

Editing the SunATM Configuration Files

5 

This chapter describes how you can configure you SunATM interfaces by editing the configuration files.

You are not required to edit these configuration files by hand. You can use the `atmadmin` configuration program, described in Section 4.3, “Using the `atmadmin` Configuration Program,” to configure the SunATM files. From the program’s command-line interface, you can change most of the SunATM parameters, with the only exception being the access list security feature described in Section 5.2, “Configuring a Classical Internet Protocol Interface” and Section 5.3, “Configuring a LAN Emulation Interface.”



Caution – When saving configuration information, `atmadmin` *will overwrite* the existing SunATM configuration files in the `/etc` directory. Therefore, any comments or other changes you manually made to the files will be lost.

5.1 Editing the /etc/atmconfig File

The /etc/atmconfig file is a generic file that must appear on every SunATM system. It provides general configuration information that is used by the SunATM startup script to bring up ATM interfaces at boot time.

The file consists of one or more entries per interface. An entry contains the following fields:

Table 5-1 /etc/atmconfig Field Descriptions

Field	Description
Interface	The physical interface, baN.
UNI/Framing	The version of the UNI specification used for signalling, 3.0 or 3.1; or, for entries with only two fields, the Framing Interface, SONET or SDH.
CIP_Host	The IP hostname used for Classical IP.
LANE_Instance	The instance number for a LAN Emulation interface; LAN Emulation interfaces will be called laneN where N is the LAN Emulation instance number. The LANE instance number must be between 0 and 999. Note: The LANE instance number is not necessarily the same as the physical instance number.
LANE_Host	The IP hostname used for LAN Emulation

The Interface and UNI fields are required for all interfaces. The CIP_Host field is required for interfaces that run Classical IP, and the LANE_Instance and LANE_Host fields are required for interfaces that run LAN Emulation. Unused fields are represented by a hyphen.

Because the atmconfig file contains information about how an interface is initially configured, the system must be rebooted in order for changes made in the /etc/atmconfig file to take effect.

5.1.1 Changing the Framing Interface in the `/etc/atmconfig` File

The framing interface defines the encapsulation method used for ATM cells as they are sent onto the wire. The default framing interface is SONET, but the SunATM software also supports the SDH interface. Your switch product information should indicate whether your switch uses either the SONET or the SDH interface.

Previous versions of the SunATM software allowed a framing interface to be chosen for the entire system (the selection was made by setting a variable in the `/etc/system` file). In the SunATM 2.1 software, the system variable may still be used to allow backwards compatibility, but the preferred method is to select the framing interface per-interface, with an entry in the `/etc/atmconfig` file. Entries in `/etc/atmconfig` will override a variable set in `/etc/system` for a particular interface. If there is no value in either `/etc/system` or `/etc/atmconfig`, the default framing interface is SONET.

Framing entries in `/etc/atmconfig` should appear on individual lines, with two fields. The first field indicates the interface, `baN`, where `N` is the instance number (for example: `ba0`). The second is either `SDH` or `SONET`, depending on the desired setting. See Figure 5-1 for an example of selecting SDH in an `/etc/atmconfig` file.

5.1.2 Example of an `/etc/atmconfig` File

The following sample `atmconfig` file creates this configuration:

- A LAN Emulation interface `lane0`, supporting UNI 3.1, on the `ba0` interface.
- An interface that supports both Classical IP and LAN Emulation on `ba1`, using UNI3.1. The LAN Emulation interface name is `lane1`.
- A Classical IP interface, supporting UNI 3.0, on `ba2`, which uses the SDH framing interface.

#Interface	UNI/Framing	CIP_Host	LANE_Instance	LANE_Host
ba0	3.1	-	0	atm0
ba1	3.1	atm1	1	atm2
ba2	3.0	atm3	-	-
ba2	SDH			

Figure 5-1 Example `/etc/atmconfig` File

5.2 Configuring a Classical Internet Protocol Interface

Classical Internet Protocol (Classical IP), specified by RFC 1577, is one way of supporting the TCP/IP and UDP/IP protocols in an ATM environment. In Classical IP, an ATM ARP server is used to resolve IP addresses to ATM addresses, replacing the traditional ARP protocol. In this configuration each host must register with the ARP server when the ATM interface is brought up. For more information on the Classical IP protocols, see Section 6.2, “Classical Internet Protocol.”

One major reason for the use of ATM ARP instead of the traditional ARP is because ATM does not support broadcast (a network capability providing transmission from one point to all points on a network). Because Classical IP over ATM does not support broadcast, you cannot use the `ypbind` UNIX command with the `-broadcast` option to automatically locate the NIS server (`ypserver`) on a Classical IP ATM subnet.

If you are planning to run NIS over your ATM network, you must specify the list of NIS servers (`ypservers`) using the `ypinit -c` command. See the `ypinit(1M)` man page for details of setting up the `ypserver`. Be sure that the IP addresses of the `ypservers` are listed in the `/etc/hosts` file.

Since Classical IP does not currently support the multicast packet delivery system, hosts cannot use the `in.rdisc` command to locate routers on the ATM subnet. The `in.rdisc` command uses IP multicasting to automatically locate routers and to pick the best router among many. Classical IP Hosts cannot use Routing Information Protocol (RIP) (`in.routed`) because RIP uses the broadcast packet delivery system. You must explicitly add the routes to the routers in the ATM subnet. You may also specify one router as the default router to provide connectivity outside of the ATM subnet. See the `route(1M)` man page for information on using the `route` command to add specific router entries and to add a default router.

5.2.1 Editing the `/etc/aarconfig` File

The `/etc/aarconfig` file is a generic file that must appear on every SunATM system which is supporting Classical IP interfaces. It allows you to specify IP to ATM address translation, permanent virtual circuits (PVCs) to destinations, and specify the address of the ATM ARP server. The environment allows for a mix of PVCs and switched virtual circuits (SVCs).

Each time the `/etc/aarconfig` file is modified, you must run the ATM ARP setup program (`aarsetup`). `aarsetup` is in the `/etc/opt/SUNWatm/bin` directory.

Every node, or client, will have both an IP address and either an ATM address or a virtual circuit identifier (VCI). See Section 6.2.1, “ATM Address Resolution,” for ATM addressing scheme information.

Table 5-3 on page 5-7 lists the flags and the options they provide. All the configuration flags are described in this section.

In the IP-ATM address table shown in the `/etc/aarconfig` file:

- `Interface` is the last part of the device name in `/dev` (`ba0`, for example).
- `Hostname` is either an IP address in “dot” notation or the name of a host that should be locally available unless a non-ATM network connection also exists.
- `ATM Address` consists of 20 octets with each octet represented by a one- or two-digit hexadecimal number and separated by colons.
- The `VCI` field is a positive decimal integer.
- An unused field is denoted by a hyphen

Table 5-2 `/etc/aarconfig` File Flags

Flag	Description
l	Represents the ATM address of the local interface on ARP clients or systems not using an ARP server for ATM address resolution, and can be used to assign an ATM address to the host. <i>Hostname</i> should not appear; <i>ATM Address</i> should be provided if and only if SVCs are used. If an <i>s</i> entry is provided to use an ARP server (see below), <i>ATM Address</i> must be provided (a server is meaningful only in an SVC environment). See Table 5-3.
L	Represents the IP and ATM address of the local interface on an ARP server. <i>Hostname</i> should not appear; <i>ATM Address</i> is required. See Table 5-3.
s	Specifies a connection to the ATM ARP server. Either <i>ATM Address</i> or <i>VCI</i> (in the case of a PVC connection) should appear, but not both. <i>Hostname</i> should not appear. The <i>s</i> entry is required on all clients that need to communicate with the server for ATM address resolution. See Table 5-3

Table 5-2 /etc/aarconfig File Flags (Continued)

Flag	Description
t	<p>Represents an IP to ATM address/VCI entry. <code>aarsetup</code> adds these entries into the local table. Any <i>t</i> entries on the server must contain <i>ATM Address</i> and may also contain <i>VCI</i> if PVC communication between the server and client is desired. In addition, there are some cases when a <i>t</i> entry may be useful on an ARP client system. If a client wants to communicate with another system over PVCs, the PVC to be used is provided in a <i>t</i> entry containing <i>VCI</i>; or if a client wishes to cache frequently used addresses to avoid frequent ARP requests, a <i>t</i> entry containing <i>ATM Address</i> may be provided. See Table 5-3.</p> <p>Note: If your naming service (NIS, DNS) server is an ATM host, you must provide the IP hostname to the address resolution for the hosts included in <i>t</i> entries, either by using the IP address in the <i>Hostname</i> field of the <i>t</i> entry, or by adding an entry to the local <code>/etc/hosts</code> file.</p>
a	<p>Represents an address that may have access to this host. If no <i>a</i> entries appear in the <code>aarconfig</code> file, access to the host is unrestricted. Including <i>a</i> entries allows access to be restricted to known hosts. As an alternative to listing individual addresses, the ATM address field may contain a prefix, followed by the wildcard <code>\$anymacsel</code>, which matches any 7-byte ESI/Selector combination following the given prefix. This allows access by any host connected to the switch specified by the given prefix. <i>Hostname</i> and <i>VCI</i> should not appear; <i>ATM Address</i> is required. See Table 5-3.</p>
m	<p>Notifies the system that the entire ATM address, including the network prefix, must be configured manually on this interface. If your interface is connected to a switch that does not support ILMI, you must include this option in your <code>/etc/aarconfig</code> or <code>/etc/lanconfig</code> file. Note that the variables <code>\$myaddress</code>, <code>\$prefix</code> and <code>\$localswitch_server</code> (which use the switch prefix obtained from the switch via ILMI) may not be used if ILMI is not running.</p>

Note – Although SunATM supports PVC connections to a server for ARP traffic, RFC 1577 does not specify this case. For interoperability with other implementations, connections to the server should use SVCs.

Note – In order for two hosts to communicate over PVCs, corresponding PVC connections must also be established in the ATM switch fabric.

Table 5-3 lists the flags and the options that they support.

Table 5-3 /etc/aarconfig File Flag Options

Interface *	Host	ATM Address	VCI	Flags	*
required	illegal	SVC only	illegal	l	local information
required	illegal	required	illegal	L	local information on server
required	illegal	required	illegal	a	access list entry
required	required	or ¹	or ¹	t	permanent table entry
required	illegal	xor ²	xor ²	s	server address/PVC
required	illegal	illegal	illegal	m	manual address registration

¹or – Means one or the other required, but using both is also legal.

²xor – Means one or the other required, but both are illegal.

Note – Entries in the `aarconfig` file must be grouped in a designated order: the local (*l* or *L*) entry must be first, the table (*t*) entries next, and then the server (*s*) entries. Other flags may appear in any order. Also, the ordering need only be maintained among entries for each physical interface; for example, all of the `ba0` entries may appear first, and then all of the `ba1` entries, etc.

5.2.2 Using Variables in the `/etc/aarconfig` File

Because the prefix portion of an ATM address specifies the ATM switch, a number of hosts specified in an `aarconfig` file may have ATM addresses which share the same prefix. To simplify setting up the `aarconfig` file, you can define variables that contain part of an ATM address.

A variable's name is an identifier consisting of a collection of no more than 32 letters, digits, and underscores. The value associated with the variable is denoted by a dollar sign followed immediately by the variable name.

Note – Variables may only be used in the ATM address field. They may not be used in any of the other fields in an entry.

Multiple variables can be concatenated to represent a single ATM address expression. A colon must be used to concatenate the variables. Thus, if one variable, `v1`, is set to `11:22` and another variable, `v2`, is set to `33:44`, the sequence `$(v1):$(v2)` represents `11:22:33:44`. Hexadecimal numbers may also be included with variables in the expression. The expression `45:$(v1):$(v2)` would have the value `45:11:22:33:44`.

Variables are defined in the `aarconfig` file according to the following format:

```
set VARIABLE = EXPRESSION
```

where `VARIABLE` is the name of a variable and `EXPRESSION` is an expression concatenating one or two-digit hexadecimal numbers and/or the values of variables that have been previously defined. The equal sign is optional, but the variable and expression must be separated either by white space (spaces or tabs), an equal sign, or both.

Several predefined variables are built into the SunATM software. These variables are summarized in Table 5-4.

Table 5-4 Predefined SunATM Variables

Variable	Description
prefix	The 13-byte prefix associated with the local switch.
mac	The 6-byte medium access control (MAC) address associated with the local host or interface.
sel	The default 1-byte selector for the local interface.
macsel	The concatenation of \$mac:\$sel.
myaddress	The concatenation of \$prefix:\$mac:\$sel, resulting in the default address for the local interface.
anymac	A wild card representing any 6-byte ESI. Should only be used in a entries.
anymacsel	A wild card representing any 7-byte ESI and Selector combination. Should only be used in a entries.
?	A wild card matching one or two hexadecimal digits within any colon-separated field. For example, \$prefix:\$anymac:? is equivalent to both \$prefix:\$anymac:?? and \$prefix:\$anymacsel. However, it is <i>not</i> the same as \$prefix:\$anymacsel:0?, which requires that the first digit of the selector byte is a 0. This wild card should only be used in a entries.
sunmacselN	The concatenation of one of a series of reserved MAC addresses and \$sel to create a block of reserved ATM ARP server addresses. N should be a decimal number in the range 0 - 199.
localswitch_server	The concatenation of \$prefix, a unique reserved MAC address, and \$sel. When used as a server address, restricts server access to clients connected to the local switch only.

Note - The \$prefix variable, and any other variables which use it (including \$myaddress and \$localswitch_server), may not be used on interfaces which are not running ILMI.

In most network configurations, the ATM address assigned to the local interface will be `$myaddress`; using this variable in the `l` entry makes it possible to use identical `aarconfig` files on all Classical IP clients using a given server.

The `sunmacselN` variables can be used in conjunction with a prefix as well as known server addresses which are not bound to a particular system. As an example, consider the case where a server that supports 50 clients fails. If the ATM address of the server is specific to that particular server, the `s` entry must be changed on all 50 clients in order to switch to a backup server. However, if the ATM address used for that server is `$prefix:$sunmacsel3`, this address is not only guaranteed to be unique, since it uses reserved medium access control (MAC) addresses, it is also possible to simply assign that address to the backup server on the same switch by changing the `l` entry to an `s` entry on one system, and bring up a new server with no changes to the clients.

Note – The `sunmacselN` variables do not include a prefix since a client and server may be on different switches and thus have different local prefix values.

In the case of a single-switch network, `localswitch_server` can be used as a well-known server address. Not only does it include the prefix associated with the local switch with a unique MAC address and appropriate Selector, it also restricts server access to clients on the local switch. Thus any host with a network prefix other than that of the local switch will be refused a connection to the ARP server if the ARP server's address is `$localswitch_server`.

Several rules apply to the use of variables in the `aarconfig` file:

1. Two variables cannot follow each other in an expression without an intervening colon. Thus `$v1:$v2` is legal while `$v1$v2` is not.
2. Fields in each line in the `aarconfig` file are separated by white space. Therefore variables should not be separated from the rest of an ATM address with whitespace. For example, `$v1: $v2` is illegal.
3. Once a variable is defined by a set command, it may not be redefined later in the `aarconfig` file.
4. The reserved variable names can not be set. These names include `prefix`, `mac`, `sel`, `macsel`, `myaddress`, `anymac`, `anymacsel`, `sunmacselN` (where `N` is a number between 0 and 199), and `localswitch_server`.

Note - The ESI portion of `localswitch_server` and the `sunmacselN` variables is a reserved MAC address. The hexadecimal values of the reserved addresses are:

```
localswitch_server      08:00:20:75:48:08
sunmacselN base        08:00:20:75:48:10
```

To calculate the ESI portion for a `sunmacselN` address, simply add the value of `N` (converted to a hexadecimal number) to the `sunmacselN` base address. For example, the ESI portion of `sunmacsel20` would be `08:00:20:75:48:10 + 0x14 = 08:00:20:75:48:24`.

5.2.3 Sample Classical IP Configurations

The following examples demonstrate entries in the `/etc/aarconfig` file for several typical network configurations.

Although some of the examples show only one sample `aarconfig` file, similarly configured files must appear on each system. Example 2 shows the files for each of the three systems in the configuration.

1. SVC-only: Clients use the default address and access to the ARP server is restricted to clients on the local switch only.

- a. The `/etc/aarconfig` file on a client:

Interface	Host	ATM Address	VCI	Flag
ba0	-	\$myaddress	-	l
ba0	-	\$localswitch_server	-	s

- b. The `/etc/aarconfig` file on the server:

Interface	Host	ATM Address	VCI	Flag
ba0	-	\$localswitch_server	-	L

2. PVC-only: *hosta* is connected to *hostb* and *hostc* over PVCs. There is no ARP server.

a. */etc/aarconfig* on *hosta*:

Interface	Host	ATM Address	VCI	Flag
ba0	-	-	-	l
ba0	hostb	-	100	t
ba0	hostc	-	101	t

b. on *hostb*:

Interface	Host	ATM Address	VCI	Flag
ba0	-	-	-	l
ba0	hosta	-	100	t
ba0	hostc	-	102	t

c. on *hostc*:

Interface	Host	ATM Address	VCI	Flag
ba0	-	-	-	l
ba0	hosta	-	101	t
ba0	hostb	-	102	t

3. SVC-only: *hosta* uses SVCs to connect to *hostb* and *hostc*. All hosts are connected to the same switch; there is no ARP server.

Interface	Host	ATM Address	VCI	Flag
ba0	-	\$myaddress	-	l
ba0	hostb	\$prefix:08:00:20:d5:08:a8:00	-	t
ba0	hostc	\$prefix:08:00:20:21:20:c3:00	-	t

4. PVC/SVC mix: *hosta* uses a SVC to connect to *hostb*, and a PVC to connect to *hostc*. *hostb* is not on the local switch; there is no ARP server.

Interface	Host	ATM Address	VCI	Flag
ba0	-	\$myaddress	-	l
ba0	hostb	45:00:00:00:00:00:00:00:0f:00:00:00:00::08:00:20:d5:08:a8:00-	-	t
ba0	hostc	-	100	t

5. ARP server: Hosts are connected to an ATM ARP server that resolves addresses. Access is restricted to the local switch subnet and one additional switch subnet.

- a. /etc/aarconfig on *hosta*:

Interface	Host	ATM Address	VCI	Flag
ba0	-	\$myaddress	-	l
ba0	-	\$prefix:\$sunmacsel0	-	s

- b. /etc/aarconfig on server:

Interface	Host	ATM Address	VCI	Flag
ba0	-	\$prefix:\$sunmacsel0	-	L
ba0	-	\$prefix:\$anymacsel	-	a
ba0	-	45:00:00:00:00:00:00:00:0f:00:00:00:00:\$anymacsel	-	a

6. Manual address configuration: Hosts are connected to a switch that does not support ILMI.

- a. /etc/aarconfig on server:

set prfx = 45:00:00:00:00:00:00:00:0f:00:00:00:00				
Interface	Host	ATM Address	VCI	Flag
ba0	-	\$prfx:\$sunmacsel	-	L

b. /etc/aarconfig on server:

```

set prfx = 45:00:00:00:00:00:00:00:0f:00:00:00
Interface Host    ATM Address          VCI    Flag
ba0       -      $prfx:$macsel       -      l
ba0       -      $prfx:$sunmacsel0   -      s

```

5.3 Configuring a LAN Emulation Interface

LAN Emulation, standardized by the ATM Forum’s LAN Emulation 1.0 specification, is another way of providing TCP/IP and UDP/IP support over an ATM interface. Address resolution information is provided by a series of LAN Emulation services. When a LAN Emulation interface is brought up, it must “join the LAN,” that is, it must register with these services. This process, and the address resolution process is described in Section 6.3, “LAN Emulation.”

Unlike Classical IP, the LAN Emulation protocol provides a broadcast service to the upper layer protocols. Therefore, the multicast and RIP limitations described in Section 5.2, “Configuring a Classical Internet Protocol Interface,” do not affect LAN Emulation interfaces.

5.3.1 Editing the /etc/laneconfig File

The /etc/laneconfig file contains the required configuration information for each interface that uses LAN Emulation. One entry is required for each SunATM interface. If you choose the automatic configuration option during the software installation, this entry will be automatically created by the installation program.

Each time you modify the /etc/laneconfig file, you must run the LAN Emulation setup program (lanesetup). lanesetup is in the /etc/opt/SUNWatm/bin directory.

The entry provides the ATM and MAC addresses which will be used by the LAN Emulation software to identify the local interface.

Each `/etc/laneconfig` entry follows this format:

Interface	MAC Address/ ELAN Name	ATM Address	VCI	Flag
-----------	---------------------------	-------------	-----	------

These entry fields are described in Table 5-5.

Table 5-5 `/etc/laneconfig` Entry Descriptions

Field	Description
Interface	Refers to the LAN Emulation interface, <code>laneN</code> .
MAC Address/ELAN Name	This field is for the 6-byte MAC address of the interface, or, in the case of an <code>n</code> entry, the name of the emulated LAN to join.
ATM Address	This field is for the 20-byte ATM address. The <code>\$myaddress</code> variable assigns the local switch prefix, local MAC address, and default selector.
VCI	The VCI field is a positive decimal integer identifying a Permanent Virtual Circuit. Place a dash in this field if VCI is not used.
Flag	This field identifies whether this entry is a local address (l), a permanent table entry (t), or an LECS address entry (c). For a complete description of the <code>laneconfig</code> flags, see Table 5-6.

Table 5-6 /etc/laneconfig Flag Descriptions

Flag	Description
l	This flag designates a local address entry. There must be an <i>l</i> entry for each interface running LAN Emulation. The interface and the ATM address must be included.
t	This flag designates a table entry for the local MAC-ATM address resolution table. If you wish to avoid the address resolution process for a frequently accessed system, for instance, you may include a <i>t</i> entry for that system; you may also create PVCs with a <i>t</i> entry. The interface, MAC address, and either ATM address or VCI (for SVC or PVC connection, respectively) must be included.
n	This flag allows you to specify the name of an emulated LAN to join. By default, the SunATM implementation will use the name provided by the LECS. If you wish to specify a different name, or if your LECS requires that a user include a name in its requests, a name may be provided with this flag. Interface is required; the name should be entered in the second field.
a	Represents an address that may have access to this host. If no <i>a</i> entries appear in the <code>laneconfig</code> file, access to the host is unrestricted. Including <i>a</i> entries allows access to be restricted to known hosts. As an alternative to listing individual addresses, the ATM address field may contain a prefix, followed by the wildcard <code>\$anymacsel</code> , which matches any 7-byte ESI/Selector combination following the given prefix. This allows access by any host connected to the switch specified by the given prefix. <i>Mac Address</i> and <i>VCI</i> should not appear; <i>ATM Address</i> is required. See Table 5-7.

Table 5-6 /etc/laneconfig Flag Descriptions (Continued)

Flag	Description
c	This flag allows an alternate LECS address to be specified. By default, the SunATM software uses the well-known address specified in the LAN Emulation standard. If, however, your LECS has a different address, or you wish to connect to the LECS over a PVC, you may provide the alternate ATM address or VCI in a c entry. If you wish to make a PVC connection, the VCI must be 17, as required by the LAN Emulation standard. The interface and ATM address or VCI must be included.
s	This flag specifies the LES address or VCI, and instructs the system to contact the LES directly, and to use default subnet configuration information. This flag should be used if your subnet does not have an LECS. Without this entry, the system first connects to the LECS, which provides the LES address and configuration information.
m	Notifies the system that the entire ATM address, including the network prefix, must be configured manually on this interface. If your interface is connected to a switch that does not support ILMI, you must include this option in your /etc/aarconfig or /etc/laneconfig file. Note that the variables \$myaddress, \$prefix and \$localswitch_server (which use the switch prefix obtained from the switch via ILMI) may not be used if ILMI is not running.

Table 5-7 describes the required, optional, and illegal fields for each flag type.

Table 5-7 laneconfig Flag Requirements and Options

Interface	MAC Address/ELAN Name	ATM Address	VCI	Flag
required	illegal	required	illegal	l
required	required	xor ¹	xor ¹	t
required	Emulated LAN name	illegal	illegal	n
required	illegal	required	illegal	a
required	illegal	xor ¹	xor	c
required	illegal	xor ¹	xor	s
required	illegal	illegal	illegal	m

¹xor means that you can use either the ATM Address field or the VCI field, but not both.

Note - Designate unused fields in the /etc/laneconfig file with a dash.

5.3.2 Using Variables in the /etc/laneconfig File

Some of the predefined variables used in the /etc/aarconfig file, may also be used in /etc/laneconfig. The applicable variables are listed in Table 5-4. For a complete description of how to use these variables, see Section 5.2.2, “Using Variables in the /etc/aarconfig File.”

Table 5-8 Predefined SunATM Variables

Variable	Description
prefix	The 13-byte prefix associated with the local switch.
mac	The 6-byte MAC address associated with the local host or interface.
sel	The default 1-byte selector for the local interface.
macsel	The concatenation of \$mac:\$sel.
myaddress	The concatenation of \$prefix:\$mac:\$sel, resulting in the default address for the local interface.
anymac	A wild card representing any 6-byte ESI. Should only be used in <i>a</i> entries.
anymacsel	A wild card representing any 7-byte ESI and Selector combination. Should only be used in <i>a</i> entries.
?	A wild card matching one or two hexadecimal digits within any colon-separated field. For example, \$prefix:\$anymac:? is equivalent to both \$prefix:\$anymac:?? and \$prefix:\$anymacsel. However, it is <i>not</i> the same as \$prefix:\$anymacsel:0?, which requires that the first digit of the selector byte is a 0. This wild card should only be used in <i>a</i> entries.

Note - The \$prefix variable, and any other variables which use it (including \$myaddress), may not be used on interfaces which are not running ILMI.

5.3.3 Sample LAN Emulation Configurations

The following examples demonstrate entries in the `/etc/laneconfig` file for several typical configurations.

Although some of the examples show only one sample `laneconfig` file, similarly configured files must appear on each LAN Emulation client.

1. Basic LAN Emulation client. The ATM and MAC address of a frequently used server is provided. The LECS provides the name of the Emulated LAN.

```
set srvr_mac = 08:00:20:01:02:03
```

Interface	MAC_Address/ELAN Name	ATM_Address	VCI	Flag
lane0	-	\$myaddress	-	l
lane0	-	\$prefix:\$srvr_mac	-	t

2. LAN Emulation client. The LECS requires that the client send the Emulated LAN name in its messages.

Interface	MAC_Address/ELAN Name	ATM_Address	VCI	Flag
lane0	-	\$myaddress	-	l
lane0	elan1	-	-	n

5.4 Supporting Logical Interfaces

A new feature in the SunATM 2.1 software is the support of logical interfaces in the LAN Emulation environment. Logical interfaces allow you to assign multiple IP addresses to a single Emulated LAN interface. A logical interface name consists of three parts: the device name (in the case of SunATM LAN Emulation, `lane`); the major number, which corresponds to the lane instance number; and the minor number, which distinguishes the logical interfaces on a single physical interface. The format of a LAN Emulation logical interface name is `laneN:X`, where `N` is the major number and `X` is the minor number.

Each logical interface will be associated with a unique IP hostname and address. All logical interfaces on a given physical interface will be associated with the same ATM and MAC addresses. Logical interfaces should be configured by placing multiple entries for a given interface in the `/etc/atmconfig` file.

The following rules and notes should be considered when using logical interfaces with the SunATM 2.1 software:

- Only one signalling protocol (for example, UNI 3.0 or 3.1) is supported per interface, and must appear in the first entry for that interface.
- Only one Classical IP hostname may be assigned to an interface; it may appear in any entry in any order in `/etc/atmconfig`.
- The first `laneN` entry on an interface must be for `laneN:0`, or simply `laneN`. `laneN` and `laneN:0` are identical and interchangeable.
- IP limits the number of logical interfaces on a physical interface to 256 (the minor number `X` must be in the range 0 - 255).

The following example shows the `atmconfig` and `laneconfig` files and the `ifconfig -a` output for a system with one physical interface, `ba0`. That interface runs both Classical IP and LAN Emulation under UNI 3.1, and has 3 different IP addresses. The IP hostnames, `cip0`, `atm0`, `atm1`, and `atm2`, should be configured appropriately in `/etc/hosts`.

The example `/etc/atmconfig` file:

Interface	UNI	CIP Hostname	LANE Instance	LANE Hostname
<code>ba0</code>	3.1	<code>cip0</code>	0	<code>atm0</code>
<code>ba0</code>	-	-	0:1	<code>atm1</code>
<code>ba0</code>	-	-	0:2	<code>atm2</code>

The corresponding example `/etc/laneconfig` file:

Interface	MAC Address/ ELAN Name	ATM Address	VC	Flag
lane0	-	\$myaddress	-	1

The resulting `ifconfig -a` output:

```
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
    inet 127.0.0.1 netmask ff000000
ba0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 9180
    inet 192.29.235.36 netmask ffffffff broadcast 192.29.235.255
    ether 8:0:20:7a:37:af
lane0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
    inet 192.29.240.36 netmask ffffffff broadcast 192.29.240.255
    ether 8:0:20:7a:37:af
lane0:1: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
    inet 192.29.241.36 netmask ffffffff broadcast 192.29.241.255
lane0:2: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
    inet 192.29.242.36 netmask ffffffff broadcast 192.29.242.255
```

5.5 Supporting Multiple Emulated LANS on a Single Interface

The SunATM 2.1 software allows a single ATM interface to join up to sixteen emulated local area networks (ELANs), provided that this is allowed by the switch and LAN Emulation (LANE) services. Each ELAN joined will be represented by a unique lane instance (e.g. lane0 or lane1).

Note – A requirement for supporting this feature is that the adapter card be assigned multiple MAC addresses, which is supported in the SunATM-622 SBus adapters and the new SunATM-155 SBus adapters 2.1. This feature *WILL NOT* work with the older SunATM SBus Adapters 2.0. You can find the number of MAC addresses assigned to your SunATM adapter by using the `atmgetmac(1M)` command with the `count` option.

The joining of multiple ELANs is configured by placing multiple entries in the `/etc/atmconfig` and `/etc/laneconfig` files. Each lane instance will have a unique IP hostname and address, ATM address, and MAC address associated with it. In addition, an ELAN name should be assigned to the instance if any

ELAN other than the default is to be joined. This information, with the exception of the MAC address, which is retrieved from the board itself, should be provided in the `/etc/atmconfig` and `/etc/laneconfig` configuration files.

Note - Only one signalling protocol (e.g. UNI 3.0 or 3.1) and one Classical IP instance are supported per physical interface. The UNI version must be specified in the first `/etc/atmconfig` entry for a given interface; the Classical IP instance may be specified in any entry.

The following example shows the `/etc/atmconfig` and `/etc/laneconfig` files and the `ifconfig -a` output for a system with one SunATM interface, `ba0`. The interface is using UNI 3.0 for signalling, and is not running Classical IP. It will join 4 emulated LANs: the default, `elan1`, `elan2`, and `elan3`.

The example `/etc/atmconfig` file:

Interface	UNI	CIP	Hostname	LANE	Instance	LANE	Hostname
ba0	3.0	-		0		atm0	
ba0	-	-		1		atm1	
ba0	-	-		2		atm2	
ba0	-	-		3		atm3	

The corresponding example `/etc/laneconfig` file:

Interface	MAC Address/ ELAN Name	ATM Address	VC	Flag
lane0	-	\$myaddress	-	l
lane1	-	\$myaddress	-	l
lane1	elan1	-	-	n
lane2	-	\$myaddress	-	l
lane2	elan2	-	-	n
lane3	-	\$myaddress	-	l
lane3	elan3	-	-	n

The resulting `ifconfig -a` output:

```
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
    inet 127.0.0.1 netmask ff000000
lane0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
    inet 192.29.240.36 netmask ffffffff broadcast 192.29.240.255
    ether 8:0:20:7a:37:af
lane1: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
    inet 192.29.241.36 netmask ffffffff broadcast 192.29.241.255
    ether 8:0:20:7a:37:b0
lane2: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
    inet 192.29.242.36 netmask ffffffff broadcast 192.29.242.255
    ether 8:0:20:7a:37:b1
lane3: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
    inet 192.29.243.36 netmask ffffffff broadcast 192.29.243.255
    ether 8:0:20:7a:37:b2
```

5.6 *Tuning Your System for Better SunATM Performance*

There are some system parameters that can greatly affect the speed and performance of your SunATM software and hardware. By adjusting these system parameters and options, you can achieve better performance out of your SunATM interfaces. For an up-to-date list of recommendations, visit the SunATM home page (<http://www.sun.com/ATM>).

