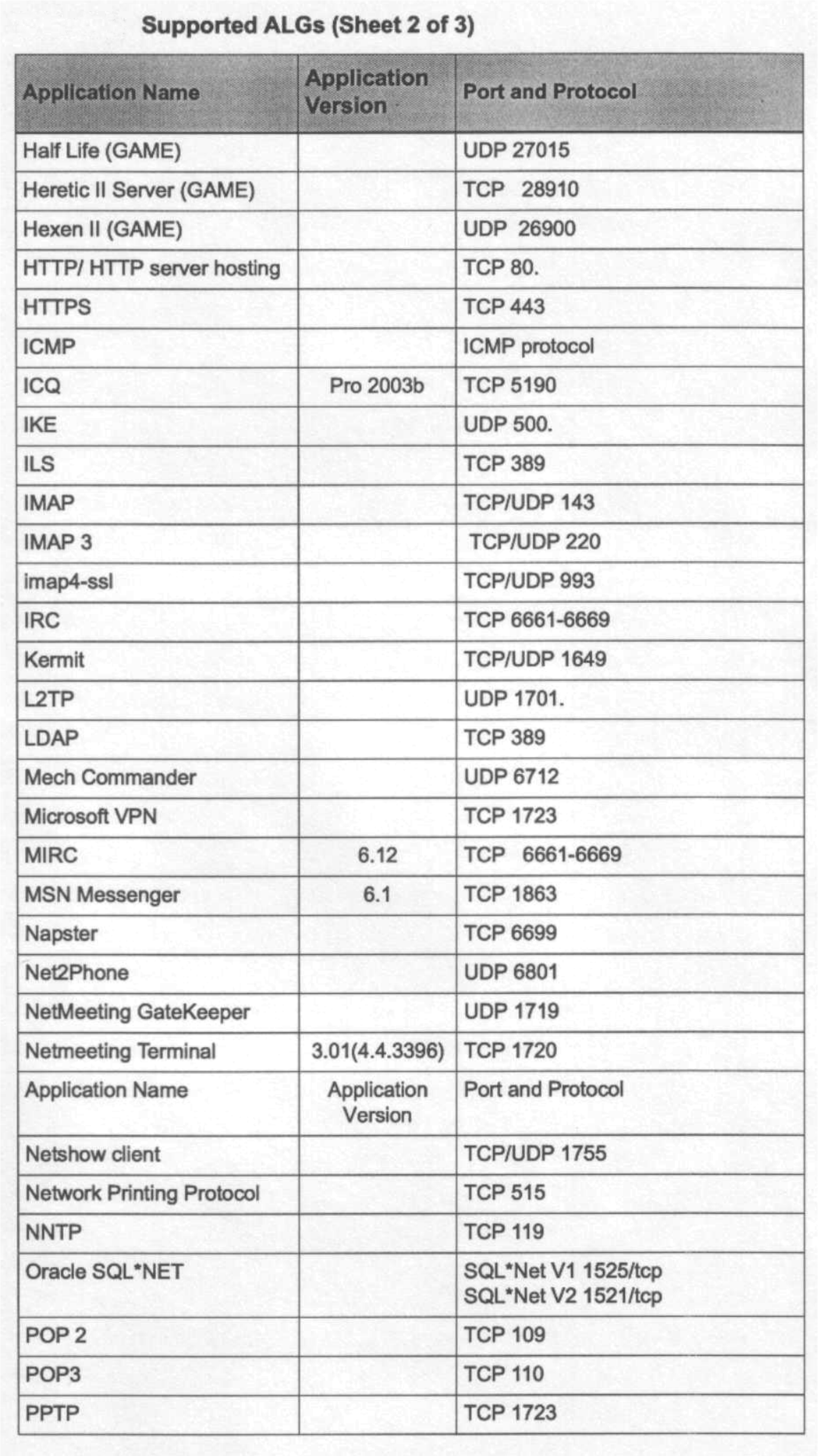
L2 Agent is located above the Ethernet module. It is implemented as a separate task. This task interacts with the GenAg and Ethernet modules through messages. When a message is received at L2AG from the Ethernet module, a message/event is sent to the L2AG task. This message is forwarded to GenAg and L2AG waits for a response from GenAg. After receiving the response from GenAg, L2AG creates the Ethernet packet to be sent to the L2-Manager, and hands it over to the EMac (functional interface) for transmitting the message over the Ethernet interface. L2AG does not interact with any other module in the modem.

# 13ALG's



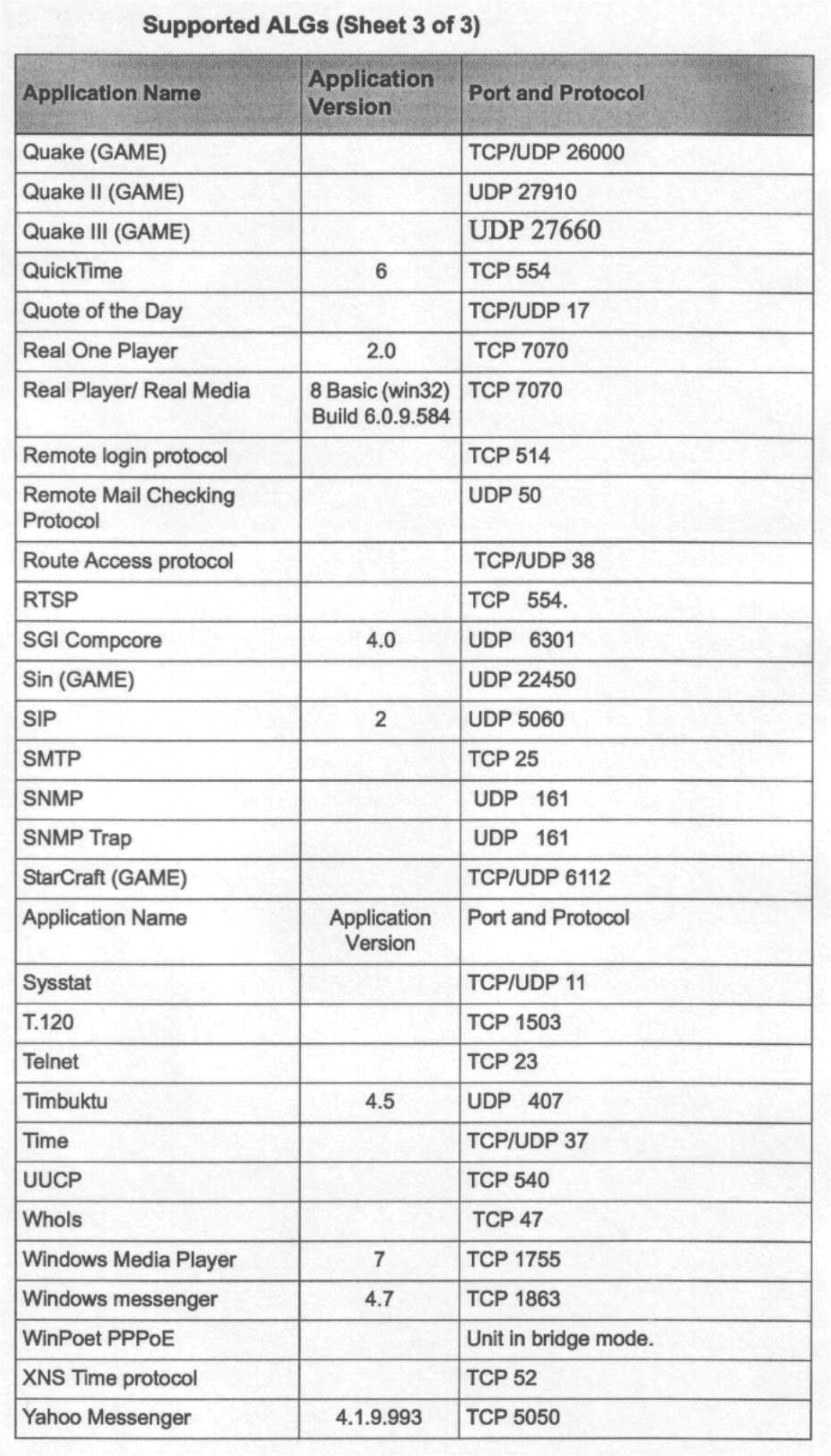
Supported ALGs (Sheet 1 of 3)

| Application Name | Application Version | Port and Protocol |
|---|---|---|
| Age of Empires | | TCP 47624 |
| AOL Instant Messenger | 5.1.3036 | TCP/UDP 5190 |
| Battle.Net (GAME) | | TCP/UDP 4000 & 6112 to 6119 |
| Bungie.Net (GAME) | | TCP 3453 |
| CUSEEME | 5.0.0.043 | TCP 7648 |
| CuteFtp | 5.0 | TCP 21 |
| DayTime RFC 867 | | TCP/UDP 13. |
| Delta Force (GAME) | | UDP 3568 |
| Diablo (GAME) | | TCP/UDP 4000 & 6112 to 6119 |
| Dialpad | | TCP 7175 |
| DirectX | 6, 7 & 8 | Version6 & 7 uses TCP 47624<br>Version 8 uses UDP 6712 |
| DNS | | TCP / UDP 53 |
| Dungeon Siege | | UDP 6712 |
| Echo | | TCP/UDP 7 |
| ESP | | ESP protocol |
| Finger | | TCP/UDP 79 |
| FTP/ FTP Server hosting | | TCP 21 |
| GNUtella | | gnutella-svc on TCP/UDP 6346.<br>And<br>gnutella-rtr on TCP/DUP 6347 |
| Application Name | Application Version | Port and Protocol |
| Gopher | | TCP/UDP 70 |
| GRE | | TCP 1723 |
| H.323 GateKeeper | | UDP 1719 |
| H.323 Terminal | | TCP 1720 |



**Supported ALGs (Sheet 2 of 3)**

| Application Name | Application Version | Port and Protocol |
|---|---|---|
| Half Life (GAME) | | UDP 27015 |
| Heretic II Server (GAME) | | TCP 28910 |
| Hexen II (GAME) | | UDP 26900 |
| HTTP/ HTTP server hosting | | TCP 80. |
| HTTPS | | TCP 443 |
| ICMP | | ICMP protocol |
| ICQ | Pro 2003b | TCP 5190 |
| IKE | | UDP 500. |
| ILS | | TCP 389 |
| IMAP | | TCP/UDP 143 |
| IMAP 3 | | TCP/UDP 220 |
| imap4-ssl | | TCP/UDP 993 |
| IRC | | TCP 6661-6669 |
| Kermit | | TCP/UDP 1649 |
| L2TP | | UDP 1701. |
| LDAP | | TCP 389 |
| Mech Commander | | UDP 6712 |
| Microsoft VPN | | TCP 1723 |
| MIRC | 6.12 | TCP 6661-6669 |
| MSN Messenger | 6.1 | TCP 1863 |
| Napster | | TCP 6699 |
| Net2Phone | | UDP 6801 |
| NetMeeting GateKeeper | | UDP 1719 |
| Netmeeting Terminal | 3.01(4.4.3396) | TCP 1720 |
| Application Name | Application Version | Port and Protocol |
| Netshow client | | TCP/UDP 1755 |
| Network Printing Protocol | | TCP 515 |
| NNTP | | TCP 119 |
| Oracle SQL*NET | | SQL*Net V1 1525/tcp<br>SQL*Net V2 1521/tcp |
| POP 2 | | TCP 109 |
| POP3 | | TCP 110 |
| PPTP | | TCP 1723 |



**Supported ALGs (Sheet 3 of 3)**

| Application Name | Application Version | Port and Protocol |
|---|---|---|
| Quake (GAME) | | TCP/UDP 26000 |
| Quake II (GAME) | | UDP 27910 |
| Quake III (GAME) | | UDP 27660 |
| QuickTime | 6 | TCP 554 |
| Quote of the Day | | TCP/UDP 17 |
| Real One Player | 2.0 | TCP 7070 |
| Real Player/ Real Media | 8 Basic (win32) Build 6.0.9.584 | TCP 7070 |
| Remote login protocol | | TCP 514 |
| Remote Mail Checking Protocol | | UDP 50 |
| Route Access protocol | | TCP/UDP 38 |
| RTSP | | TCP 554. |
| SGI Compcore | 4.0 | UDP 6301 |
| Sin (GAME) | | UDP 22450 |
| SIP | 2 | UDP 5060 |
| SMTP | | TCP 25 |
| SNMP | | UDP 161 |
| SNMP Trap | | UDP 161 |
| StarCraft (GAME) | | TCP/UDP 6112 |
| Application Name | Application Version | Port and Protocol |
| Sysstat | | TCP/UDP 11 |
| T.120 | | TCP 1503 |
| Telnet | | TCP 23 |
| Timbuktu | 4.5 | UDP 407 |
| Time | | TCP/UDP 37 |
| UUCP | | TCP 540 |
| WhoIs | | TCP 47 |
| Windows Media Player | 7 | TCP 1755 |
| Windows messenger | 4.7 | TCP 1863 |
| WinPoet PPPoE | | Unit in bridge mode. |
| XNS Time protocol | | TCP 52 |
| Yahoo Messenger | 4.1.9.993 | TCP 5050 |