MAX TNT 2.0.0 Addendum

Ascend Communications, Inc.
Part Number: 7820-0502-003
For Software Version 2.0.0 or earlier

February 17, 1998

Ascend Customer Service

You can request assistance or additional information by telephone, email, fax, or modem, or over the Internet.

Obtaining Technical Assistance

If you need technical assistance, first gather the information that Ascend Customer Service will need for diagnosing your problem. Then select the most convenient method of contacting Ascend Customer Service.

Information you will need

Before contacting Ascend Customer Service, gather the following information:

- Product name and model.
- Software and hardware options.
- Software version.
- Service Profile Identifiers (SPIDs) associated with your product.
- Your local telephone company's switch type and operating mode, such as AT&T 5ESS Custom or Northern Telecom National ISDN-1.
- Whether you are routing or bridging with your Ascend product.
- Type of computer you are using.
- Description of the problem.
- How to contact Ascend Customer Service

After you gather the necessary information, contact Ascend in one of the following ways:

Telephone in the United States	800-ASCEND-4 (800-272-3634)
Telephone outside the United States	510-769-8027 (800-697-4772)
Austria/Germany/Switzerland	(+33) 492 96 5672
Benelux	(+33) 492 96 5674
France	(+33) 492 96 5673
Italy	(+33) 492 96 5676
Japan	(+81) 3 5325 7397
Middle East/Africa	(+33) 492 96 5679
Scandinavia	(+33) 492 96 5677
Spain/Portugal	(+33) 492 96 5675
UK	(+33) 492 96 5671
Email	support@ascend.com
Email (outside US)	EMEAsupport@ascend.com
Facsimile (fax)	510-814-2312
Customer Support BBS by modem	510-814-2302

You can also contact the Ascend main office by dialing 510-769-6001, or you can write to Ascend at the following address:

Ascend Communications
1701 Harbor Bay Parkway
Alameda, CA 94502

Need information about new features and products?

Ascend is committed to constant product improvement. You can find out about new features and other improvements as follows:

- For the latest information about the Ascend product line, visit our site on the World Wide Web:

  `http://www.ascend.com`

- For software upgrades, release notes, and addenda to this manual, visit our FTP site:

  `ftp.ascend.com`

MAX TNT 2.0.0 Addendum

Ascend Communications, Inc.
Part Number: 7820-0502-003
For Software Version 2.0.0 or earlier

February 17, 1998

Ascend Customer Service

You can request assistance or additional information by telephone, email, fax, or modem, or over the Internet.

Obtaining Technical Assistance

If you need technical assistance, first gather the information that Ascend Customer Service will need for diagnosing your problem. Then select the most convenient method of contacting Ascend Customer Service.

Information you will need

Before contacting Ascend Customer Service, gather the following information:

- Product name and model.
- Software and hardware options.
- Software version.
- Service Profile Identifiers (SPIDs) associated with your product.
- Your local telephone company's switch type and operating mode, such as AT&T 5ESS Custom or Northern Telecom National ISDN-1.
- Whether you are routing or bridging with your Ascend product.
- Type of computer you are using.
- Description of the problem.
- How to contact Ascend Customer Service

After you gather the necessary information, contact Ascend in one of the following ways:

Telephone in the United States	800-ASCEND-4 (800-272-3634)
Telephone outside the United States	510-769-8027 (800-697-4772)
Austria/Germany/Switzerland	(+33) 492 96 5672
Benelux	(+33) 492 96 5674
France	(+33) 492 96 5673
Italy	(+33) 492 96 5676
Japan	(+81) 3 5325 7397
Middle East/Africa	(+33) 492 96 5679
Scandinavia	(+33) 492 96 5677
Spain/Portugal	(+33) 492 96 5675
UK	(+33) 492 96 5671
Email	support@ascend.com
Email (outside US)	EMEAsupport@ascend.com
Facsimile (fax)	510-814-2312
Customer Support BBS by modem	510-814-2302

You can also contact the Ascend main office by dialing 510-769-6001, or you can write to Ascend at the following address:

Ascend Communications
1701 Harbor Bay Parkway
Alameda, CA 94502

Need information about new features and products?

Ascend is committed to constant product improvement. You can find out about new features and other improvements as follows:

- For the latest information about the Ascend product line, visit our site on the World Wide Web:

  `http://www.ascend.com`

- For software upgrades, release notes, and addenda to this manual, visit our FTP site:

  `ftp.ascend.com`

Contents

Contents

Introduction

What is in this addendum

The documentation that came with the MAX TNT unit describes how to install the hardware and configure the system. However, since the documentation was published, new system software has been released that contains features that are not yet included in the product documentation. This addendum describes those new features.

Related publications

Additional information is available in the MAX TNT documentation set, which consists of the following manuals:

- *The Ascend Command-Line Interface*. Shows you how to use the MAX TNT command-line interface effectively.
- *MAX TNT Hardware Installation Guide*. Describes how to install the MAX TNT hardware and use the command-line interface to configure its slot cards for a variety of supported uses. Describes how calls are routed through the system. Includes the MAX TNT technical specifications.
- *MAX TNT Network Configuration Guide*. Describes how to use the command-line interface to configure WAN connections and other related features.
- *MAX TNT RADIUS Configuration Guide.* Describes how to use RADIUS to configure WAN connections and other related features.
- *MAX TNT Reference Guide*. An alphabetic reference to all MAX TNT profiles, parameters, and commands.

New features in release 2.0.0

This section describes all features introduced in 2.0.0.

Rockwell code version

Release 2.0.0 supports Rockwell K56flex code version 1.160T.

Modified profiles

This section describes parameters that have been moved or deleted from the profiles in which they resided in previous releases.

Ethernet profile

The read-only MAC-Address and Link-State parameters have been moved from the Ethernet profile to the Ether-Info profile, which is not written to NVRAM. Ether-Info profiles are created when the card is in active state, and are deleted when the slot is brought down.

Following is an example that lists the contents of an Ether-Info profile:

```
admin> dir ether-info
     0  08/13/1997 13:34:26  { shelf-1 slot-2 1 }
     0  08/13/1997 13:34:26  { shelf-1 slot-2 2 }
     0  08/13/1997 13:34:26  { shelf-1 slot-2 3 }
     0  08/13/1997 13:34:26  { shelf-1 slot-2 4 }
     0  08/13/1997 13:34:30  { shelf-1 controller 1 }

admin> read ether-info { 1 2 1}
ETHER-INFO/{ shelf-1 slot-2 1 } read

admin> list
interface-address* = { shelf-1 slot-2 1 }
mac-address = 00:c0:7b:68:ef:98
link-state = up
```

Note: The MAC-Address and Link-State parameters no longer appear in the Ethernet profile. Because these parameters are read-only, there are no backward compatibility issues.

Frame-Relay profile

The FR-Link-Up parameter has been deleted from Frame-Relay profiles, and the corresponding RADIUS attributes Ascend-FR-LinkUp (157) and supporting values Ascend-LinkUp-Default and Ascend-LinkUp-AlwaysUp) are deprecated. Attribute 157 will be ignored in the current release, and may be reused for other purposes in a future release.

Administrators are encouraged to remove Ascend-FR-LinkUp (157) from their users file as soon as possible to avoid conflicting with other uses in the future.

New PCMCIA flash format

In releases earlier than 2.0.0, the system had to be running new shelf-controller code before you could successfully load a Tar file that contained images for new cards that were not supported in the earlier software version. To remove this requirement and facilitate future upgrades, the PCMCIA flash card format has been modified to store loads by a card type's hardware identifier.

Note: The new flash format is incompatible with the format used previously, so flash cards must be reformatted when upgrading to 2.0.0. This means that before reformatting, you must save your configured profiles to an external location.

The most recent upgrade instructions are available on the Ascend FTP server.

New dual-stage boot procedure

In previous releases, the shelf-controller code image resided in on-board flash and slot-card code resided in PCMCIA flash. In this release, both code images reside on the PCMCIA card, and a boot-loader program resides in on-board flash memory. For details about how to upgrade the system to versions including the boot-loader software, see the most recent upgrade instructions, which are available on the Ascend FTP server.

.How the boot-loader operates

The sole purpose of the boot-loader program is to load the operational code images. When the system comes up, the boot-loader initializes the shelf-controller by running POST, invoking a command-line interface on the local serial port, and enabling IP networking on the shelf-controller's built-in Ethernet port. At this point, it the boot-loader reads the System profile to determine whether it is running on a master (or standalone) shelf, or on a slave shelf.

If it is running on a master or standalone shelf, the boot-loader attempts to load the operational code for the shelf-controller by reading from a local flash card. If the attempt succeeds, it executes the operational code and brings up all slot cards. The master shelf then satisfies all code requests from slot cards and slave shelves via the Image Transfer Protocol (ITP).

If the boot-loader is running on a slave shelf, it requests the operational code for the shelf-controller from the master shelf via ITP. (If the master shelf does not have the code at that point, the ITP transfer fails and the slave does not retry until it is reset.)

What happens if the dual-stage boot fails

Generally, the dual-stage boot works so that the administrator might not even be aware that it takes place. However, if the boot procedure fails for any reason—for example, if the flash card containing the code has been removed or is corrupted, or if the master shelf is not available—the system comes up in the boot-loader state. In the boot-loader state, the system prompt is:

```
BOOT>
```

The command-line interface in this state is accessible from the shelf-controller's onboard Ethernet and Serial ports. It supports a subset of commands and profiles, so standard command such as Show and Status are not available. The following profiles are used by the boot-loader to configure the remote management communication:

- System profile
- IP-Global profile
- IP-Interface profile
- Log profile
- Serial profile

Note: Some of the parameters contained in these profiles may not be visible in the boot-loader state. Although it is possible to edit the profiles while the system is in the boot-loader state, this is not recommended. Settings for parameters supported in the operational but not the boot-loader environment would be lost if you write the profile.

In addition, in the boot-loader state, slot cards neither boot nor operate. If the failure occurred on the master of a multishelf system, all cards in the system are in the RESET state. If the failure occurred a slave shelf, only the cards on that shelf are in the RESET state.

In the boot-loader state, the following functionality is still available to the administrator:

- Coredump
- DNS
- IP routing (no OSPF)
- Syslog
- SNTP
- Fatal error log
- Telnet (inbound and outbound)
- Ping (inbound and outbound)
- TFTP and serial code load

Troubleshooting the dual-stage boot procedure

If the dual-stage boot procedure fails and the system is in the boot-loader state, first try to manually initiate the second stage of the boot procedure by resetting the system with the following command:

```
BOOT> reset
```

Note: The boot-loader does not support the –a option to the Reset command. On a multishelf system, you must manually reset each shelf, beginning with the master shelf. If the problem that caused the boot failure is corrected by resetting the master shelf, the slave shelves should not require any further action—they will come up when the master shelf is operational.

If this fails to resolve the difficulty and the system is a master shelf (or a standalone system), verify that the flash card containing the code load has not been removed from the system. If the system is a slave shelf, verify that the master shelf is available by checking the status of the master shelf and the intershelf cabling. If you have ruled out these possibilities, follow this procedure:

1 Use the Dircode command to verify that the flash card is present and contains shelf-controller operational code. For example, the Dircode output should include a line such as this:

```
BOOT> dircode
Flash card code directory:
```

```
Card 1, directory size 16
          4ether-card reg   good   141521 Jan 6 18:49      2.0.0
          32idsl-card reg   good   511317 Jan 6 18:51      2.0.0
              t3-card reg   good   205111 Jan 6 18:49      2.0.0
         48modem-card reg   good   547614 Jan 6 18:49      2.0.0
         capadsl-card reg   good   409774 Jan 6 18:50      2.0.0
           4swan-card reg   good   366848 Jan 6 18:50      2.0.0
    10-unchan-t1-card reg   good   459811 Jan 6 18:50      2.0.0
             8t1-card reg   good   177351 Jan 6 18:49      2.0.0
     shelf-controller reg   good   940282 Jan 6 18:51      2.0.0
         192hdlc-card reg   good   543436 Jan 6 18:49      2.0.0
     48modem-56k-card reg   good   556918 Jan 6 18:50      2.0.0
            sdsl-card reg   good   382271 Jan 6 18:50      2.0.0
```

2 If the shelf-controller image is *not* listed in the Dircode output, reload the Tar file and then reset. For example:

```
BOOT> load tar network host1 /vol/src/tntrel.tar

BOOT> reset
```

3 If shelf-controller code *is* listed in the Dircode output, use the Fsck command to check the flash file system. For example:

```
BOOT> fsck 1
ffs check in progress for card 1...

Dir 1 has magic, size 16, sequence 0x1d5
Dir 2 not in use

Using dir entry: 1, total data blocks: 0x40, directory size: 16

4ether-card:(0x08)
    reg   good 141521 (0x0228d1) Jan 6 18:49
32idsl-card:(0x14)
    reg   good   511317 (0x07cd55) Jan 6 18:51
t3-card:(0x0d)
    reg   good   205111 (0x032137) Jan 6 18:49
48modem-card:(0x06)
    reg   good   547614 (0x085b1e) Jan 6 18:49
capadsl-card:(0x11)
    reg   good   409774 (0x0640ae) Jan 6 18:50
4swan-card:(0x09)
    reg   good   366848 (0x059900) Jan 6 18:50
10-unchan-t1-card:(0x0b)
    reg   good   459811 (0x070423) Jan 6 18:50
8t1-card:(0x04)
    reg   good   177351 (0x02b4c7) Jan 6 18:49
shelf-controller:(0x02)
    reg   good   940282 (0x0e58fa) Jan 6 18:51
192hdlc-card:(0x07)
    reg   good   543436 (0x084acc) Jan 6 18:49
48modem-56k-card:(0x0e)
    reg   good   556918 (0x087f76) Jan 6 18:50
sdsl-card:(0x10)
    reg   good   382271 (0x05d53f) Jan 6 18:50
flash card 1 fsck: good.
```

4 If errors are reported in the Fsck output, use the Format command to reformat the flash card, and then reset the system. For example:

```
BOOT> format 1
format will erase existing card 1 data; confirm: [y/n] y
format in progress...
format complete.

BOOT> reset
```

5 When the system has reset, Telnet back into the system and load the code again. For example:

```
BOOT> load tar network host1 /vol/src/tntrel.tar
```

6 Reset again.

```
BOOT> reset
```

Multishelf features

Loadslave command for updating slave shelf code

The Loadslave command enables the administrator to update slave shelves from the master shelf interface. It uses the following syntax:

```
admin> help loadslave
LoadSlave usage: LoadSlave shelf [image (1 or 2)]

- shelf: The Slave Shelf Controller to be loaded.
- image: The low (1) or high (2) boot image of the Master.
default is image2
```

The first argument is the shelf number of a slave shelf. The second argument specifies which of two load images to use to update the specified shelf. Both load images are maintained in the master shelf-controller's NVRAM. For example, the following command updates slave shelf 3 with the code image in the master shelf-controller's high-address section of NVRAM (the default):

```
admin> loadslave 3
```

When you load a binary to the master shelf-controller via TFTP or a serial connection, the compressed image is stored in the high-address section of NVRAM, referred to as *image2* in Figure 1. When you then reset the system to execute the new shelf-controller software, the system first verifies that the compressed image is good and copies it into the low-address section of memory. The copy is referred to as *image1*. The system then decompresses image1, loads it into memory, and boots from image1.

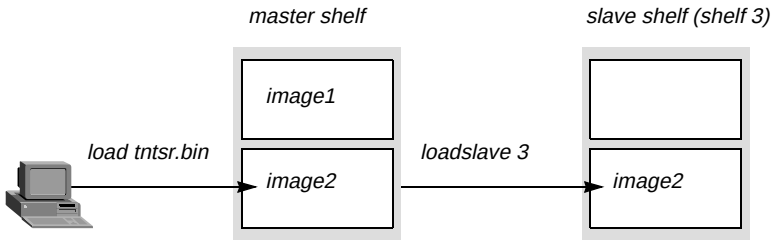

Figure 1. Loading new shelf-controller software

The slave shelf always stores the code image in the high-address section of its NVRAM (image2). However, you can specify in the Loadslave command whether you want it to load the

binary from image1 or image2 in the master shelf. The default is image2. After you reset the master shelf, both images are identical.

Reset –a to reset the mutishelf system

To reset the master shelf and all slaves in a multishelf system, append the –a flag to the Reset command. For example, while logged into the master shelf, the following command resets the entire multishelf system:

```
admin> reset –a
```

The system prompts for confirmation:

```
Reboot the entire system, dropping all connections? [y/n] y

Please stand by. System reset in progress...
```

The –a flag is not valid on slave shelves.

MP and MP+ bundled channels span HDLC cards and shelves

The MAX TNT can now bundle channels for an MP or MP+ connection across multiple HDLC cards, which may reside in different shelves of a multishelf system. The behavior of the Call-Routing-Sort-Method parameter in the System profile has been modified to enable bundling channels across HDLC cards transparently. In addition, a new TNTMP –i command enables the administrator with debug permissions to check bundles.

Changes to the Call-Routing-Sort-Method

When the system resets, the MAX TNT creates its call-routing database by sorting the list of all installed devices. (During active use, the sort order depends on system activity, but the initial sort determines the order in which the MAX TNT first uses host channels.)

If Call-Routing-Sort-Method is set to Item-First in the System profile, as shown in the following example, calls are supposed to be distributed across multiple host cards:

```
admin> read system
SYSTEM read

admin> set call-routing-sort-method = item-first

admin> write
SYSTEM written
```

However, previously, calls were not distributed evenly on multishelf systems. For example, if a multishelf system had an HDLC card in slots 1/15 and 9/2, the system created the following call-routing database after a reset:

```
admin> callroute -a
 1:15:01/1   0 0:00:00/0 digital-call-type 0 0
 1:15:01/2   0 0:00:00/0 digital-call-type 0 0
 1:15:01/3   0 0:00:00/0 digital-call-type 0 0
 ...
 1:15:01/32 0 0:00:00/0 digital-call-type 0 0
 ...
 9:02:01/1   0 0:00:00/0 digital-call-type 0 0
 9:02:01/2   0 0:00:00/0 digital-call-type 0 0
```

```
 9:02:01/3   0 0:00:00/0 digital-call-type 0 0
 ...
 9:02:01/32 0 0:00:00/0 digital-call-type 0 0
```

In this case, the first 32 calls were routed to 1/15, the next 32 calls were routed to 9/2, the next 32 calls were routed to 1/15, and so forth. In this release, if Call-Routing-Sort-Method is set to Item-First in the System profile and the system has an HDLC card in slots 1/15 and 9/2, the system creates the following call-routing database after a reset:

```
admin> callroute -a
 1:15:01/1   0 0:00:00/0 digital-call-type 0 0
 9:02:01/1   0 0:00:00/0 digital-call-type 0 0
 1:15:01/2   0 0:00:00/0 digital-call-type 0 0
 9:02:01/2   0 0:00:00/0 digital-call-type 0 0
 1:15:01/3   0 0:00:00/0 digital-call-type 0 0
 9:02:01/3   0 0:00:00/0 digital-call-type 0 0
 ...
 1:15:01/32 0 0:00:00/0 digital-call-type 0 0
 9:02:01/32 0 0:00:00/0 digital-call-type 0 0
```

The new order distributes the calls evenly across the two HDLC cards in different shelves.

The TNTMP –i command

The following example requires that debug permissions be enabled for the current User profile. It shows how the TNTMP –i command displays information about MP and MP+ bundles and their channels:

```
admin> tntmp -i
mpBundleID=13 masterSlot=1/15 masterMpID=2 ifCount=2 rtIf=1/17:6
           routeID     slot    ifNum localIfNum localMpID
                32     1/15        1          1          2
                33     9/ 2      193          1          2
```

This command works on HDLC cards as well, as shown in the following example:

```
admin> open 1 15

hdlc-1/15> tntmp -i
mpBundleID=13 masterSlot=1/15 masterMpID=2 ifCount=2 rtIf=1/17:6
           routeID     slot    ifNum localIfNum localMpID
                32     1/15        1          1          2
                33     9/ 2      193          1          2
```

In both examples, the output shows a two-channel MP or MP+ bundle with the first channel in slot 1/15 and the second (slave) channel in slot 9/2. The fields in TNTMP –i command output are explained below.

Field	Description
`mpBundleID`	The bundle ID known to the whole system. If the connection adds channels for additional bandwidth on demand, the call for those channels is compared to the current bundle and assigned the same bundle ID as the other channels of the call.

Field	Description
masterSlot	The master slot shows the channel that was established as the base channel of the connection. When the MAX TNT receives a call that is not part of an existing bundle, it authenticates the call and establishes the base channels of the connection. That channel becomes the master of the multilink connection.
masterMpID	The MpID is the bundle ID known locally to the slot card. The masterMpID field shows the MpID at the master slot card. (The masterMpID is always the same as the localMpID for channels on the master slot card.)
ifCount	The interface count shows the number of channels in the bundle.
rtIf	The rtIf field shows the shelf/slot:ID for the Route Logical Interface.
routeID	The routeID column shows the globally known ID for each call.
slot	The slot column shows the shelf/slot numbers of the channels in the MP or MP+ bundle.
localIfNum	The localIfNum is the channel number on the slot card. For HDLC cards, the channels are numbered 1–128. In the sample output, the master slot (1/15) shows channel number 1. The interface number for the slave slot (9/2) is also 1, meaning the first channel on that card. However, at the master slot card, the slave interface number is mapped to a pseudo-interface number greater than 128, to prevent its being confused with channels on the master slot.
localMpID	The MpID is the bundle ID known locally to the slot card. The localMpID field shows the MpID for channels on the local slot card.

Multishelf Show and Open commands

Changes have been made in shelf-to-shelf communications that affect how master and slave shelves communicate. In particular, the Open command now works for slot 17 (the controller) on slave shelves, and the Show command displays information about slot 17 on slave shelves.

Using the Show command

On the master shelf, the Show command output now includes slave shelf-controllers that are UP. For example:

```
Shelf 1 ( master ):
    { shelf-1 slot-1 0 }       UP       8t1-card
    { shelf-1 slot-4 0 }       UP       128hdlc-card
    { shelf-3 slot-1 0 }       UP       128hdlc-card
    { shelf-3 slot-2 0 }       UP       4ether-card
    { shelf-3 slot-3 0 }       UP       8t1-card
    { shelf-3 slot-4 0 }       UP       48modem-56k-card
    { shelf-3 slot-5 }         OCCUPIED
    { shelf-3 controller 0 }   UP       shelf-controller
```

You cannot change the state of a slave shelf-controller by using the Slot –u or Slot –d commands. If you do execute attempt to bring the slave shelf up or down by using one of these commands, the following error message appears:

```
can’t force slot 3/17 state change
```

Using the Open command

On the master shelf of a multishelf system, you can open a session with a slave shelf (for example, shelf 3) as follows:

```
admin> open 3 17
```

You can then execute commands on the slave shelf as usual, except that you cannot use the Open command from the slave shelf. If you do execute the Open command, the following error appears:

```
Can’t use open command on a slave shelf.
```

Slave shelves log reset to master’s fatal log

Slave shelves now log an Operator Reset message both to the master shelf fatal error log and to their own log. The message uses the following format:

```
OPERATOR RESET: Index: 99 Revision: 2.0.0 Shelf 9 (tntsr)
Date: 11/07/1997.       Time: 17:56:06
Reset from unknown, user profile super.
```

The shelf number indicates which slave shelf was reset. In addition, if the shelf is reset locally (from a Telnet or console session) the following message is displayed in that session:

```
Please stand by. System reset in progress...
```

If the slave shelf is reset indirectly from the master shelf, the message appears in all debug-enabled sessions.

Features for T1, E1, and T3 cards

R2 signaling support for E1

R2 signaling is an ITU-T standardized signaling protocol, which can be used on E1 digital trunks for establishing and clearing 64Kbps switched circuits. It uses a combination of A/B bit manipulation in channel 16 of the E1 frame (line signaling), and in-band MF tone generation and detection (register signaling). The relevant specifications are in ITU-T recommendations Q.400 to Q.490.

R2 signaling is widely implemented in international markets where ISDN PRI is not yet available. Ascend supports this protocol on the MAX TNT E1 platform. The following parameters in an E1 profile, shown with sample values, are relevant to R2 signaling:

```
E1 {shelf-N slot-N N}
   line-interface
      signaling-mode = e1-r2-signaling
      switch-type = switch-cas
      number-complete = 1-digits
      group-b-signal = signal-b-6
```

```
group-ii-signal = signal-ii-1
answer-delay = 200
caller-id = no-caller-id
```

Following is an example that sets these parameters for R2 signaling:

```
admin> read e1 {1 7 1}
E1/{ shelf-1 slot-7 1 } read

admin> set line signaling-mode = e1-r2-signaling

admin> set line switch-type = switch-cas

admin> set line group-b-signal = signal-b-6

admin> set line group-ii-signal = signal-ii-1

admin> write
E1/{ shelf-1 slot-7 1 } written
```

The parameters and their effects are shown below:

Parameter	Effect
Signaling-Mode	Specifies the type of signaling used on the E1 line. For R2 signaling, specify E1-R2-Signaling (R2 signaling), E1-Korean-Signaling (a version of the R2 signaling protocol specified for use in Korea), E1-P7-Signaling (P7 signaling), E1-Chinese-Signaling (a version of the R2 signaling protocol specified for use in China), or E1-Metered-Signaling (metered R2 signaling protocol, used in Brazil and South Africa).
Switch-Type	Specifies the type of network switch. For E1 R2 signaling, it should always be set to Switch-CAS.
Number-Complete	Specifies how many digits must be received of the dialed number of an incoming call using R2 signaling. You can indicate up to 10 digits of a phone number that must be received, or specify the End-of-Pulsing signal (to signify that the full number has been received). In all cases, the digits received before the call is answered are considered the called number for call-routing purposes.
Group-B-Signal Group-II-Signal	The Group B signal is the signal sent immediately before answering an incoming call. See “E1_R2 Israeli signaling” on page 16 for updated information about this parameter. The Group II signal is the signal sent in the course of an outgoing call, immediately after acknowledgment by the called end that all necessary address digits have been received. You can set the Group-B-Signal parameter to a value from B-1 through B-15, and the Group-II-Signal to a value from II-1 through II-15. For systems in Mexico and Korea, set these values to Signal-B-1 and Signal-II-2, respectively. For systems in Argentina, set these values to Signal-B-6 and Signal-II-1, respectively. For information about the proper settings for other countries, please contact your carrier.
Answer-Delay	Specifies the number of milliseconds to delay before answering a call. Use this parameter if the MAX TNT answers calls too quickly. You can set it to a number from 100 to 3000.

Parameter	Effect
Caller-ID	Used only with E1-Chinese-Signaling, to request the Calling Line ID (CLID) from the switch.

PRIdisplay command enhancements

PRIdisplay command output for a T1, E1, or T3 card now includes a time stamp relative to the time the card booted, and also includes PRI messages that have bad CRCs or are too long. In addition, the administrator can now specify a single line to be monitored.

To use the PRIdisplay command, open a session with a T1, E1, or T3 card. For example, the following command opens a session with a card in shelf 1, slot 15:

```
admin> open 1 15
```

The PRIdisplay command uses the following syntax:

```
t3-1/15> help pridisplay
priDisplay <n> [ <line> ]
     where <n> is the number of octets to display
     set <n> to zero to turn display off
     where <line> is the line whose d-channel to display
     set <line> to zero specify any line
```

In the following example, the PRIdisplay command displays the first 160 bytes of PRI messages. Notice the time stamp in each message:

```
t3-1/15> pridisplay 160
Display the first 160 bytes of PRI messages
PRI-XMIT-24: 01:38:53: 3 of 3 octets
1010A850: 00 01 7f                                              ...
PRI-RCV-24: 01:38:55: 3 of 3 octets
10112C10: 00 01 7f                                              ...
PRI-RBAD-22: 01:38:53: 2 of 2 octets
1010A850: 00 01
```

In the following example, the first PRIdisplay command displays the first 32 bytes of PRI messages for line 12 only. The second command enables display of the first 32 bytes of messages for any line on the card. The third command turns off the message display:

```
t3-1/15> prid 32 12
Display the first 32 bytes of PRI messages for line 12

t3-1/15> prid 32 0
Display the first 32 bytes of PRI messages

t3-1/15> prid 0
PRI message display terminated
```

To close the session with the card and return to the shelf-controller:

```
t3-1/15> quit
admin>
```

DS3 –i for internal loopback

The DS3link command on a T3 card now supports the –i option for performing an internal loopback for diagnostic purposes. The –i option connects the DS3 receive path to the DS3 transmit path at the D3MX. The transmitted DS3 signal is still sent to the network as well.

To use the DS3link command, open a session with a T3 card. For example, to open a session with a T3 card in shelf 1, slot 15:

```
admin> open 1 15
```

The DS3link command uses the following syntax:

```
t3-1/15> ? ds3link
ds3link get information about the DS3 interface
usage: ds3link [ - a|b|c|d|i|l|s|t|? ]
     -a display current DS3 line alarms
     -b < on | off > transmit DS3 Alarm Indication Signal
     -c display and clear line error statistics
     -d < 1 - 7 > display current DS2 line state
     -i < on | off > loop the DS3 inwards
     -l < on | off > loop the DS3 outwards
     -s display line error statistics without clearing
     -t toggle debug output
     -? display this summary
```

For example, the following command activates a DS3 internal loopback:

```
t3-1/15> ds3link -i on
DS3 internal loopback activated
```

To deactivate the DS3 internal loopback:

```
t3-1/15> ds3link -i off
DS3 internal loopback deactivated
```

Clock-Source command enhancements

The Clock-Source command on T1, E1, or T3 slot cards now lists only currently eligible local clock sources. Sources with layer 2 up, which are preferred, are marked with an asterisk. In addition, a message is now logged whenever the system clock source changes.

To use the Clock-Source command, first open a session with a T1, E1, or T3card. For example, if the T3 card is in shelf 1, slot 15:

```
admin> open 1 15
```

Then enter the Clock-Source command. The following example shows the new Clock-Source output:

```
t3-1/15> clock-source
Master line: 1
Source List:
        Source: line 1 Available*       priority: 2
        Source: line 3 Available        priority: 2
```

Following are examples of log messages generated for clock source transitions:

```
LOG notice, Shelf 1, Controller, Time: 19:44:39--
 Master clock source changed to slot-1/8 line 1
LOG notice, Shelf 1, Controller, Time: 10:34:56--
 Master clock source changed to local oscillator
```

Support for 64K and 56K calls over channels using R2 signaling

The default bandwidth for data calls coming in over E1 channels using R2 signaling is now 64K. To configure a connection to use 56K instead, use the following parameter (shown with its default setting):

```
CONNECTION station
   telco-options
      force-56kbps = no
```

Following is an example of a procedure that specifies 56K for a call coming in over channels using R2:

```
admin> read connection test
CONNECTION/test read

admin> set telco force-56kbps = yes

admin> write
CONNECTION/test written
```

Multiple NFAS groups on T1 and T3 cards

In earlier releases, the MAX TNT supported NFAS (Non-Facility Associated Signaling), a service which allows a single D-channel on one DS1 (with an optional backup D-channel) to control all B channels in some number of additional DS1s (up to 8 on a T1 card and 28 on a T3 card, subject to the limitations of the switch). Each card was limited to a single NFAS group. However, some sites require multiple NFAS groups on a single card to enabled grouped DS1s for different applications.

The MAX TNT now supports NFAS groups. An NFAS group contains a minimum of two PRIs, so a T1 card supports up to four NFAS groups, and a T3 card supports up to 14 NFAS groups. To support this configuration, the T1 profile contains the following new parameter, shown with its default value:

```
T1 { shelf-N slot-N line-N }
   nfas-group-id = 0
```

For a T1 card, you can set the NFAS-Group-ID to a value of from 0 to 3. For a T3 card, valid values are from 0 to 13. Lines with the same NFAS-Group-ID value are in the same NFAS group.

To configure multiple NFAS groups, you must set both the NFAS-Group-ID parameter and the NFAS-ID parameter for each DS1. Within the group, all PRIs share the same NFAS-Group-ID value and have different, unique NFAS-ID values.

In the following example, two NFAS groups are configured on a T1 card. Each group contains four DS1s. The example uses the NFAS group IDs 1 and 2, but you can assign any valid NFAS-Group-ID values.

Note: You must first obtain an NFAS ID for each DS1 from the Telco. Within an NFAS group, each DS1 must have a unique NFAS-ID. Telcos often use NFAS-ID=0 for the PRI with the

primary D-Channel, and NFAS-ID=1 for the PRI with the secondary D-Channel. They then assign to each PRI that does not have a D channel a unique numeric value.

Following is an example of configuring two NFAS groups on a T1 card, with each group containing four PRIs. The next group of commands configure NFAS group 1, which contains lines 1 through 4:

```
admin> read t1 {1 2 1}
T1/{ shelf-1 slot-2 1 } read
admin> set line signaling-mode = isdn-nfas
admin> set line nfas-id = 0
admin> set line nfas-group-id = 1
admin> set channel 24 channel = nfas-primary
admin> write
T1/{ shelf-1 slot-2 1 } written
admin> read t1 {1 2 2}
T1/{ shelf-1 slot-2 2 } read
admin> set line sig = isdn-nfas
admin> set line nfas-id = 1
admin> set line nfas-group-id = 1
admin> set line channel 24 channel = nfas-secondary
admin> write
T1/{ shelf-1 slot-2 2 } written
admin> read t1 {1 2 3}
T1/{ shelf-1 slot-2 3 } read
admin> set line sig = isdn-nfas
admin> set line nfas-id = 2
admin> set line nfas-group-id = 1
admin> write
T1/{ shelf-1 slot-2 3 } written
admin> read t1 {1 2 4}
T1/{ shelf-1 slot-2 4 } read
admin> set line sig = isdn-nfas
admin> set line nfas-id = 3
admin> set line nfas-group-id = 1
admin> write
T1/{ shelf-1 slot-2 4 } written
```

The following commands configure NFAS group 2, which contains lines 5 through 8:

```
admin> read t1 {1 2 5}
T1/{ shelf-1 slot-2 5 } read
admin> set line signaling-mode = isdn-nfas
admin> set line nfas-id = 0
admin> set line nfas-group-id = 2
admin> set channel 24 channel = nfas-primary
```

```
admin> write
T1/{ shelf-1 slot-2 5 } written
admin> read t1 {1 2 6}
T1/{ shelf-1 slot-2 6 } read
admin> set line sig = isdn-nfas
admin> set line nfas-id = 1
admin> set line nfas-group-id = 2
admin> set line channel 24 channel = nfas-secondary
admin> write
T1/{ shelf-1 slot-2 6 } written
admin> read t1 {1 2 7}
T1/{ shelf-1 slot-2 7 } read
admin> set line sig = isdn-nfas
admin> set line nfas-id = 2
admin> set line nfas-group-id = 2
admin> write
T1/{ shelf-1 slot-2 7 } written
admin> read t1 {1 2 8}
T1/{ shelf-1 slot-2 8 } read
admin> set line sig = isdn-nfas
admin> set line nfas-id = 3
admin> set line nfas-group-id = 2
admin> write
T1/{ shelf-1 slot-2 8 } written
```

E1_R2 Israeli signaling

The relevant specifications for this signaling type are in ITU-T recommendations Q.400 to Q.490 and Israeli MFC-R2 Register Signaling documentation. The following parameters support configuration for E1_R2 Israeli signaling, shown with their default values, which work for systems in Israel:

```
E1 {shelf-N slot-N N}
   line-interface
      group-b-answer-signal = signal-b-6
      group-b-busy-signal = signal-b-3
```

For example:

```
admin> read e1 {1 7 1}
E1/{ shelf-1 slot-7 1 } read
admin> set line group-b-answer-signal = signal-b-6
admin> set line group-b-busy-signal = signal-b-3
admin> write
E1/{ shelf-1 slot-7 1 } written
```

Group-B-Answer-Signal replaces the Group-B-Signal parameter found in earlier releases. It specifies the group-B signal that the MAX TNT sends before answering a call, and can be set

to a value from Signal-B-1 to Signal-B-15. The default is Signal-B-6, which is the recommended setting for E1_R2 Israeli signaling.

Group-B-Busy-Signal specifies the group-B signal that the MAX TNT sends before sending a busy signal, and can be set to a value from Signal-B-1 to Signal-B-15. The default is Signal-B-3, which is the recommended setting for E1_R2 Israeli signaling.

Note: When the MAX TNT does not have sufficient resources to handle the call correctly (for example, if all of its modems are busy), it sends the group-B signal specified by the Group-B-Busy-Signal parameter.

Features for modem and HDLC cards

See also “MP and MP+ bundled channels span HDLC cards and shelves” on page 7.

V.34 setting for 56k modem cards

This release supports V.34 modem modulation for the 56K modem cards. You configure this feature by setting the following parameter, which is shown with its default value:

```
TERMINAL-SERVER
   modem-configuration
      modem-mod = k56-modulation
```

To support the ITU standard V.8bis (Voice Call Ready), a 56K modem in the MAX TNT normally sends a tone at the beginning of modem training. This is commonly referred to as CRe and is a dual tone (1375Hz + 2002 Hz) followed by a single tone at 400Hz with a combined duration of approximately 500 ms. Although V.8bis is designed not to interfere with V.32bis (which supports a maximum rate of 14.4 Kbps) modem negotiation, some V.32 and V.34 modems do not successfully complete modem training after reception of the V.8bis tone.

The Modem-Mod parameter has two settings: K56-Modulation and V34-Modulation. When it is set to V34-modulation, 56K modem cards never exceed the speeds used by V.34 modems (33.6k) and do not send the V.8bis tone.

For example, the following commands configure V.34 modulation for calls coming in to 56K modem cards:

```
admin> read terminal-server
TERMINAL-SERVER read

admin> set modem-configuration modem-mod = v34-modulation

admin> write
TERMINAL-SERVER write
```

Command for displaying WAN session data

The wanDisplay and wanOpening commands already supported on host cards (modem or HDLC cards) enable the administrator to turn on or turn off a display of WAN session data as it is received and transmitted. The wanDisplay command shows WAN data for all sessions; wanOpening shows WAN data only during connection establishment. This release provides a new command, wanDSess, that shows WAN data for particular user sessions.

All of the commands display the WAN session data in the following format:

```
RECV-93:: 58 of 58 octets
1017FD44: 7e ff 7d 23 c0 21 7d 21 7d 21 7d 20 7d 3b 7d 21    ~.}#.!}! }!}
1017FD54: 7d 24 7d 25 f4 7d 22 7d 26 7d 20 7d 2a 7d 20 7d    }$}%.}"} &}
1017FD64: 20 7d 27 7d 22 7d 28 7d 22 7d 33 7d 29 7d 23 7d     }'}"}(} "}3
1017FD74: 20 c0 7b 6d 28 20 c4 7d 2c 7e                      .{m( .} ,~

XMIT-93:: 58 of 58 octets
100173F8: 7e ff 7d 23 c0 21 7d 21 7d 21 7d 20 7d 3b 7d 21    ~.}#.!}! }!}
10017408: 7d 24 7d 25 f4 7d 22 7d 26 7d 20 7d 2a 7d 20 7d    }$}%.}"} &}
10017418: 20 7d 27 7d 22 7d 28 7d 22 7d 33 7d 29 7d 23 7d     }'}"}(} "}3
10017428: 20 c0 7b 63 42 cf 7d 23 9b 7e                      .{cB.}# .~
```

The wanDSess command uses the following syntax:

```
wandsess session-name octets
```

where *session-name* is the name of a local Connection profile or a RADIUS profile, and *octets* specifies the maximum number of octets per frame. For example:

```
modem-1/7> wandsess tlynch 16

RECV-95:: 16 of 89 octets
101808DC: 7e 21 45 00 00 54 6f 2c 00 00 ff 01 81 a3 01 02    ~!E..To,
XMIT-95:: 16 of 90 octets
100138B8: 7e 21 45 00 00 54 31 f9 00 00 ff 01 be d6 c0 a8    ~!E..T1.
RECV-95:: 16 of 89 octets
1017FD44: 7e 21 45 00 00 54 71 2c 00 00 ff 01 7f a3 01 02    ~!E..Tq,
XMIT-95:: 16 of 89 octets
100138B8: 7e 21 45 00 00 54 33 f9 00 00 ff 01 bc d6 c0 a8    ~!E..T3.
```

Terminal server login timeout

When a user logs into the terminal server in terminal mode, a login prompt appears. If the user does not proceed any further than the login prompt within 300 seconds, the login times out. The administrator can now configure this timeout value by setting the following parameter, shown with its default value:

```
TERMINAL-SERVER
   terminal-mode-configuration
      login-timeout = 300
```

For example, to set the timeout to 60 seconds:

```
admin> read terminal
TERMINAL-SERVER read

admin> set terminal login-timeout = 60

admin> write
TERMINAL-SERVER written
```

If you set the Login-Timeout parameter to zero, the login never times out.

New call-routing sort method for digital calls

The following new parameter has been added to the System profile to enable the administrator to use different call-routing sort methods for digital and analog calls:

```
SYSTEM
   digital-call-routing-sort-method = slot-first
```

When Digital-Call-Routing-Sort-Method is set to Slot-First (the default), the MAX TNT sorts digital calls by shelf and slot number, and then by item number. This improves system performance for MP or MP+ calls by concentrating the channels of a call on one HDLC card.

When Digital-Call-Routing-Sort-Method is set to Item-First, the MAX TNT sorts the calls by item number, then shelf, and then slot number. This setting distributes incoming calls evenly across multiple HDLC cards. Distributing calls across cards for bundled channels creates extra processing overhead.

The Call-Routing-Sort-Method parameter in the System profile now specifies the sort method for analog calls only. It is set to Item-First by default, which means that analog calls are distributed evenly across multiple host cards.

Direct-access modem dialout over 56K modems

Direct-access dialout enables users to dial out over the MAX TNT 56K modems. You can configure the feature by setting the following parameters, shown with their default values:

```
TERMINAL-SERVER
   dialout-configuration
      enabled = no
      direct-access = no
      port-for-direct-access = 5000
      password-for-direct-access = ""
      security-for-direct-access = none
```

Following are descriptions of the Dialout-Configuration parameters:

Parameter	Description
Enabled	Controls whether modem dialout is allowed. If set to No, none of the other parameters in the Dialout-Configuration subprofile apply.
Direct-Access	Enables the direct-access dialout feature. If set to Yes, users can Telnet to a particular port on the MAX TNT to get immediate dialout service. The port number configured as the Port-for-Direct-Access tells the MAX TNT that all Telnet sessions to port want immediate modem access. If set to No, the remaining parameters in the Dialout-Configuration subprofile do not apply.
Port-for-Direct-Access	Specifies the TCP port for immediate dialout service. Must be set to an integer from 5000 to 32767 if Direct-Access is enabled. The default setting is 5000.
Security-for-Direct-Access	Specifies the type of security used for direct-access dialout. If set to Global, the Password-for-Direct-Access parameter must specify a password, which will be required from users Telneting to the specified TCP port. If Security-for-Direct-Access is set to None, no password is required for dialing out. If it is set to User, a local Connection or RADIUS profile must be configured to allow dialout.

Parameter	Description
Password-for-Direct-Access	The password (up to 64 characters) used for Global mode authentication. If Security-for-Direct-Access is not set to Global, this parameter is ignored.

Example of direct-access dialout with global password

Following is an example of setting up direct-access dialout, in this case on TCP port 5028 with a single global password (“pizza”) required for modem access:

```
admin> read terminal-server
TERMINAL-SERVER read

admin> list dialout-configuration
enabled = no
direct-access = no
port-for-direct-access = 5000
password-for-direct-access = ""
security-for-direct-access = none

admin> set enabled = yes

admin> set direct-access = yes

admin> set port = 5028

admin> set password = pizza

admin> set security = global

admin> write
TERMINAL-SERVER written
```

Example of direct-access dialout with user passwords

The commands entered in the following example set up direct-access dialout on TCP port 5000 and specify that Connection or RADIUS profiles are required for modem access:

```
admin> read terminal-server
TERMINAL-SERVER read

admin> list dialout-configuration
enabled = no
direct-access = no
port-for-direct-access = 5000
password-for-direct-access = ""
security-for-direct-access = none

admin> set enabled = yes

admin> set direct-access = yes

admin> set security = user

admin> write
TERMINAL-SERVER written
```

Example of a Connection profile for modem dialout

You can configure a Connection profile to allow modem dialout by setting the following parameters, shown with their default values:

```
CONNECTION station
   telco-options
      data-service = 56k-clear
      dialout-allowed = no
```

Following are descriptions of the Connection Telco-Options parameters related to modem dialout:

Parameter	Description
Data-Service	Specifies the bandwidth and service of outgoing calls. When the MAX TNT initiates a dialout, it requests a data service and bandwidth rate, and the answering end rate-adapts. The Modem data service setting is related only to dialouts, it is not required for inbound modem calls.
Dialout-Allowed	Specifies whether the profile may be used for dialing out on one of the MAX TNT unit's digital modems. Only if it is set to Yes will the local user be allowed to dial out on a modem to the destination specified in the Connection profile.

The following example shows how to configure the Telco-Options subprofile to allow dialout:

```
admin> read connection test
CONNECTION/test read

admin> set telco data-service = modem

admin> set telco dialout-allowed = yes

admin write
CONNECTION/test written
```

How a user dials out with direct-access

A user executes the following steps to dial out over one of the MAX TNT 56K modems when direct-access has been configured with global security:

1. Telnet to the MAX TNT from a workstation, specifying the direct-access port number on the command line. For example:

   `telnet tnt01 5000`

   Where the first argument, tnt01, is the system name of the MAX TNT and the second argument, 5000, is the direct-access port.

2. When prompted for a password, enter the Password-for-Direct-Access.

3. Use the standard Rockwell AT commands to dial out on the modem, just as if using a modem connected directly to a workstation. For example:

   `ATDT 555-1212` `^M`

4. To terminate the session with the modem, terminate the Telnet session.

Note: Direct-access dialout uses the Telnet protocol, rather than a raw TCP connection, for communicating with client processes. This means that any client process wishing to use this service to transmit or receive binary data must, at a minimum, escape outgoing IAC (0xFF) characters, handle escaped incoming IAC characters, and strip out incoming Telnet options. For a description of the Telnet protocol and how it differs from a raw TCP connection, see RFCs 854 and 855.

Performance enhancements for TCP-Clear calls

In this release, data from TCP-Clear dialup sessions that do not require V.120 processing can be buffered and transmitted as TCP packets rather than as a continuous data stream, which increases performance for these connections. In addition, unless V.120 processing is required, TCP-Clear WAN data is now sent directly to the HDLC interface rather than to the terminal-server subsystem. (If V.120 processing is required, the call is processed by the terminal server, as in previous releases.)

The system does not collect session statistics for TCP-Clear calls that make use of these enhancements.

Parameters for setting up packet buffering

Following are the parameters relevant to TCP-Clear packet buffering. The parameters are shown with their default values:

```
CONNECTION
   tcp-clear-options
      detect-end-of-packet = no
      end-of-packet-pattern = ""
      flush-length = 256
      flush-time = 20

ANSWER-DEFAULTS
   tcp-clear-answer
      detect-end-of-packet = no
      end-of-packet-pattern = ""
      flush-length = 256
      flush-time = 20
```

Detect-End-of-Packet specifies whether the MAX TNT buffers incoming WAN data. If set to Yes, after the dialup session has been authenticated, the MAX TNT begins buffering incoming WAN data until it receives the specified End-of-Packet-Pattern, or until it reaches the specified timeout (Flush-Time) or maximum packet length (Flush-Length), whichever comes first. If Detect-End-of-Packet is set to No (the default), none of the related parameters apply.

Flush-Length specifies the maximum number of bytes to buffer. Valid values are from 1 to 8192. The default value is 256. (Note that buffering large packets consumes more system resources.) If the system has buffered the specified number of bytes without matching the End-of-Packet-Pattern, it flushes the buffer by writing the data to TCP.

Flush-Time specifies a timer in milliseconds. Valid values are from 1 to 1000. The timer begins counting down on received of the first byte of buffered data. If the specified number of msecs has elapsed without matching the End-of-Packet-Pattern, the system flushes the buffer by writing the data to TCP.

End-of-Packet-Pattern defines a character pattern that signals the end of a packet. When the MAX TNT matches this pattern in the buffered data, it immediately flushes the buffer by writing all data up to and including the pattern out to TCP. Note that data may be written before a match occurs based on a specified timeout (Flush-Time) or packet length (Flush-Length).

The character pattern can be up to 64 characters long. It can contain both ASCII characters and other binary data using the backslash (\) as an escape mechanism. To insert a literal backslash in the pattern, escape it by entering two backslash characters (\\).

To insert a 1 to 3 digit octal number, escape the value using the single backslash. (To avoid confusion between the literal ASCII characters 0 through 7 and an octal value, you can pad the octal value with leading zeros.) For example, the following pattern represents a carriage return (octal 15):

```
\015
```

To insert a 1 or 2 digit hexadecimal number in the pattern, precede the number with \x. For example, the following pattern represents a carriage return (hex 0D):

```
\x0D
```

Other special escape sequences are shown below:

Escape Sequence	Description	Value
\a	Alarm	7
\b	Backspace	8
\f	Form feed	12
\n	New line	10
\r	Carriage return	13
\t	Tab	9
\v	Vertical tab	11
\\	Backslash	92
\’	Apostrophe	44
\"	Double Quote	34
\?	Wildcard	Matches any single character

Example configuration for packet buffering

The following procedure shows one example of how to set up a Connection profile to buffer WAN packets. This is a test configuration to the echo generator server of a host (port 7). The End-of-Packet-Patter is set to three hex numbers.

```
admin> read connection jim
CONNECTION/jim read

admin> set encaps = tcp-raw

admin> list tcp-clear
host = sparky
port = 7
detect-end-of-packet = no
end-of-packet-pattern = ""
flush-length = 256
flush-time = 20

admin> set detect-end-of-packet = yes

admin> set end-of-packet-pattern = \xfe\xfd\xfe

admin> set flush-length = 16

admin> write
CONNECTION/jim written
```

Features for xDSL cards

POST and loopback test for xDSL cards

An all-channel loopback test is now part of POST for SDSL and ADSL-CAP cards. In addition, SDSL and ADSL-CAP cards now support the XDSLcmd command to activate a loopback test manually. For SDSL cards, the command uses the following syntax:

```
sdsl-1/6> xdslcmd -?
usage: xdslCmd [ - l|? ][ channel ][[count] [bufferSize]]
   -l  LoopBack on Channel
   -?  display this summary
   channel: channel to test (0-15, none = all)
```

For ADSL-CAP cards, the command uses the following syntax:

```
adsl-1/16> xdslcmd -?
usage: xdslCmd [ - l|? ][ channel ][[count] [bufferSize]]
   -l  LoopBack on Channel
   -?  display this summary
   channel: channel to test (0-5, none = all)
```

The command arguments are described below:

Argument	Description
–l	Initiates a loopback on the specified channel.
channel	The channel to test. If no channel is specified, all channels are tested.
Count	The number of looped frames. The default is 10.
Bufsize	The size of the looped frames. The default is 128 bytes.

The following example shows how to run a loopback test on channel 8 of an SDSL card in shelf 1, slot 6:

```
admin> open 1 6
sdsl-1/6> xdslcmd -l 8
```

New commands for checking the IDSL card

Tests have been added to the POST functions for the IDSL card, and two new commands are supported for testing the IDSL card. To use the commands, you must first open a session with the card. For example, to open a session with an IDSL card in shelf 1, slot 7:

```
admin> open 1 7
```

IDSL command for loopback and error tests

The new IDSLcmd command uses the following syntax:

```
idsl-1/7> idslcmd
usage: idslCmd [ - 1|2|f|z|c|u|r|n|? ][ channel ]
       -1  LoopBack on B1-channel
       -2  LoopBack on B2-channel
       -a  Analog loopBack on B1-channel
```

```
        -f  Fetch Block error counters
        -z  Clear Block error counters
        -c  corrupt CRC
        -u  cancel corrupt CRC
        -r  request corrupt CRC
        -n  cancel request corrupt CRC
        -?  display this summary
channel: channel to test (0-31, none = all)
```

EOC loopback over the B channels

The first two options, 1 and 2, initiate EOC (Embedded Operations Channel) loopback on the specified B channel. EOC refers to the out-of-band mechanism available in the BRI-U interface to implement maintenance functions. It is out-of-band in that it does not use either the D or B channels but uses the maintenance bits of U-interface superframe, so it is non-intrusive. Maintenance functions include test loopbacks as well as statistics gathering (block error counters) and request to generate errors (to check that the counters work).

When you use the 1 or 2 option, the command accepts additional arguments in the following syntax:

```
idsl-1/7> idslcmd -channel [EOC address] [count] [bufsize]
```

The command arguments are described below:

Argument	**Description**
channel	The channel may be 1 or 2, representing a B channel.
EOC address	A number from 0 to 7. The default is zero, which addresses the remote TA (NT). The numbers 1 to 6 address the nodes in between, with 1 being closest to the IDSL card.The number 7 broadcasts the EOC loopback to all the nodes.
Count	The number of buffers to be sent in the loopback. The default is 10.
Bufsize	The size of the buffer to be sent. The default is 128 bytes.

For example:

```
idsl-1/7> idslcmd -1 0 64 32
```

The preceding command requests the remote TA or BRI-U device (specified by the EOC address) to go into Loopback mode over the B1 channel and sends 64 frames of size 32. A message is displayed to report the number of frames received back from the TA in which the size and the payload matches what was sent.

Analog loopback over the B channels

With the –a option, the IDSLcmd command requests an analog loopback, testing data paths between components of the card itself. The –a option requires that you specify a channel number and accepts additional arguments in the following syntax:

```
idsl-1/7> idslcmd -a channel [count] [bufsize]
```

For example:

```
idsl-1/7> idslcmd -a 1 64 32
```

The preceding command puts the U interface echo canceller (IEC-2091) in analog loopback mode and sends 64 frames of size 32 over channel B1. A message is displayed to report the number of frames received back in which the size and the payload matches. This test is adequate to verify the path between the HDLC controllers and the IEC-2091 echo canceller.

Block error fetching and error clearing

The remote U interface/echo canceller provides internal counters for far-end and near end block errors. This allows comfortable surveillance of the transmission quality at the U-interface. A block error is detected each time when the calculated checksum of the received data does not correspond to the control checksum transmitted in the successive superframe. One block error thus indicates that one U-superframe has not been transmitted correctly. No conclusion with respect to the number of bit errors is possible from the block error counters.

A near-end block error (NEBE) indicates that the error has been detected in the receive direction. A far-end block error (FEBE) identifies errors in the transmission direction. These are all from the point of view of the remote TA, since the error counters we are fetching are those of the remote TA.

With the –f option, the IDSLcmd command fetches block error counters for the specified channel. For example:

```
idsl-1/7> idslcmd -f 1
```

Block error counters are cumulative and stop accumulating once the upper limit of 65534 has been reached. To clear the block error counters for a channel, use the –z option, as shown in the next command:

```
idsl-1/7> idslcmd -z 1
```

Performing CRC error tests

To test the NEBE and FEBE counters, you can simulate transmission errors by artificially corrupting CRCs. With the –c option, the ISDLcmd command inverts CRC to purposely generate CRC errors. The remote FEBE should then increment for every corrupt frame it receives. For example, to invert CRC on channel 2:

```
idsl-1/7> idslcmd -c 2
```

To cancel this command and return CRC to normal, use the –u option. For example:

```
idsl-1/7> idslcmd -u 2
```

Conversely, you can use the –r option to request the remote TA to invert CRC. Then, the remote NEBE should increment for every corrupt frame sent. For example:

```
idsl-1/7> idslcmd -r 1
```

To cancel this command and return CRC to normal, use the –n option. For example:

```
idsl-1/7> idslcmd -n 2
```

BRIdisplay command for displaying D-channel traffic

The new BRIdisplay command uses the following syntax:

```
idsl-1/7> bridisplay
bridisplay <count> [channel]
```

The command displays up to the specified *count* number of bytes of the specified D channel. The optional channel argument specifies one of the 32 D channels of an IDSL card—its valid range is 0–31. If you specify a channel number, only traffic on that D channel is displayed, otherwise, no filter on D channels is applied and all traffic on all D channels is displayed. For example, the following command displays up to 12 bytes of the traffic in every D channel on the card:

```
idsl-1/7> bridisplay 12
```

To turn off the display, set *count* to zero; for example:

```
idsl-1/7> bridisplay 0
```

New SDSL statistics reported

The HDLC-Rx-CRC-Error-Cnt parameter has been added to the SDSL-Statistics profile to show the number of CRC errors occurring on the line.

```
admin> get sdsl-statistics { 1 3 1}
physical-address* = { shelf-1 slot-3 1 }
line-up-timer = { 0 18 45 }
rx-signal-present = yes
line-quality = 15
up-dwn-cntr = 2
self-test = passed
far-end-db-attenuation = 4
firmware-startup-stage = normal-operation
hdlc-rx-crc-error-cnt = 0
```

The value of the HDLC-Rx-CRC-Error-Cnt parameter shows how many CRC errors have occurred. It is normal to show a few CRC errors, but the line is disconnected if 1500 errors occur within 2 second time period.

New maximum down-stream rate for ADSL-CAP

In this release, the Asymmetric Digital Subscriber Line (ADSL) Carrierless Amplitude Phase (CAP) card supports the 5120000 down-stream rate configuration. Following is the relevant parameter, which is shown with its default value:

```
ADSL-CAP { shelf-N slot-N N}
   line-config
      max-down-stream-rate = 2560000
```

The Max-Down-Stream-Rate parameter already supported settings of 7168000, 5120000, and 2560000. It now also supports the additional settings in the following list. These settings represent the maximum down-stream rates the transceiver supports.

- 640000
- 960000
- 1280000
- 1600000
- 1920000
- 2240000
- 2560000
- 2688000

- 3200000
- 4480000
- 5120000
- 6272000
- 7160000

Note: The CPE maximum down-stream rate defaults to 7160000. The COE maximum down-stream rate defaults to 2560000. To adjust the down-stream rates, configure the COE ADSL-CAP profile. The loop will train to the lower of the two rates.

If there is poor loop quality, the transceiver will choose a rate lower than the maximum rate, and a good loop quality causes the transceiver to choose a rate close to or at the Maximum-Down-Stream-Rate setting. If the loop quality is very poor, the transceiver will not train at all, and will be unable to connect to the remote side. In that case, the administrator must specify a lower maximum down-stream rate, because the transceiver does not cross rate boundaries.

For example, if the transceiver is configured for 7160000bps and the loop quality is very poor to the point that the transceiver will not connect to the remote side, the transceiver does not automatically adjust the down-rate into the 5120000bps range. The administrator needs to configure the Max-Down-Stream-Rate to the lower rate.

The following example shows commands that set the maximum down-stream rate to 5.12Mbps, and the system's responses:

```
admin> read adsl-cap {1 11 1}
ADSL-CAP/{ shelf-1 slot-11 1 } read

admin> set line-config max-down-stream-rate = 5120000

admin> write
ADSL-CAP/{ shelf-1 slot-11 1 } written
```

The status of the down-stream functionality is displayed as follows int he ADSL-CAP-Status profile. For example:

```
admin> read adsl-cap-status { 1 11 1}
ADSL-CAP-STATUS/{ shelf-1 slot-11 1 } read

admin> list
physical-address* = {  shelf-1 slot-11 1 }
if-group-index = 0
unit-type = coe
dev-line-state = port-up
up-stream-rate = 952000
down-stream-rate = 5120000
major-firmware-ver = 232
minor-firmware-ver = 0
hardware-ver = 0
up-stream-constellation = 256
down-stream-constellation = 256u
down-stream-operational-baud = 680
```

The Down-Stream-Operational-Baud parameter now displays 680 and Down-Stream-Rate displays 5120000.

Note: You can set the maximum down-stream rate for using SNMP utilities by writing the DownRate object in the AdslCapLineStatusEntry MIB. The DownRate object supports Read

and Write operations, and supports the same settings as the Maximum-Down-Stream-Rate parameter.

Nailed connections supported via IDSL

In previous releases, the IDSL card supported only switched connections. A terminal adapter (TA) device, such as a Pipeline, was configured for a switched line and assigned an arbitrary phone number. In this release, the IDSL card also supports nailed connections, and the TA connection can be configured for a nailed/leased line.

You can configure only one channel on an IDSL line for nailed usage. You must also assign that channel a group number. The Connection profile to the TA then references the assigned group number in its Nailed-Groups setting, to direct the connection to use the IDSL nailed channel.

The following parameters, shown with sample settings, configure an IDSL channel for nailed usage:

```
IDSL/{ shelf-N slot-N N }
   line-interface
      enabled = yes
      channel-config N
         channel-usage = nailed-64-channel
         nailed-group = 1
```

The Enabled parameter was previously named Line-Enabled. It enables the IDSL line for use.

The Channel-Usage parameter was previously Read-Only. It specifies the usage for the channel. Channel-Usage can specify one of the following values:

- Unused-Channel specifies that the channel is unused. The MAX TNT sends the single idle code defined for the channel.
- Switched-Channel (the default) specifies a switched channel.
- Nailed-64-Channel specifies a clear-channel 64k circuit. It does not require any setup information.

The Nailed-Group parameter is new. It specifies a group number for the nailed channel (from 0 to 65535, set to zero by default). To use the nailed/leased line, the Connection profile to the TA must reference the assigned group number in its Nailed-Groups setting.

The following example shows how to configure a nailed channel on the first channel of line 18 of an IDSL card in shelf 1, slot 7:

```
admin> read idsl {1 7 18}
IDSL/{ shelf-1 slot-7 18 } read

admin> set line enabled = yes

admin> list line channel 1
channel-usage = switched-channel
nailed-group = 0

admin> set channel-usage = nailed

admin> set nailed-group = 10

admin> write
IDSL/{ shelf-1 slot-7 18 } written
```

For information about using the BRIchannels command to verify channel usage, see the *MAX TNT Hardware Installation Guide*.

Following is an example of a Connection profile that makes use of the nailed IDSL channel:

```
admin> read connection pipeline
CONNECTION/pipeline read

admin> list telco
answer-originate = ans-and-orig
callback = no
call-type = off
nailed-groups = 0
ft1-caller = no
force-56kbps = no
data-service = 56k-clear
call-by-call = 0
billing-number = ""
transit-number = ""
expect-callback = no
dialout-allowed = no
delay-callback = 0

admin> set nailed-groups = 10

admin> write
CONNECTION/pipeline written
```

Frame-Relay circuit switching on xDSL cards

Frame-Relay-Circuit mode is now enabled in Connection profiles that are connected through ports on ADSL, SDSL, and IDSL cards. For details about configuring this feature, see “Frame Relay Switching” on page 61.

IP and OSPF routing features

OSPF does not support virtual IP interfaces

The Ascend OSPF implementation conforms with RFC 1583 and does not support virtual IP interfaces. A virtual IP interface is one created by the administrator and associated with a physical LAN interface in the MAX TNT. For example, in the following listing the first port on the Ethernet card in slot 15 (shelf-1, slot-15, port 1) has three virtual interfaces:

```
admin> dir ip-int
     8  01/14/1998 14:43:14  { { shelf-1 slot-15 2 } 0 }
     8  01/14/1998 14:43:14  { { shelf-1 slot-15 3 } 0 }
     8  01/14/1998 14:43:14  { { shelf-1 slot-15 4 } 0 }
    20  01/14/1998 14:57:48  { { shelf-1 controller 1 } 0 }
    11  01/14/1998 15:24:28  { { shelf-1 slot-15 1 } 0 }
    10  02/04/1998 11:56:47  { { shelf-1 slot-15 1 } 1 }
    10  02/04/1998 11:57:01  { { shelf-1 slot-15 1 } 2 }
    10  02/04/1998 11:57:09  { { shelf-1 slot-15 1 } 3 }
```

OSPF can be enabled on any one of the port’s IP interfaces, but not on more than one interface for the same port.

IP port caching

Although IP route-caching has been implemented in the MAX TNT for some time, the route caching mechanism does not affect traffic that is being directed to the MAX TNT itself at a higher protocol layer, such as the traffic in a TCP-Clear session.

In a TCP-Clear session, a TCP connection is established between the receiving slot card for the client dial-in (such as a modem card) and a server on the IP network, which is accessible through the destination card (such as an Ethernet card). TCP packets containing the client terminal byte stream are created by the modem card and sent to the server. In this example, the packets from modem card to server can be routed via IP route-caching directly to the Ethernet card. In the reverse direction, server to client, there is no IP route cache, because the packet is destined for the MAX TNT system itself. So the packet is delivered to the router, where it is forwarded to the modem card based on the destination port number.

The following parameter, shown with its default value, enables IP packet forwarding card-to-card based on the packet destination IP address and port:

```
IP-GLOBAL
   ip-port-cache-enable = yes
```

If you set this parameter to no, packets destined for the MAX TNT itself are routed from the receiving slot card to the destination slot card through the shelf-controller, rather than being forwarded directly from the receiving slot card.

TCP-timeout parameter enabled

In this release, the TCP-timeout parameter in the IP-Global profile is enabled. TCP-timeout applies to all TCP connections initiated from the MAX TNT, including Telnet, Rlogin, TCP-clear, and the TCP portion of DNS queries. It applies to established TCP connections as well as to initial attempts to connect. For example, when a user enters a hostname using client software in a terminal server session, and DNS returns a list of IP addresses for the host, if the first address proves unreachable and the timeout on each attempt is long, the client software often times out before finding a good address. The TCP-timeout parameter enables the administrator to adjust the TCP retry timer so that each unsuccessful connection attempt will terminate quickly, allowing more rapid progress through the list, to a good address if one is present.

Following are the relevant parameters, which are shown with their default values:

```
IP-GLOBAL
   tcp-timeout = 0
   dns-list-size = 6
```

Note that the maximum value for DNS-List-Size is 35 addresses.

Valid values for TCP-timeout are from 0 to 200 seconds. At the default value (0), the system attempts a fixed number of retries at escalating intervals, adding up to about 170 seconds total. (Other limits in the system terminate TCP retries after about 170 seconds, even if the parameter is set to a higher number.) If you set TCP-timeout to a non-zero value, the value is the number of seconds TCP retries persist. After the specified number of seconds, the retries stop and the connection is considered lost.

The optimal setting for the TCP-timeout parameter must be determined by experience, and depends on the characteristics of the TCP destination (server) hosts. For example, if the

destinations are all on a LAN under the same administrative control as the MAX TNT and are lightly loaded, then a short timeout (such as a few seconds) might be reasonable, because a host that does not respond within that interval is probably down. Conversely, if the environment includes servers with longer network latency times, such as those connected across the WAN, or load is high in the network or the router, or the characteristics of the remote hosts are not well known, a longer timeout is appropriate. Values of 30 to 60 seconds are common in UNIX TCP implementations.

OSPF global option for disabling ASBR calculations

Autonomous System Boundary Routers (ASBRs) perform calculations related to external routes. Normally, when the MAX TNT imports external routes from RIP (for example, when it establishes a WAN link with a caller that does not support OSPF) it performs the ASBR calculations for those routes. Now, you can prevent the MAX TNT from performing ASBR calculations by setting the following parameter, which is shown with its default value:

```
IP-GLOBAL
   ospf-global
      as-boundary-router = yes
```

If you set the AS-Boundary-Router parameter to No, the MAX TNT does not perform ASBR calculations.

Change in ASE route handling for NSSA configurations

For Not So Stubby Areas(NSSAs), all routes imported to OSPF must have the P-bit set (P stands for *propagate*). When the P-bit is enabled, Area Border Routers translate Type-7 LSAs to Type-5 LSAs, which can then be flooded to the backbone.

When the MAX TNT is configured to route OSPF in an NSSA, all external routes that are imported to OSPF now have the P-bit enabled in their respective link-state entry. These external routes are considered Type-7 ASE LSAs. They may be routes defined in local Connection profiles or RADIUS profiles, or static routes defined in IP-Route profiles.

Note: In previous releases, the ASE7-Adv parameter in IP-Route profiles provided a way to disable the P-bit for static routes imported to OSPF in an NSSA, to prevent those routes from being propagated to the backbone. This is no longer the case. The P-bit is now always enabled for ASE routes, so the MAX TNT disregards the setting of this parameter.

Local DNS table

The MAX TNT can now maintain a DNS table in RAM of up to 8 hostnames and their IP addresses. It consults the table in RAM for address resolution only if requests to the DNS server fail. The local table acts as a safeguard to ensure that the MAX TNT can resolve the local set of DNS names in case all DNS servers become unreachable or go down.

The local DNS table is propagated to RAM from a configured DNS-Local-Table subprofile in the IP-Global profile. At system startup, the values in the profile are copied to the table in RAM. If the administrator subsequently modifies the DNS-Local-Table subprofile, the changes are propagated to the table in RAM when the profile is written.

The DNS table in RAM has space for up to 35 IP addresses per hostname entry (the limit set by the maximum DNS-List-Size). The DNS-Local-Table subprofile allows a single IP address per hostname. For related information, see “Using the Auto-Update feature” on page 34.

To set up the local DNS table, the administrator configures the following parameters in the IP-Global profile, which are shown with their default values:

```
IP-GLOBAL
   dns-local-table
      enabled = no
      auto-update = no
      table-config [1]–[8]
         host-name = ""
         ip-address = 0.0.0.0
```

Parameter	Effect
Enabled	Specifies whether the local DNS table in RAM will be available if DNS queries fail. If set to No (the default), and a DNS query times out, the request fails. If set to Yes, the MAX TNT attempts to resolve the query by consulting the DNS table in RAM. If the hostname in the DNS query has an entry in the table in RAM, the system returns the associated IP address(es) to the requester.
Auto-Update	Specifies whether the local table is automatically updated by regular successful DNS queries. For details, see “Using the Auto-Update feature” on page 34.
Table-Config *N*	The Table-Config subprofiles, numbered 1 to 8, each contain a single hostname field and a single IP address field.
Host-Name	Specifies a hostname, which must be unique within the table. For details, see “Hostname matching” next.
IP-Address	Specifies a valid IP address for the Host-Name, or the null address.

Hostname matching

The hostname specified in the DNS-Local-Table subprofile must start with an alphabetic character, and must be less than 256 characters. Trailing periods are ignored in the comparison.

The hostname may be a host name or a fully qualified name that includes the domain name. If it does not contain a domain name, and the following parameters are set to valid domain-name values in the IP-Global profile:

```
IP-GLOBAL
 domain-name = eng.abc.com
 sec-domain-name = abc.com
```

Then the system appends the specified domain name when looking up the hostname. For example, for a DNS query on the following hostname:

```
host-name = wheelers
```

The MAX TNT searches for the hostname as well as the following domain names:

```
wheelers.eng.abc.com
wheelers.abc.com
```

Defining the local table

Following is an example of configuring a local table specifying three hosts:

```
admin> read ip-global
IP-GLOBAL read

admin> list dns-local
enabled = no
auto-update = no
table-config = [ { "" 0.0.0.0 } {"" 0.0.0.0 } {"" 0.0.0.0 } {"" 0.0.0.0+

admin> set enabled = yes

admin> list table 1
host-name = ""
ip-address = 0.0.0.0

admin> set host = host1.abc.com

admin> set ip = 10.1.2.3

admin> list ..
table-config[1] = { host1.abc.com 10.1.2.3 }
table-config[2] = { "" 0.0.0.0 }
table-config[3] = { "" 0.0.0.0 }
table-config[4] = { "" 0.0.0.0 }
table-config[5] = { "" 0.0.0.0 }
table-config[6] = { "" 0.0.0.0 }
table-config[7] = { "" 0.0.0.0 }
table-config[8] = { "" 0.0.0.0 }

admin> set 2 host = host2.xyz.

admin> set 2 ip = 11.1.2.3

admin> set 3 host = localhost

admin> set 3 ip = 10.0.0.1

admin> write
IP-GLOBAL written
```

If you specify an IP address without also specifying a hostname, a message such as the following is displayed when you write the profile:

```
error: dns-local-table: host-name missing (#3 1.2.3.4)
```

If you enter an invalid hostname, a message such as the following is displayed when you write the profile:

```
error: dns-local-table: host-name must start with alpha char (#5 11foo)
```

Using the Auto-Update feature

The Auto-Update parameter determines whether the local table is updated by regular successful DNS queries. If it is set to No (the default), the contents of the local table are not affected by successful DNS queries. If set to Yes, when a regular DNS query succeeds, a lookup on that hostname is made to the local table. If there is an entry for the hostname, the entry's IP address(es) is replaced by the query response. Note that you can use the Auto-Update parameter to build the local table.

The following parameters, which are shown with their default values, affect how the table is updated when Auto-Update is set to Yes:

```
IP-GLOBAL
 dns-list-attempt = no
 dns-list-size = 6
```

If DNS-List-Attempt is set to No, a successful DNS query returns a single address for a given hostname. In the DNS table in RAM, that address is stored and the remaining 34 addresses are cleared (set to zero).

If DNS-List-Attempt is set to Yes, a successful DNS query returns the number of addresses it finds for the host, up to DNS-List-Size. In the DNS table in RAM, those addresses are stored, overwriting the configured address or the addresses retrieved from earlier DNS queries. To prevent stale entries in the table in RAM, addresses greater than DNS-List-Size are cleared at each update.

Note: If the administrator modifies the DNS-Local-Table subprofile, assigning a single address to a host, the newly configured address is propagated to the table in RAM. The first address of the hostname entry is overwritten with the configured address, and all remaining addresses are cleared. If Auto-Update is set to Yes, the next successful DNS query overwrites the configured address and restores the multiple addresses (up to DNS-List-Size).

Following is an example that configures 8 hostnames with null addresses and then sets Auto-Update to Yes. The DNS-Local-Table changes will be propagated to RAM, and successful DNS queries to the specified hostnames will build the table with up to 14 addresses for each of the hosts.

```
admin> read ip-global
IP-GLOBAL read

admin> set dns-list-attempt = yes

admin> set dns-list-size = 14

admin> list dns-local
enabled = no
auto-update = no
table-config = [ { "" 0.0.0.0 } {"" 0.0.0.0 } {"" 0.0.0.0 } {"" 0.0.0.0+

admin> set enabled = yes

admin> set auto-update = yes

admin> list table
table-config[1] = { "" 0.0.0.0 }
table-config[2] = { "" 0.0.0.0 }
table-config[3] = { "" 0.0.0.0 }
table-config[4] = { "" 0.0.0.0 }
table-config[5] = { "" 0.0.0.0 }
table-config[6] = { "" 0.0.0.0 }
table-config[7] = { "" 0.0.0.0 }
table-config[8] = { "" 0.0.0.0 }

admin> set 1 host = mercury

admin> set 2 host = venus

admin> set 3 host = earth

admin> set 4 host = mars

admin> set 5 host = jupiter

admin> set 6 host = saturn

admin> set 7 host = uranus
```

```
admin> set 8 host = neptune

admin> write
IP-GLOBAL written
```

New command to display the local table

A new DNSTAB system-level command is provided for displaying the local DNS table. The new command uses the following syntax:

```
dnstab -s [<entry number>]
```

where entry-number is an optional argument specifying a number from one to eight indicating a host entry in the local table.

In the following example, DNS-List-Size is set to 14, and 11 addresses were resolved for the system named *neptune* (entry #8 based on the sample table configuration shown in the previous section).

```
admin> nslookup neptune
Resolving host neptune.
IP address for host neptune is 11.65.212.211.
IP address for host neptune is 10.168.10.9.
IP address for host neptune is 10.168.6.14.
IP address for host neptune is 10.168.6.144.
IP address for host neptune is 11.65.212.9.
IP address for host neptune is 10.168.6.141.
IP address for host neptune is 10.168.6.143.
IP address for host neptune is 11.65.212.182.
IP address for host neptune is 10.168.6.145.
IP address for host neptune is 10.168.6.86.
IP address for host neptune is 11.65.212.23.

admin> dnstab -s 8

Local DNS Table: enabled, AutoUpdate: enabled.
Local DNS Table
   Name                          IP Address      # Reads Time of last read
   ____________________________ _______________ _______ _________________
8: "neptune"                     11.65.212.211 *       1  ---
   10.168.10.9       10.168.6.14       10.168.6.144       11.65.212.9
   10.168.6.141      10.168.6.143      11.65.212.182      10.168.6.145
   10.168.6.86       11.65.212.23      -------            -------
   -------
```

This sample output indicates that DNS-List-Size = 14. Eleven addresses are displayed, with the remaining addresses (up to the DNS-List-Size) displayed as a row of hyphens.

The first line of output indicates that Enabled and Auto-Update are both set to Yes in the IP-Global profile. The asterisk following the first IP address indicates that the entry has been updated automatically by a DNS query. The number 1 in the # Reads column indicates that the hostname has been referenced once. The Time of Last Read column shows the time the entry is used, if SNTP is in use. If SNTP is not in use, this column will contain a row of hyphens, as shown in the output above.

Parameters for handling directed broadcasts

Two parameters have been added to help the administrator defend against Denial of Service (DoS) attacks that use directed broadcast traffic. Following are the relevant parameters, which are shown with their default values:

```
IP-INTERFACE {{shelf-N slot-N N} N}
   directed-broadcast-allowed = yes

IP-GLOBAL
   icmp-reply-directed-bcast = yes
```

Directed-Broadcast-Allowed specifies whether the MAX TNT forwards directed broadcast traffic onto the interface and its network. If set to No, the system drops directed broadcast traffic, preventing it from propagating onto intermediary networks. To protect all of the LAN interfaces against DoS attacks that use directed broadcast traffic, you must set the Directed-Broadcast-Allowed parameter to No in all IP-Interface profiles.

ICMP-Reply-Directed-Bcast specifies whether the MAX TNT responds to directed-broadcast ICMP echo requests. If set to No, the system does not respond to any directed-broadcast ICMP requests. The setting of this parameter is also shown in a new Directed-Broadcast field in the output of the Ifmgr debug command.

Following is an example that sets these parameters to No:

```
admin> read ip-int {{1 c 1} 0}
IP-INTERFACE/{ { shelf-1 controller 1 } 0 } read

admin> set directed-broadcast-allowed = no

admin> write
IP-INTERFACE/{ { shelf-1 controller 1 } 0 } written

admin> read ip-global
IP-GLOBAL read

admin> set icmp-reply-directed-bcast = no

admin> write
IP-GLOBAL written
```

IPX routing for the MAX TNT

A MAX TNT configured for IPX routing enables NetWare clients and distributed Novell networks to use NetWare services across the WAN. Figure 2 shows a MAX TNT that routes IPX between WAN interfaces (connections) and a local Novell network.

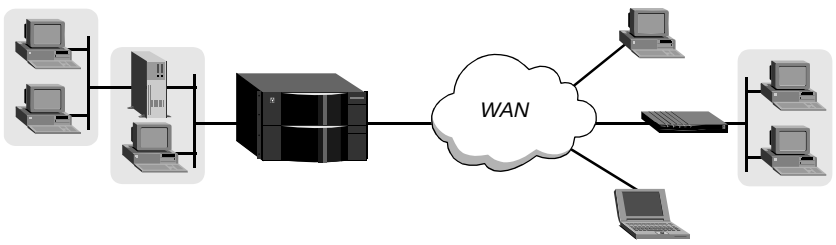

Figure 2. Routing IPX between LAN and WAN interfaces

Note: The NetWare version must be 3.11 or newer.

IPX routing on the WAN

Ascend has optimized IPX routing for the WAN, which required some modifications of standard IPX behavior as well as IPX extensions to enable the MAX TNT to operate as clients expect for NetWare LANs. This section discusses issues related to scaling LAN protocols to the WAN.

How Ascend units use IPX SAP

The MAX TNT follows standard IPX SAP behavior for routers when connecting to non-Ascend units across the WAN. However, when it connects to another Ascend unit configured for IPX routing, both ends of the connection exchange their entire SAP tables, so all remote services are immediately added to the MAX TNT unit's SAP table and vice versa.

When a NetWare client sends a SAP request to locate a service, the MAX TNT consults its SAP table and replies with its own hardware address and the internal network address of the requested server. This is analogous to proxy ARP in an IP environment. The client can then transmit packets whose destination address is the internal address of the server. When the MAX TNT receives those packets, it consults its RIP table. If it finds an entry for that destination address, it brings up the connection (unless it is already up) and forwards the packet.

How Ascend units use IPX RIP

The MAX TNT follows standard IPX RIP behavior for routers when connecting to non-Ascend units. However, when it connects to another Ascend unit configured for IPX routing, both ends of the connection immediately exchange their entire RIP tables. In addition, the MAX TNT maintains those RIP entries as static until the unit is reset or power cycled.

How IPX RIP works

IPX RIP is similar to the routing information protocol in the TCP/IP protocol suite, but it is a different protocol. IPX routers broadcast RIP updates periodically and when a WAN connection is established. The MAX TNT receives IPX RIP broadcasts from a remote device, adds 1 to the hop count of each advertised route, updates its own RIP table, and broadcasts updated RIP packets on connected networks in a split-horizon fashion.

The IPX RIP default route

The MAX TNT recognizes network number –2 (0xFFFFFFFE) as the IPX RIP default route. When it receives a packet for an unknown destination, the MAX TNT forwards the packet to the IPX router advertising the default route. If more than one IPX router is advertising the default route, the unit makes a routing decision based on Hop and Tick count. For example, if the MAX TNT receives an IPX packet destined for network 77777777 and it does not have a RIP table entry for that destination, the MAX TNT forwards the packet towards network number FFFFFFFE, if available, instead of simply dropping the packet.

Support for IPXWAN negotiation

The MAX TNT supports the IPXWAN protocol, which is essential for communicating with Novell software (such as NetWare Connect2) that supports dial-in connections, and with the

Multi-Protocol Router. For full specifications of the IPXWAN protocol, see RFC 1634 and *NetWare Link Services Protocol Specification—IPX WAN Version 2*.

When an IPX connection is brought up between two Ascend units, all options are negotiated during the IPXCP phase. IPXWAN negotiation never takes place between two Ascend units, because neither unit initiates the negotiation process by sending out an IPXWAN Timer_Request packet.

Connections with non-Ascend devices that use Novell software operating over PPP do not negotiate options during the IPXCP phase, so all options are negotiated during the IPXWAN phase of link establishment. The far-end device sends an IPXWAN Timer_Request packet, which triggers IPXWAN negotiation in the MAX TNT. The devices compare internal network numbers and assign the slave role to the unit with the lower number. The other unit becomes the master of this link for the duration of the IPXWAN negotiation. The slave unit returns an IPXWAN Timer_Response packet, and the master unit initiates an exchange of information about the final router configuration. The MAX TNT supports the following routing options:

- Ascend Routing (Unnumbered RIP/SAP without aging).
- Novell Routing (Unnumbered RIP/SAP with aging).
- None (The peer is a dial-in client. No RIP/SAP except on request and we may assign Net and Node Numbers.)

Header compression is rejected as a routing option. After IPXWAN negotiation is completed, transmission of IPX packets begins, using the negotiated routing option.

Extensions to standard IPX

NetWare uses dynamic routing and service location, so clients expect to be able to locate a server dynamically, regardless of where it is physically located. Because this scheme was designed to work in a LAN environment, not for WAN operations, Ascend provides extensions to standard IPX. The added features enhance WAN functionality, as shown in Table 1

Table 1. Ascend extensions to IPX

Ascend extension	Purpose
Virtual network for dial-in clients	To enable the MAX TNT to route IPX to non-routers (NetWare clients), it supports a virtual IPX network defined in the MAX TNT unit's IPX-Global profile. The unit can then assign a unique network address to the client. The client's connection must specify that it is a Dialin-Peer.
Accepting or rejecting RIP and SAP updates	The MAX TNT can transmit RIP and SAP updates, receive them, or both, or you can disable RIP or SAP updates for any IPX routing connection.

Table 1. Ascend extensions to IPX

Ascend extension	Purpose
Bringing up connections in response to a SAP query	The Dial-Query feature is designed for sites that support many clients and connections to only a few remote IPX networks. The MAX TNT brings up all connections that enable Dial-Query when it receives a SAP query for a file server (service type 0x04) and its SAP table has no entry for that service type.
Static routes to servers	Even though the MAX TNT learns its routes via RIP, it clears the entire RIP table when reset or powered down. Some sites configure a static IPX route to enable the MAX TNT to open a connection to that location and download the RIP table when the unit is powered up.
SAP filters	IPX SAP filters enable you to prevent the SAP table from becoming too large by explicitly including or excluding servers, services, or service types on any interface.

Recommendations for NetWare client software

NetWare clients on a WAN do not need special configuration in most cases. However, if the local network supports NetWare servers, you should configure NetWare clients with a preferred server on the local network, not at a remote site. If the local network does not support NetWare servers, configure local clients with a preferred server that is on the network with the lowest connection costs. For more information, see the NetWare documentation.

Due to possible performance issues, executing programs remotely is not recommended. For best results, put LOGIN.EXE on each client's local drive.

Both Macintosh and UNIX clients can use IPX to communicate with servers. However, both types of clients also support native protocols: AppleTalk (Macintosh) or TCP/IP (UNIX). If Macintosh clients must access NetWare servers across the WAN by using AppleTalk software (rather than MacIPX), the MAX TNT must support AppleTalk routing. Otherwise, AppleTalk packets will not make it across the connection. If UNIX clients access NetWare servers via TCP/IP (rather than UNIXWare), the MAX TNT must also be configured as an IP router. Otherwise, TCP/IP packets will not make it across the connection.

Note: Packet Burst lets servers send a data stream across the WAN before a client sends an acknowledgment. The feature is included automatically in server and client software for NetWare 3.12 or later. If local servers are running NetWare 3.11, they should have PBURST.NLM loaded. For more information, see your NetWare documentation.

IPX-Global profile settings

To create IPX-Interface profiles for routing on the MAX TNT LAN interfaces, you must enable IPX routing in the IPX-Global profile. In addition, to support dial-in NetWare clients

that are not routers, you must configure a virtual IPX network to be used for assigning IPX addresses to those clients. Following are the relevant parameters, shown with their default settings:

```
IPX-GLOBAL
   ipx-routing-enabled = yes
   ipx-dialin-pool = 12:34:56:78
```

Enabling IPX routing mode

The IPX-Routing-Enabled parameter enables IPX routing. When you write the profile, the MAX TNT comes up in IPX routing mode. At that time, it creates an IPX-Interface profile for each installed Ethernet port.

Defining a virtual IPX network for dial-in clients

When a NetWare client dials in, the MAX TNT negotiates a routing session with the client by assigning the client a node address on the virtual IPX network. The client must accept the network number defined in this pool. If it has its own node number, the MAX TNT uses that number to form the full network:node address. If the client does not have a node number, the MAX TNT assigns it a unique node address on the virtual network.

The IPX network number you assign must be unique within the entire IPX routing domain of the MAX TNT. The MAX TNT advertises the route to this virtual IPX network.

Example of an IPX-Global configuration

Following is an example of how to enable IPX routing mode and define a network for address assignment to dial-in clients that are not routers:

```
admin> read ipx-global
IPX-GLOBAL read
admin> set ipx-routing-enabled = yes
admin> set ipx-dialin = cccc1234
admin> write
IPX-GLOBAL written
```

When you write the profile, the MAX TNT comes up in IPX routing mode and creates IPX-Interface profiles for each Ethernet interface. Be sure that the network number you assign to the IPX-Dialin parameter is unique in the MAX TNT routing domain.

IPX-Interface profile settings

After you enable IPX routing in the IPX-Global profile, the system creates an IPX-Interface profile for each Ethernet interface in the system. *IPX-Interface profiles do not exist until you enable IPX routing globally.*

The IPX-Interface profiles contain the following parameters, which are shown with their default settings:

```
IPX-INTERFACE/{ { shelf-1 slot-6 2 } 0 }
   interface-address* = { { shelf-1 slot-6 2 } 0 }
   ipx-routing-enabled = no
   ipx-frame = None
```

```
ipx-net-number = 00:00:00:00
ipx-type-20 = no
ipx-sap-filter-name = ""
```

Note: IPX-Routing-Enabled must be set to Yes, and a valid IPX-Frame type must be specified, in the IPX-Interface profile for the shelf-controller Ethernet port.

Enabling IPX routing and spoofing on the interface

To enable the MAX TNT to route IPX on an Ethernet interface, you must set both the IPX-Routing-Enabled parameter and the IPX-Frame parameter. The IPX-Frame parameter specifies which IPX frame type the MAX TNT will route and spoof.

Note: The MAX TNT routes and spoofs only one IPX frame type. If some NetWare software transmits IPX in a frame type other than the type you specify, the MAX TNT drops those packets. If you are not familiar with the concept of packet frames, see the Novell documentation.

To see which frame type to use on a LAN interface, go to a NetWare server's console on that segment and type LOAD INSTALL to view the AUTOEXEC.NCF file. Following is an example AUTOEXEC.NCF line that specifies 802.3 frames:

```
Load 3c509 name=ipx-card frame=ETHERNET_8023
```

Assigning an IPX network number

If there are other NetWare routers (servers) on the LAN interface, the IPX number assigned to the MAX TNT for that interface must be consistent with the number in use by the other routers. The best way to ensure this is to leave the default null address in the IPX-Net-Number parameter. The null address causes the MAX TNT to learn its network number from another router on the interface, or from the RIP packets received from the local IPX server.

If you enter an IPX network number other than zero, the MAX TNT becomes a seed router, and other routers can learn their IPX network number from the MAX TNT. For details about seed routers, see the Novell documentation.

Propagating IPX type 20 packets on a LAN interface

Some applications, such as NetBIOS over IPX, use IPX Type 20 packets to broadcast names over a network. By default, these broadcasts are not propagated over routed links (as Novell recommends) and are not forwarded over links that have less than 1 Mbps throughput. However, if you are using an application such as NetBIOS over IPX, which requires these packets in order to operate, you can enable the router to propagate IPX Type 20 packets over a LAN interface by setting the IPX-Type-20 parameter to Yes.

Applying a SAP filter to the LAN interface

You can apply a SAP filter to a local interface by specifying the filter profile name as the value of the IPX-SAP-Filter-Name parameter. When applied to a LAN interface, a SAP filter includes or excludes specific services from the MAX TNT unit's SAP table and its responses to SAP queries on the interface. If the directory services feature is not supported, servers or services that are not in the MAX TNT table will be inaccessible to clients across the WAN. A filter applied to a LAN interface takes effect immediately.

For information about defining a SAP filter, see “IPX-SAP-Filter profile settings” on page 50. For an example that shows how to apply the filter, see “Example of applying an IPX SAP filter to a LAN interface” on page 51.

Example of an IPX-Interface configuration

Following is an example of input that enables the MAX TNT to route 802.3 IPX frames to and from the LAN interface and propagate IPX Type 20 packets:

```
admin> read ipx-int { {1 12 2 } 0 }
IPX-INTERFACE/{ { shelf-1 slot-12 2 } 0 } read

admin> set ipx-routing-enabled = yes

admin> set ipx-frame = 802.3

admin> set ipx-type-20 = yes

admin> write
IPX-INTERFACE/{ { shelf-1 slot-12 2 } 0 } written
```

Note that this example does not specify an IPX-Net-Number, which means the MAX TNT is a nonseed router that will learn its address from another IPX router on the network or from the RIP packets received from the local IPX server.

Answer-Defaults profile settings

You must set Enabled to Yes in the IPX-Answer subprofile of the Answer-Defaults profile to allow the MAX TNT to answer incoming IPX routing calls. In addition, because the MAX TNT does not have a built-in authentication mechanism (such as matching addresses) for IPX connections, they require PPP authentication.

Following are the relevant parameters, shown with their sample settings:

```
ANSWER-DEFAULTS
   ipx-answer
      enabled = yes
   ppp-answer
       receive-auth-mode = any-ppp-auth
```

Connection profile settings

Connection profiles for IPX routing connections typically use PPP authentication (described in the current *MAX TNT Network Configuration Guide*). In addition, they specify one or more of the following IPX options, which are shown with their default values:

```
CONNECTION station
   ipx-options
      ipx-routing-enabled = no
      peer-mode = router-peer
      rip = both
      sap = both
      dial-query = no
      net-number = 00:00:00:00
      net-alias = 00:00:00:00
      sap-filter = ""
      ipx-sap-hs-proxy = no
```

```
ipx-sap-hs-proxy-net = [ 0 0 0 0 0 0 ]
ipx-header-compression = no
```

Enabling IPX routing on a WAN interface

To enable IPX routing for this connection, set IPX-Routing-Enabled to Yes. If it is set to No, the MAX TNT will not route IPX packets on this interface.

Specifying whether the remote device is a router or dial-in client

The Peer-Mode parameter specifies whether the remote site is a dial-in NetWare client or another IPX router. When Peer-Mode is set to Dialin-Peer, the MAX TNT negotiates a routing session with the dial-in NetWare client by assigning the client a node address on the virtual IPX network defined in the IPX-Global profile. The client must accept the network number that is assigned. If the client has its own node number, the MAX TNT uses that number to form the full network address. If it does not have a node number, the MAX TNT assigns it a unique node address on the virtual network.

Note: When connecting to a Dialin-Peer, the MAX TNT does not send RIP and SAP advertisements across the connection, and it ignores RIP and SAP advertisements received from the far end. However, it does respond to RIP and SAP queries received from dial-in clients.

Controlling RIP and SAP updates to and from the remote router

When the remote end of the connection is a router (Peer-Mode), you can specify how RIP and SAP packets are handled across this WAN connection. Both parameters are set to Both by default, which means that the MAX TNT both sends updates across the WAN connection (informing other routers on the remote network of its routes or services), and receives updates from the remote router (including those routes or services in its RIP or SAP table).

You can set the RIP parameter to Send to cause the MAX TNT to send its routes to the remote router, but not to receive any updates on this interface. Or, you can set it to Recv to receive updates from the remote router, but not propagate the local IPX routes to the remote site. If you set it to Off, no routes are propagated in either direction.

The same settings apply to the SAP parameter. If SAP is set to both send and receive broadcasts on the WAN interface, the MAX TNT broadcasts its entire SAP table to the remote network and listens for SAP table updates from that network. Eventually, both networks have a full table of all services on the WAN. To control which services are advertised and where, you can disable the exchange of SAP broadcasts across a WAN connection, or specify that the MAX TNT will only send or only receive SAP broadcasts on that connection.

When to use net-number and net-alias

The Net-Number specifies the IPX network number of the remote-end router. This parameter, which is rarely needed, accommodates those remote-end routers that require the MAX TNT to know that router's network number before connecting.

The Net-Alias parameter may specify a second IPX network number, to be used only when connecting to non-Ascend routers that use numbered interfaces.

Applying a SAP filter to a WAN interface

You can apply a SAP filter to a WAN interface by specifying the filter profile name as the value of the SAP-Filter parameter. When applied to a WAN interface, a SAP filter includes or excludes specific services from the MAX TNT unit's SAP table and its responses to SAP queries on the interface. A filter applied to a WAN interface takes effect when the connection next becomes active.

For information about defining a SAP filter, see "IPX-SAP-Filter profile settings" on page 50. For an example that shows how to apply the filter, see "Example of applying an IPX SAP filter to a LAN interface" on page 51.

Using dial-query

Dial-Query configures the MAX TNT to bring up a connection when it receives a SAP query for service type 0x04 (File Server) and that service type is not present in the MAX TNT SAP table. If the MAX TNT has no SAP table entry for service type 0x04, it brings up every connection that has Dial Query set. For example, if 20 Connection profiles have Dial-Query set, the MAX TNT brings up all 20 connections in response to the query.

Note: If the MAX TNT unit has a static IPX route for even one remote server, it chooses to bring up that connection as opposed to the more costly solution of bringing up every connection that has Dial-Query set.

Home server proxy

For mobile NetWare clients, you can specify the network number of from one to six NetWare servers that should receive SAP queries across this connection. Without this feature, when the client is in a distant location and sends a Get Nearest Server Request query, the client receives responses from servers closer to that location, rather than the expected home server or servers. With the home-server proxy feature, mobile clients can bring up a connection to the server or servers they usually use.

To enable the home-server proxy, set the IPX-SAP-HS-Proxy parameter to Yes, and configure the IPX-SAP-HS-Proxy-Net parameter with from one to six IPX network numbers. The MAX TNT then directs the client's SAP queries only to the specified networks.

Using IPX header compression

The IPX-Header-Compression parameter specifies whether or not the MAX TNT should use IPX header compression on this connection if the specified encapsulation method supports it.

Example of a connection between two Novell LANs

Figure 3 shows a MAX TNT providing a connection between an IPX network, which supports both servers and clients, and a remote site that also supports both servers and clients, and an Ascend unit.

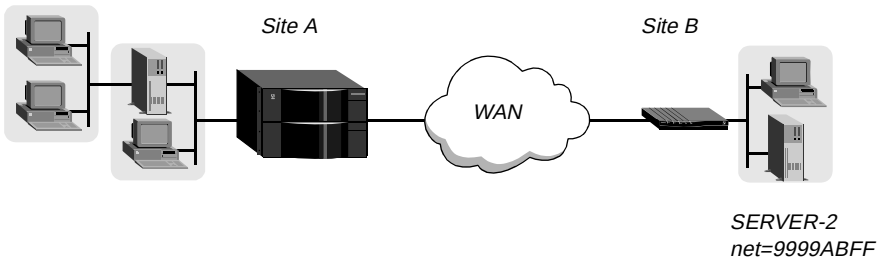

Figure 3. A connection with NetWare servers on both sides

In this example, the NetWare server at site B is configured with the following specifications:

```
Name=SERVER-2
internal net 013DE888
Load 3c509 name=net-card frame=ETHERNET_8023
Bind ipx net-card net=9999ABFF
```

Following is an example of specifying a connection to the Ascend unit at Site B:

```
admin> new conn sitebgw
CONNECTION/sitebgw read

admin> set active = yes

admin> set ppp recv-password = sitebpw

admin> list ipx
ipx-routing-enabled = no
peer-mode = router-peer
rip = both
sap = both
dial-query = no
net-number = 00:00:00:00
net-alias = 00:00:00:00
sap-filter = ""
ipx-sap-hs-proxy = no
ipx-sap-hs-proxy-net = [ 0 0 0 0 0 0 ]
ipx-header-compression = no

admin> set peer = router

admin> set rip = off

admin> write
CONNECTION/sitebgw written
```

Because RIP is turned off, you might want to create a static route to the server at the remote site, to ensure that the MAX TNT can bring up this connection, even immediately after a system reset. The following example shows how to configure a route to Server-2 at Site B:

```
admin> new ipx-route SERVER-2
IPX-ROUTE/SERVER-2 read

admin> set server-type = 0004

admin> set dest-network = 013DE888

admin> set server-node = 000000000001

admin> set server-socket = 0451

admin> set profile-name = sitebgw
```

```
admin> write
IPX-ROUTE/SERVER-2 written
```

Note: The destination network number is the server's internal network number. For more information about IPX routes, see "IPX-Route profile settings" on page 48.

Example of a connection to a dial-in client

Figure 4 shows a NetWare client dialing into a corporate IPX network using PPP software.

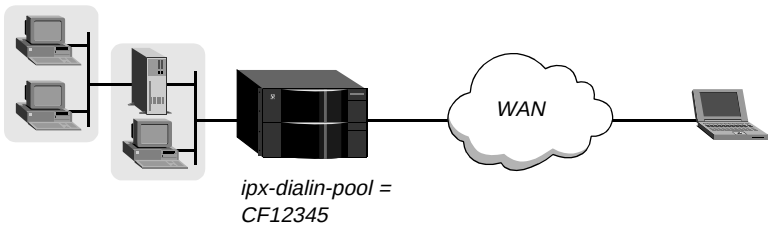

Figure 4. A dial-in NetWare client

Dial-in NetWare clients do not have an IPX network address. To have an IPX routing connection to the local network, the clients must dial in using PPP software, and the Connection profile must set Peer-Mode to Dialin-Peer. In addition, the MAX TNT must have a virtual IPX network defined for assignment to these clients. For information about defining a virtual IPX network, see "IPX-Global profile settings" on page 40.

Following is an example of input that configures an IPX routing connection for the client shown in Figure 4:

```
admin> new conn client
CONNECTION/client read

admin> set ppp recv-password = client-pw

admin> list ipx
ipx-routing-enabled = no
peer-mode = router-peer
rip = both
sap = both
dial-query = no
net-number = 00:00:00:00
net-alias = 00:00:00:00
sap-filter = ""
ipx-sap-hs-proxy = no
ipx-sap-hs-proxy-net = [ 0 0 0 0 0 0 ]
ipx-header-compression = no

admin> set ipx-routing = yes

admin> set peer = dialin

admin> write
CONNECTION/client written
```

Example of enabling home-server proxy

Following is an example of how to enable the home-server proxy feature in an IPX-routing Connection profile:

```
admin> read conn ipxclient
CONNECTION/ipxclient read

admin> list ipx
ipx-routing-enabled = no
peer-mode = router-peer
rip = both
sap = both
dial-query = no
net-number = 00:00:00:00
net-alias = 00:00:00:00
sap-filter = ""
ipx-sap-hs-proxy = no
ipx-sap-hs-proxy-net = [ 0 0 0 0 0 0 ]
ipx-header-compression = no

admin> set ipx-sap-hs-proxy = yes

admin> set ipx-sap-hs-proxy-net 1 = ccff1234

admin> write
CONNECTION/ipxclient written
```

Setting IPX-SAP-HS-Proxy to Yes enables the feature. You must then specify at least one (and up to six) IPX network addresses to which SAP broadcasts will be directed.

IPX-Route profile settings

When the MAX TNT is reset or power cycled, it clears its RIP and SAP tables from memory. Static routes create entries in new RIP and SAP tables as the unit initializes. The static routes enable the MAX TNT to reach a NetWare server and download more complete tables from there.

In the case where a MAX TNT is connecting to another Ascend unit, you might choose not to configure any static routes. However, that means that after a power-cycle or reset, you must dial the initial IPX routing connection manually. After that connection is established, the MAX TNT downloads the RIP table from the other Ascend unit and maintains the routes as static until its next power-cycle or reset.

The disadvantage of static routes is that they require manual updating whenever the specified server is removed or has an address change. Their advantages are that they ensure that the MAX TNT can bring up the connection in response to clients' SAP requests, and they help to prevent timeouts when a client takes a long time to locate a server on the WAN.

Note: You do not need to create IPX routes to servers that are on the local Ethernet.

Static IPX routes use the following parameters, which are shown with their default settings:

```
IPX-ROUTE name
   name* = name
   server-type = 00:00
   dest-network = 00:00:00:00
   server-node = 00:00:00:00:00:00
   server-socket = 00:00
   hops = 8
   ticks = 12
   profile-name = ""
   active-route = yes
```

Identifying the target

The service type is a number included in SAP advertisements. For example, NetWare file servers are SAP Service type 0x04.

The destination of an IPX route is the internal network of a server. For example, NetWare file servers are assigned an internal IPX network number by the network administrator and typically use the default node address of 000000000001. This is the destination network address for file read/write requests. (If you are not familiar with internal network numbers, see your NetWare documentation for details.)

Typically, Novell file servers use socket 0x451. The number you specify must be a well-known socket number. Services that use dynamic socket numbers may use a different socket each time they load, and will not work with IPX Route profiles. To bring up a connection to a remote service that uses a dynamic socket number, specify a master server with a well-known socket number on the remote network.

Specifying how to get to the server's network

To reach the remote server's network, the default hop count of 2 and tick count of 12 are usually appropriate, but you might need to increase these values for very distant servers. Ticks are IBM PC clock ticks (1/18 second). Note that best routes are calculated on the basis of on tick count, not hop count.

The Profile-Name parameter specifies the Connection profile to use. When the MAX TNT receives a query for the specified server or a packet addressed to that server, it finds the referenced Connection profile and dials the connection.

Activating the route

For the MAX TNT to use this route, the Active-Route parameter must be set to Yes.

Example of a static IPX route

The following example shows how to create a new IPX-Route profile for a remote server named Server-1.

```
admin> new ipx-route Server-1
IPX-ROUTE/Server-1 read
admin> set server-type = 0004
admin> set dest-network = cc1234ff
admin> set server-node 1 = 000000000001
admin> set server-socket = 0451
admin> set profile-name = sitebgw
admin> write
IPX-ROUTE/Server-1 read
```

IPX-SAP-Filter profile settings

IPX SAP filters include or exclude services from the MAX TNT SAP table. You can prevent the MAX TNT from sending its SAP table or receiving a remote site's SAP table by turning off IPX SAP in a Connection profile.

Each filter contains up to eight Input filters and Output filters, numbered from 1 to 8, which are defined individually and applied in order (1–8) to the packet stream.

The MAX TNT applies input filters to all SAP packets received by the MAX TNT. Input filters screen advertised services and exclude them from (or include them in) the MAX TNT service table as specified by the filter conditions.

The MAX TNT applies output filters to SAP response packets it transmits. If it receives a SAP request packet, the MAX TNT applies output filters before transmitting the SAP response, and excludes services from (or includes them in) the response packet as specified by the filter conditions.

Following are the subprofiles and parameters used to define a SAP filter, shown with their default values:

```
IPX-SAP-FILTER ipx-sap-filter-name
   ipx-sap-filter-name* = ipx-sap-filter-name
   input-ipx-sap-filters
      input-ipx-sap-filters [1–8]
         valid-filter = no
         type-filter = exclude
         server-type = 00:00
         server-name = ""
   output-ipx-sap-filters
      output-ipx-sap-filters [1–8]
         valid-filter = no
         type-filter = exclude
         server-type = 00:00
         server-name = ""
```

Each of the eight input and output filters include the same parameters.

Activating an input or output filter for use

The Valid-Filter parameter enables the input or output filter for use. If it is set to No, the MAX TNT skips the filter when it applies the entire IPX SAP filter to SAP data.

Specifying the action to take

The Type-Filter parameter specifies whether this filter will explicitly include the service in the SAP table or SAP response packets, or will explicitly exclude the service. The Include setting is typically used to include a specific service when previous input or output filters have excluded a general type of service. Exclude indicates a specific service to filter out of the SAP table or SAP response packets.

Identifying the service to be filtered

Server-Type specifies a hexadecimal number representing a type of NetWare service. For example, the number for file services is 0x04. In an output filter, the Server-Type parameter

specifies whether to include or exclude advertisements for this service type in SAP response packets. In an Input filter, the Server-Type parameter specifies whether to include or exclude services of this type in the MAX TNT service table.

If you specify a Server-Name, you can specify a local or remote NetWare server. If the server is on the local network and you are specifying an output filter, the Server-Name parameter specifies whether to include or exclude advertisements for this server in SAP response packets. If the server is on the remote IPX network and you are specifying an input filter, the Server-Name parameter specifies whether to include or exclude this server in the MAX TNT service table.

Example of defining an IPX SAP filter

For background information about SAP filters, see “IPX-SAP-Filter profile settings” on page 50. The following example shows how to create a SAP filter that prevents local NetWare users from accessing a remote NetWare server, by excluding it from the MAX TNT SAP table:

```
admin> new ipx-sap-filter server_1
IPX-SAP-FILTER/server_1 read

admin> list
ipx-sap-filter-name* = no-server1
input-ipx-sap-filters = [ { no exclude 00:00 "" } { no exclude 00:00 ""+
output-ipx-sap-filters = [ { no exclude 00:00 "" } { no exclude 00:00 "+

admin> list input 1
valid-filter = no
type-filter = exclude
server-type = 00:00
server-name = ""

admin> set valid-filter = yes

admin> set server-type = 0409

admin> set server-name = server_1

admin> list
valid-filter = yes
type-filter = exclude
server-type = 04:09
server-name = server_1

admin> write
IPX-SAP-FILTER/server_1 written
```

Example of applying an IPX SAP filter to a LAN interface

Following is an example of applying the SAP filter created in the previous section:

```
admin> read ipx-interface { {1 12 2 } 0 }
IPX-INTERFACE/{ { shelf-1 slot-12 2 } 0 } read

admin> set ipx-sap-filter-name = server_1

admin> write
IPX-INTERFACE/{ { shelf-1 slot-12 2 } 0 } written
```

For background information, see “IPX-Interface profile settings” on page 41.

Example of applying an IPX SAP filter to a WAN interface

Following is an example of applying the SAP filter created in the previous section:

```
admin> read conn client
CONNECTION/client read

admin> list ipx
ipx-routing-enabled = yes
peer-mode = dialin-peer
rip = both
sap = both
dial-query = no
net-number = 00:00:00:00
net-alias = 00:00:00:00
sap-filter = ""
ipx-sap-hs-proxy = no
ipx-sap-hs-proxy-net = [ 0 0 0 0 0 0 ]
ipx-header-compression = no

admin> set sap-filter = server_1

admin> write
CONNECTION/client written
```

AppleTalk routing and remote access

A MAX TNT configured for AppleTalk routing enables dial-in connections from AppleTalk Remote Access (ARA) client software, PPP dial-in software that supports AppleTalk, and AppleTalk-enabled Ascend units. Figure 5 shows a MAX TNT that routes AppleTalk between WAN interfaces (connections) and a local AppleTalk interface.

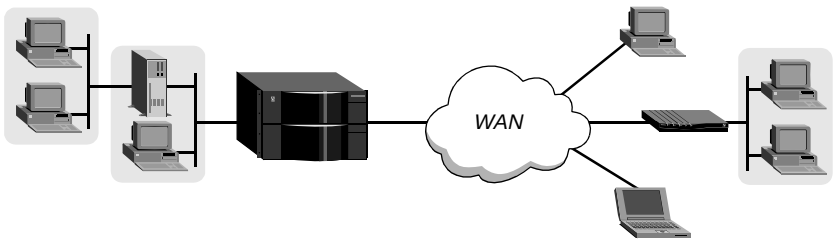

Figure 5. Routing AppleTalk between LAN and WAN interfaces

Note: AppleTalk routing be enabled on the shelf-controller to enable the system to forward AppleTalk packets from a host card to the shelf-controller. This is required for any kind of AppleTalk connection, even if the individual Connection profile to a remote device does not use routing.

Atalk-Global profile settings

When an ARA or AppleTalk PPP client dials in, the MAX TNT assigns the client an AppleTalk address on a virtual AppleTalk network. You define the virtual AppleTalk network in the Atalk-Global profile by setting the following parameters, which are shown with sample settings:

```
ATALK-GLOBAL
   atalk-dialin-pool-start = 100
   atalk-dialin-pool-end = 200
```

AppleTalk networks are assigned a network range, which is a contiguous range of integers between 1 and 65,199. Each network range must be unique, No two networks can use the same range, and no two network ranges can overlap.

Each number in the range can be associated with up to 253 nodes, so the range determines how many clients can dial in. For example, a network with a range 1000-1002 could support up to 2 x 253, or 506 clients. Following is an example of defining a virtual network. In this case, the network range is 1000–1002:

```
admin> read atalk-global
ATALK-GLOBAL read
admin> set atalk-dialin-pool-start = 1000
admin> set atalk-dialin-pool-end = 1002
admin> write
ATALK-GLOBAL written
```

Atalk-Interface profile settings

In the Atalk-Interface profile, you enable AppleTalk routing and specify whether the MAX TNT will operate as a seed or nonseed router on the interface. In this release, only the built-in Ethernet interface on the shelf-controller can be configured an AppleTalk interface. The Atalk-Interface profile contains the following parameters, which are shown with sample settings:

```
ATALK-INTERFACE { { shelf-1 controller 1 } 0 }
   interface-address* = { { shelf-1 controller 1 } 0 }
   atalk-routing-enabled = yes
   hint-zone = "SLC Engineering"
   atalk-Router = atlk-router-seed
   atalk-Net-Start = 1001
   atalk-Net-End = 1010
   atalk-Default-Zone = "SLC Engineering"
   atalk-Zone-List = [ "SLC Engineering" "SLC Test 1" "SLC Test
```

Configuring a seed router

A seed router has its own hard-coded network and zone configuration. You configure the MAX TNT as a seed AppleTalk router on a LAN interface by using the following parameters:

Parameter	Effect
Atalk-Routing-Enabled	Enables the MAX TNT to route AppleTalk on the shelf-controller Ethernet interface. If set to No, none of the other AppleTalk parameters applies.

Parameter	Effect
Atalk-Router	Specifies a routing mode. If set to Atlk-Router-Off, none of the remaining parameters applies. If set to Atlk-Router-Seed, the unit comes up with the specified zone and network configuration. If there are other AppleTalk routers on the LAN interface, you must be sure that the zone and network information you specify is completely consistent with the corresponding specifications for those routers.
Atalk-Net-Start Atalk-Net-End	Specify the network range for the shelf-controller Ethernet interface. The network range is a contiguous range of integers between 1 and 65,199. Each range must be unique. No two interfaces may use the same range, and no two network ranges may overlap. Each number in the range can be associated with up to 253 nodes, so the range determines how many clients the interface can support. For example, an interface with the range 1005-1010 could support up to 5 x 253, or 1265 clients.
Default-Zone	Specifies the default AppleTalk zone for the shelf-controller Ethernet interface. The default zone is the zone assigned to an AppleTalk service on this interface if the service does not select a zone in which to reside.
Zone-List	Specifies the zone list for the shelf-controller Ethernet interface. The zone list is a list of 1 to 32 AppleTalk zone names. Each name consists of from 1 to 33 characters, including embedded spaces. The characters must be in the standard printing character set, and must not include an asterisk (*).

In the following example, the MAX TNT is configured as a seed router on the shelf-controller Ethernet interface. The sample configuration defines the network range 1005–1010, three zones, and the default zone for the LAN interface.

```
admin> read atalk-int {{1 c 1} 0}
ATALK-INTERFACE/ { { shelf-1 controller 1 } 0 } read

admin> set atalk-routing = yes

admin> set atalk-router = atlk-router-seed

admin> set atalk-net-start = 1005

admin> set atalk-net-end = 1010

admin> set atalk-default-zone = engineering

admin> set atalk-zone-list 1 = admin

admin> set atalk-zone-list 2 = test

admin> set atalk-zone-list 2 = engineering

admin> write
ATALK-INTERFACE/ { { shelf-1 controller 1 } 0 } written
```

Configuring a nonseed router

A nonseed router acquires its network and zone configuration from another router on the network. You configure the MAX TNT as a nonseed AppleTalk router on a LAN interface by using the following parameters:

Parameter	Effect
Atalk-Routing-Enabled	Enables the MAX TNT to route AppleTalk on the shelf-controller Ethernet interface. If set to No, none of the other AppleTalk parameters applies.
Atalk-Router	Specifies a routing mode. If set to Atlk-Router-Off, none of the remaining parameters applies. If set to Atlk-Router-Nonseed, the unit learns its zone and network configuration from another AppleTalk router (a seed router) on the network. If the MAX TNT is configured in nonseed mode, a seed router must be available at startup time, or the MAX TNT cannot come up in AppleTalk routing mode. (If the MAX TNT comes up without AppleTalk routing enabled because no seed routers were available at startup, you must reset the system after a seed router is up.)
Hint-Zone	Specifies the zone in which the MAX TNT resides. The MAX TNT can include the specified zone name in the ZipGetNetInfo request packet it sends out to get its configuration from a seed router, and the router can return a valid network range for that zone.

In the following example, the MAX TNT is configured as a nonseed router:

```
admin> read atalk-int {{1 c 1} 0}
ATALK-INTERFACE/ { { shelf-1 controller 1 } 0 } read

admin> set atalk-routing = yes

admin> set atalk-router = atlk-router-non-seed

admin> set hint-zone = engineering

admin> write
ATALK-INTERFACE/ { { shelf-1 controller 1 } 0 } written
```

Answer-Defaults profile settings

To enable ARA client connections, you must enable ARA-Answer in the Answer-Defaults profile. In addition, if you intend to allow ARA Guest access set the Profiles-Required parameter to No (it is typically set to Yes for security purposes). These are the relevant parameters:

```
ANSWER-DEFAULTS
   profiles-required = no
   ara-answer
      enabled = yes
```

Following is an example of input that enables ARA-Answer and disables ARA Guest access:

```
admin> read answer
ANSWER-DEFAULTS read

admin> set ara-answer enabled = yes
```

```
admin> set profiles-required = yes

admin> write
ANSWER-DEFAULTS written
```

(Setting Profiles-Required to Yes disables ARA Guest access.)

Connection profile settings

PPP and ARA are the encapsulation protocols used for AppleTalk dialin on the MAX TNT. AppleTalk PPP and ARA Client software are available from Apple Computer (both ARA and PPP are supported in ARA 3.0) and from other vendors such as Netmanage Pacer PPP. Both AppleTalk PPP and ARA can be used over a modem or V.120 ISDN TA connection. AppleTalk PPP can also be used over sync-PPP when the calling unit is an Ascend router (Pipeline or MAX series).

You configure ARA or AppleTalk PPP connections by using the following parameters, which are shown with sample settings:

```
CONNECTION station
   encapsulation-protocol = ara
   ara-options
      recv-password = test
      ara-enabled = yes
      maximum-connect-time = 0
   appletalk-options
      atalk-routing-enabled = no
      atalk-static-ZoneName = ""
      atalk-static-NetStart = 0
      atalk-static-NetEnd = 0
      atalk-Peer-Mode = router-peer
```

Note: AppleTalk routing must be enabled for incoming PPP connections, but it is not necessary for ARA client connections.

Using these parameters, there are three ways to configure a Connection profile for AppleTalk connectivity:

- ARA client connection
- PPP dialin connection
- Synchronous PPP connection with an Ascend router

Configuring an ARA client Connection profile

An ARA client connection uses the ARA encapsulation protocol, and does not require AppleTalk routing in the Connection profile. You configure an ARA client connection by using the following parameters:

Parameter	Effect
Encapsulation-Protocol	Must specify ARA for ARA client connections.
ARA-Enabled	In the ARA-Options subprofile, the ARA-Enabled parameter turns on ARA processing for the connection.
Recv-Password	Specifies the password sent to the MAX TNT by the ARA client.

Parameter	Effect
Maximum-Connect-Time	Specifies the maximum number of minutes an ARA session can remain connected. The default setting, 0 (zero) disables the timer. If you specify a maximum connect time, the MAX TNT initiates an ARA disconnect when that time is up. The ARA link goes down cleanly, but remote users are not notified. Users will find out the ARA link is gone only when they try to access a device.

In Figure 6, the dial-in client is running ARA 3.0, with ARA encapsulation selected and with an internal modem. In this example, the client will be assigned a network address on the virtual 1000–1002 network, and a maximum ARA connection time of 60 minutes.

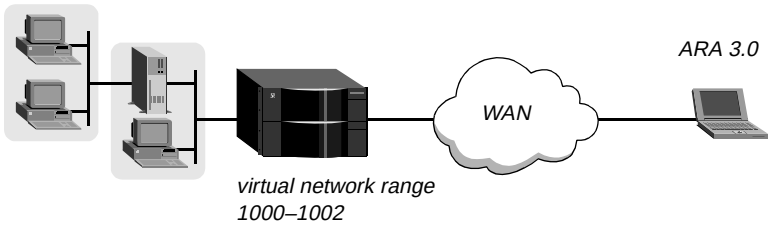

Figure 6. ARA Client dial-in

The administrator enters the following commands at the system prompt:

```
admin> read connection araclient
CONNECTION/araclient read
admin> set active = yes
admin> set encaps = ara
admin> set ara-enabled = yes
admin> set ara recv-password = ara-password
admin> set maximum-connect-time = 60
admin> write
CONNECTION/araclient written
```

Configuring a PPP dialin client connection for AppleTalk

An AppleTalk PPP dialin client connection uses the PPP encapsulation protocol. You configure an AppleTalk PPP client connection by using the following parameters:

Parameter	Effect
Encapsulation-Protocol	Must specify PPP for an AppleTalk PPP dialin client connection.
Recv-Password	In the PPP-Options subprofile, specifies the password sent to the MAX TNT by the PPP client.
Atalk-Routing-Enabled	In the AppleTalk-Options subprofile, enables AppleTalk routing for the connection. If AppleTalk routing has not been enabled in the Atalk-Interface profile, or if the Answer-Defaults profile does not enable ARA-Answer, this parameter has no effect.

Parameter	Effect
Atalk-Peer-Mode	In the Appletalk-Options subprofile, specifies whether the remote site is a dial-in client or another AppleTalk router. When set to Dialin-Peer, the MAX TNT negotiates a routing session with the dial-in client by assigning the client a node address on the virtual AppleTalk network defined in the Atalk-Global profile. The client must accept the network number assigned.

In Figure 7, the dial-in client is running ARA 3.0, and has selected PPP encapsulation, or is using another PPP dialer that supports AppleTalk. The client will be assigned a network address on the virtual 1000-1002 network.

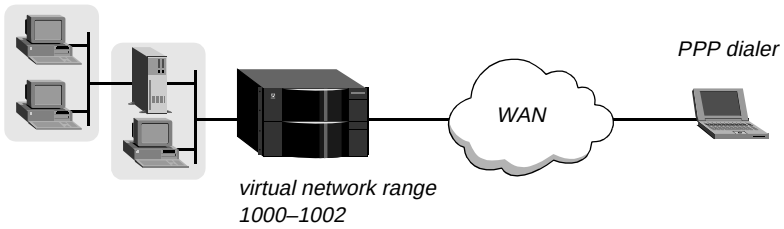

Figure 7. AppleTalk connection using a PPP dialer

The administrator might configure a Connection profile for the client as follows:

```
admin> new connection ppp-atalk
CONNECTION/ppp-atalk read
admin> set active = yes
admin> set encapsulation-protocol = ppp
admin> set ppp recv-password = localpw
admin> set appletalk atalk-routing-enabled = yes
admin> set appletalk atalk-peer-mode = dialin
admin> write
CONNECTION/ppp-atalk written
```

For details about other PPP settings, see the *MAX TNT Network Configuration Guide*.

Configuring an AppleTalk routing connection

An AppleTalk routing connection uses the PPP encapsulation protocol or one of its multilink variants (MP or MP+). You configure an AppleTalk routing connection by using the following parameters:

Parameter	Effect
Encapsulation-Protocol	Specifies PPP, MP, or MPP for an AppleTalk routing connection.
Recv-Password	In the PPP-Options subprofile, specifies the password sent to the MAX TNT by the PPP client.
Atalk-Routing-Enabled	In the AppleTalk-Options subprofile, enables AppleTalk routing for the connection. If AppleTalk routing has not been enabled in the Atalk-Interface profile, or if the Answer-Defaults profile does not enable ARA-Answer, this parameter has no effect.
Atalk-Peer-Mode	In the Appletalk-Options subprofile, specifies whether the remote site is a dial-in client or another AppleTalk router. When set to Router-Peer (the default), the MAX TNT acquires the remote site's network information during session negotiation.

Note: In this release, the Atalk-Static parameters have no useful purpose, because they configure a dialout static route to a remote AppleTalk router, and only dial-in is currently supported. In a future release, Atalk-Static-ZoneName will specify a zone name to be used with the static route to a remote site, and the Atalk-Static-NetStart and Atalk-Static-NetEnd parameters will define the network range for packets that are to be routed to the remote site. These settings will follow the rules described in "Atalk-Interface profile settings" on page 53.

In Figure 8, the remote Pipeline unit is configured as an AppleTalk router on the extended AppleTalk network 2000-2001, in the Branch zone.

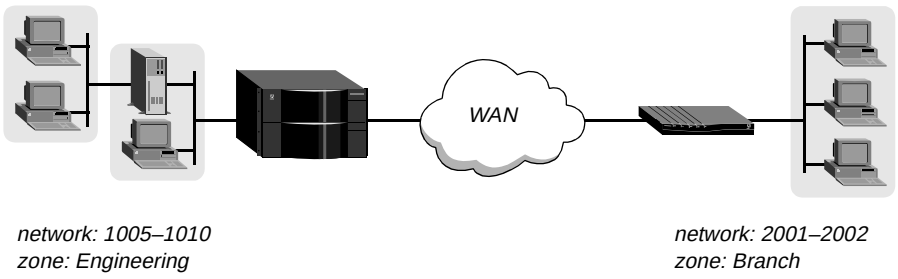

Figure 8. AppleTalk routing connection

Following is an example Connection profile for the remote Pipeline:

```
admin> read connection atalk-router
CONNECTION/atalk-router read

admin> set active = yes

admin> set encaps = ppp

admin> set ppp recv-password = rtr-password

admin> set appletalk atalk-routing enabled = yes

admin> set appletalk atalk-peer-mode = router-peer
```

```
admin> write
CONNECTION/atalk-router written
```

Supporting IP over AppleTalk

To route IP and AppleTalk, the MAX TNT must be configured both as an IP router as well as an AppleTalk router. For details about configuring the IP router and individual IP connections, see the *MAX TNT Network Configuration Guide*.

To support IP, the Connection profile for a dial-in client must specify an IP configuration, and the client must configure Macintosh TCP/IP software (such as Open Transport). Table 2 describes Macintosh TCP/IP configurations for a PPP connection:

Table 2. Macintosh TCP/IP settings for PPP connections

Macintosh software	IP settings for a PPP AppleTalk connection
Open Transport	The TCP/IP Control Panel must specify a PPP connection and the client IP address. If the Connection profile has a hard-coded IP address, type that address manually in the Control Panel. If the Connection profile specifies dynamic address assignment, set the Control Panel to obtain an address from the PPP server.
MacTCP	The MacTCP Control Panel should select the PPP icon, and the client IP address. If the Connection profile has a hard-coded IP address, type that address manually in the Control Panel. If the Connection profile specifies dynamic assignment of an address, set the Control Panel to obtain an address from a Server. (The Dynamic option in MacTCP is not supported.)

The same requirements apply to apply to an ARA connection. When ARA encapsulation is in use, the MAX TNT handles IP packets by encapsulating and decapsulating the packets in DDP. Table 3 describes Macintosh TCP/IP configurations for a PPP connection:

Table 3. Macintosh TCP/IP settings for ARA connections

Macintosh software	IP settings for an ARA connection
Open Transport	The TCP/IP Control Panel must specify a connection via Mac-IP, and the client IP address. If the Connection profile has a hard-coded IP address, type that address manually in the Control Panel. If the Connection profile specifies dynamic assignment of an address, set the Control Panel to obtain an address from the Mac-IP server.
MacTCP	The MacTCP Control Panel should select the ARA icon, and the client IP address. If the Connection profile has a hard-coded IP address, type that address manually in the Control Panel. If the Connection profile specifies dynamic assignment of an address, set the Control Panel to obtain an address from a Server. (The Dynamic option in MacTCP is not supported.)

In Figure 9, the dial-in client is running ARA 3.0 (which includes DDP-IP tunneling capabilities) and an IP application such as Telnet to communicate with an IP host on the MAX TNT local interface. The client has a hard-coded IP address.

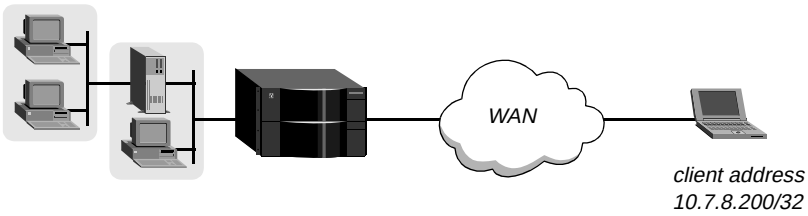

Figure 9. ARA connection that encapsulates IP packets in DDP

The following sample configuration enables the client to dial in using ARA client 3.0 and then initiate a Telnet connection to a host on the MAX TNT unit's IP network:

```
admin> read connection ddpip-client
CONNECTION/ddpip-client read

admin> set active = yes

admin> set encaps = ara

admin> set ara ara-enabled = yes

admin> set ara recv-password = ara-password

admin> set ip-options remote = 10.7.8.200/32

admin> write
CONNECTION/ddpip-client written
```

Frame Relay Switching

In this release, the MAX TNT supports a switched mode of operation between Frame-Relay interfaces (as opposed to routed or bridged operation). When a Frame Relay circuit has been specified in two Connection profiles, the MAX TNT switches data received on one of the paired interfaces to the other paired interface. Circuit switching occurs at layer 2 based on the assigned DLCIs. The layer 3 IP router never receives the packets.

A circuit is a Permanent Virtual Circuit (PVC) segment that consists of two DLCI endpoints, and typically uses two different Frame-Relay profiles. In this release, the Frame-Relay profiles can specify Network-to-Network Interface (NNI) operations. NNI is a mode of operation in which the system acts as a Frame Relay switch communicating with another Frame Relay switch.

Note: NNI operations are not required for the MAX TNT to operate in switched mode using Frame Relay circuits. However, to make an NNI interface useful for passing data, it must be paired with another interface to complete a switch circuit.

New parameters for Frame Relay circuits

Following are the parameters related to Frame Relay circuits, which are shown with sample settings:

```
CONNECTION station-1
   encapsulation-protocol = frame-relay-circuit
```

```
      fr-options
         frame-relay-profile = max
         dlci = 55
         circuit-name = frcir1

   CONNECTION station-2
      encapsulation-protocol = frame-relay-circuit
      fr-options
         frame-relay-profile = p130-east
         dlci = 77
         circuit-name = frcir1
```

Parameter	Effect
Encapsulation-Protocol	Both endpoints of the circuit must specify Frame-Relay-Circuit encapsulation.
Frame-Relay-Profile	Specifies the name of a Frame-Relay profile.
DLCI	Specifies a DLCI number for this endpoint of the circuit.
Circuit-Name	Specifies a name for the circuit (up to 16 characters). The other endpoint must specify the same circuit name.

Note: IP routing is not enabled in the circuit profiles. Circuit switching occurs at layer 2 based on the assigned DLCIs. The layer 3 IP router never receives the packets.

New parameter setting for NNI

The following parameter specifies NNI operations on a data link to another FR switch. The parameter is shown with a sample setting:

```
FRAME-RELAY fr-name
   link-type = nni
```

Parameter	Effect
Link-Type	Specifies the kind of logical interface between the MAX TNT and the Frame Relay network on the data link. Setting the Link-Type to NNI puts the link into NNI operating mode. The MAX TNT performs bidirectional LMI on the link.

Following are important differences in NNI as opposed to UNI operation:

- Received frames are switched to another DLCI circuit segment, rather than being routed (layer 3) or bridged.
- Link management is bidirectional on an NNI interface.
- Asynchronous update reports are sent when DLCI state changes occur.

Configuring the physical link for a Frame Relay interface

Frame Relay switching requires two Frame Relay interfaces (Frame-Relay profiles) and two Connection profiles to define the circuit.

Each Frame-Relay profile requires its own nailed physical link. For example, you could set all 24 channels on a T1 to the same nailed group number to dedicate a T1 line to Frame-Relay, or you could configure the link on a FrameLine card, SWAN card, or any other card that supports sufficient nailed bandwidth.

Following is an example that shows how to define one physical link by configuring all channels of a T1 line for nailed usage and assigning them to the Nailed-Group number 11:

```
admin> read t1 { 1 13 6 }
T1/{ shelf-1 slot-13 6 } read
admin> set line channel 1 channel-usage = nailed-64-channel
admin> set line channel 1 nailed-group = 11
admin> set line channel 2 channel-usage = nailed-64-channel
admin> set line channel 2 nailed-group = 11
admin> set line channel 3 channel-usage = nailed-64-channel
admin> set line channel 3 nailed-group = 11
admin> set line channel 4 channel-usage = nailed-64-channel
admin> set line channel 4 nailed-group = 11
admin> set line channel 5 channel-usage = nailed-64-channel
admin> set line channel 5 nailed-group = 11
admin> set line channel 6 channel-usage = nailed-64-channel
admin> set line channel 6 nailed-group = 11
admin> set line channel 7 channel-usage = nailed-64-channel
admin> set line channel 7 nailed-group = 11
admin> set line channel 8 channel-usage = nailed-64-channel
admin> set line channel 8 nailed-group = 11
admin> set line channel 9 channel-usage = nailed-64-channel
admin> set line channel 9 nailed-group = 11
admin> set line channel 10 channel-usage = nailed-64-channel
admin> set line channel 10 nailed-group = 11
admin> set line channel 11 channel-usage = nailed-64-channel
admin> set line channel 11 nailed-group = 11
admin> set line channel 12 channel-usage = nailed-64-channel
admin> set line channel 12 nailed-group = 11
admin> set line channel 13 channel-usage = nailed-64-channel
admin> set line channel 13 nailed-group = 11
admin> set line channel 14 channel-usage = nailed-64-channel
admin> set line channel 14 nailed-group = 11
admin> set line channel 15 channel-usage = nailed-64-channel
admin> set line channel 15 nailed-group = 11
admin> set line channel 16 channel-usage = nailed-64-channel
admin> set line channel 16 nailed-group = 11
admin> set line channel 17 channel-usage = nailed-64-channel
```

```
admin> set line channel 17 nailed-group = 11
admin> set line channel 18 channel-usage = nailed-64-channel
admin> set line channel 18 nailed-group = 11
admin> set line channel 19 channel-usage = nailed-64-channel
admin> set line channel 19 nailed-group = 11
admin> set line channel 20 channel-usage = nailed-64-channel
admin> set line channel 20 nailed-group = 11
admin> set line channel 21 channel-usage = nailed-64-channel
admin> set line channel 21 nailed-group = 11
admin> set line channel 22 channel-usage = nailed-64-channel
admin> set line channel 22 nailed-group = 11
admin> set line channel 23 channel-usage = nailed-64-channel
admin> set line channel 23 nailed-group = 11
admin> set line channel 24 channel-usage = nailed-64-channel
admin> set line channel 24 nailed-group = 11
admin> write
T1/{ shelf-1 slot-13 6 } written
```

For details about configuring lines and channels, see the *MAX TNT Hardware Installation Guide*.

Configuring a circuit between UNI interfaces

Figure 10 shows a circuit configuration in the MAX TNT. In this example, assume that the two T1 lines in the MAX TNT are configured for nailed usage. The channels of the T1 line connecting to the Pipeline 130 labeled *P130-East* are in Nailed-Group 33, and the channels of the T1 line connecting to the MAX are in Nailed-Group 11 (see “Configuring the physical link for a Frame Relay interface” on page 62).

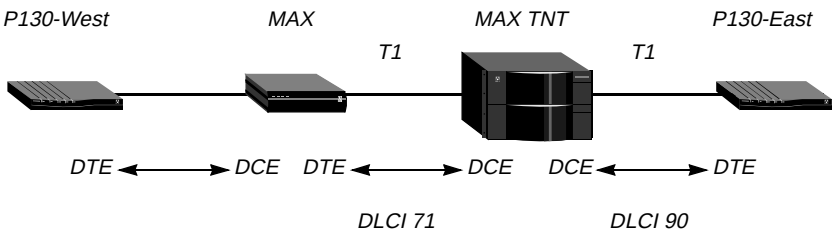

Figure 10. Frame Relay circuit with UNI interfaces

The following commands define a Frame-Relay profile to the MAX in Figure 10:

```
admin> new frame max
FRAME-RELAY/max read
admin> set active = yes
admin> set nailed-up-group = 11
admin> set link-type = dce
```

```
admin> write
FRAME-RELAY/max written
```

The following commands define a Frame-Relay profile to *P130-East* in Figure 10:

```
admin> new frame p130east
FRAME-RELAY/p130east read
admin> set active = yes
admin> set nailed-up-group = 33
admin> set link-type = dce
admin> write
FRAME-RELAY/p130east written
```

The following commands configure a Connection profile to send or receive data to or from the MAX on DLCI 71:

```
admin> read conn max-frcir
CONNECTION/max-frcir read
admin> set active = yes
admin> set encaps = frame-relay-circuit
admin> set ip-options ip-routing-enabled = no
admin> set fr-options frame-relay-profile = max
admin> set fr-options dlci = 71
admin> set fr-options circuit-name = frcir1
```

The following commands configure a Connection profile to send or receive data to or from *P130-East* on DLCI 90:

```
admin> read conn p130-frcir
CONNECTION/p130-frcir read
admin> set active = yes
admin> set encaps = frame-relay-circuit
admin> set ip-options ip-routing-enabled = no
admin> set fr-options frame-relay-profile = p130east
admin> set fr-options dlci = 90
admin> set fr-options circuit-name = frcir1
```

Configuring a circuit between NNI interfaces

Figure 11 shows a circuit configuration that uses NNI interfaces. In this example, assume that the lines connecting the MAX TNT to each of the switches have been configured for nailed usage. For this example, the line connecting to the switch labeled *FR-Asnd-A* is in Nailed-Group 52, and the line connecting to the switch labeled *FR-Asnd-B* is in Nailed-Group 128.

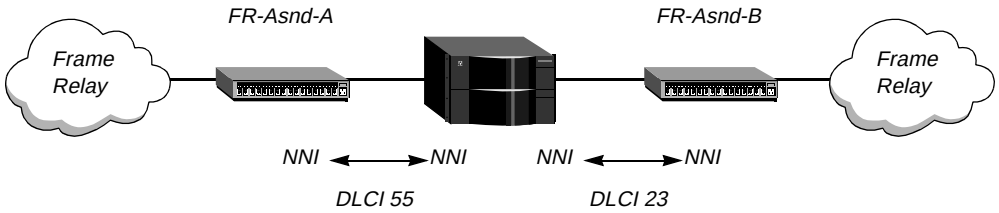

Figure 11. Frame-Relay profile defines NNI operation

The following commands define a Frame-Relay profile to *FR-Asnd-A* in Figure 11:

```
admin> new frame fr-asnd-a
FRAME-RELAY/fr-asnd-a read
admin> set active = yes
admin> set nailed-up-group = 52
admin> set link-type = nni
admin> set link-mgmt = ansi-t1.617d
admin> set n391-val = 6
admin> set t391-val = 10
admin> set t392-val = 15
admin> write
FRAME-RELAY/fr-asnd-a written
```

The following commands define a Frame-Relay profile to *FR-Asnd-B* in Figure 11:

```
admin> new frame fr-asnd-b
FRAME-RELAY/fr-asnd-b read
admin> set active = yes
admin> set nailed-up-group = 128
admin> set link-type = nni
admin> set link-mgmt = ansi-t1.617d
admin> set n391-val = 6
admin> set t391-val = 10
admin> set t392-val = 15
admin> write
FRAME-RELAY/fr-asnd-b written
```

The following commands configure a Connection profile to send or receive data to or from the switch named *FR-Asnd-A* on DLCI 55:

```
admin> new conn asnd-a
CONNECTION/asnd-a read
admin> set active = yes
admin> set encaps = frame-relay-circuit
admin> set ip-options ip-routing-enabled = no
admin> set fr-options frame-relay-profile = fr-asnd-a
admin> set fr-options dlci = 55
```

```
admin> set fr-options circuit-name = pvc-pipe
admin> write
CONNECTION/asnd-a written
```

The following commands configure a Connection profile to send or receive data to or from the switch named *FR-Asnd-B* on DLCI 23:

```
admin> new conn asnd-b
CONNECTION/asnd-b read
admin> set active = yes
admin> set encaps = frame-relay-circuit
admin> set ip-options ip-routing-enabled = no
admin> set fr-options frame-relay-profile = fr-asnd-b
admin> set fr-options dlci = 23
admin> set fr-options circuit-name = pvc-pipe
admin> write
CONNECTION/asnd-b written
```

Ascend Tunnel Management Protocol (ATMP)

ATMP is a UDP/IP based protocol that provides a tunneling mechanism between two Ascend units across an IP network. The data is transported in Generic Routing Encapsulation (GRE) as described in RFC 1701. For a complete description of ATMP, see RFC 2107, K. Hamzeh, "Ascend Tunnel Management Protocol - ATMP."

ATMP tunnels

Figure 12 shows an ATMP tunnel between two MAX TNT units. The unit that authenticates the mobile client is the ATMP *foreign agent*. The unit that accesses the home network is the ATMP *home agent.* The *home network* is the destination network for mobile clients. For example, in Figure 12, the mobile client might be a sales person who logs into an ISP (the foreign agent) to access his or her home network.

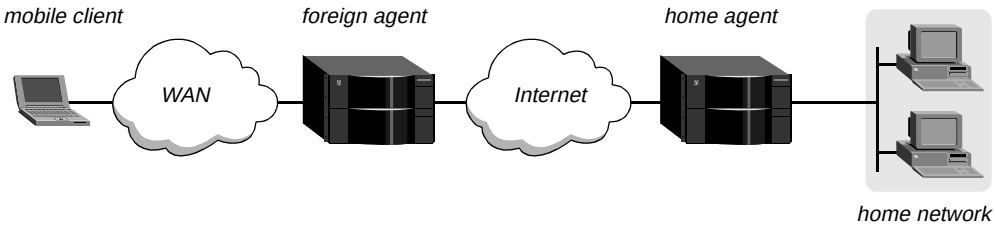

Figure 12. ATMP tunnel across the Internet

A mobile client dials into the foreign agent, which authenticates the client by means of a Connection profile or RADIUS. The foreign agent then establishes an IP connection to the home agent, and requests an ATMP tunnel on top of the established IP connection.

The home agent is the terminating part of the tunnel, where most of the ATMP intelligence resides. It must be able to communicate with the home network through a direct connection, another router, or across a nailed connection.

Setting the system address

If the home agent or foreign agent has multiple interfaces into the IP cloud that separates the two units, it is very important that you set a system IP address. Otherwise, you might encounter communication problems if a route changes within the IP cloud. Following is the relevant parameter, shown with a sample setting:

```
IP-GLOBAL
   system-ip-addr = 10.2.3.4
```

When configuring mobile clients, the IP address of the home agent must be the IP address of the unit (the system address), not the IP address of the interface on which the home agent receives tunneled data.

For more information about the System-IP-Addr parameter, see the *MAX TNT Network Configuration Guide* or the *MAX TNT Reference Guide*.

ATMP profile settings

In the ATMP profile, you specify whether the MAX TNT operates as a home or foreign agent. Create the ATMP profile as follows:

```
admin> new atmp
ATMP read

admin> write
ATMP written
```

The ATMP profile contains the following parameters, which are shown with their default values:

```
ATMP
   agent-mode = tunnel-disabled
   agent-type = gateway-home-agent
   udp-port = 5150
   home-agent-password = ""
   retry-timeout = 3
   retry-limit = 10
```

Table 4. ATMP profile parameters

ATMP parameter	Home agent	Foreign agent
Agent-Mode	Required	Required
Agent-Type	Required	N/A
UDP-Port	Optional	Optional
Home-Agent-Password	Recommended	N/A
Retry-Timeout	Optional	Optional
Retry-Limit	Optional	Optional

Specifying the agent mode

The Agent-Mode parameter specifies whether the MAX TNT operates as a foreign agent, a home agent, or as both on a tunnel-by-tunnel basis. The default tunnel-disabled mode disables ATMP.

Setting the agent type

When Agent-Mode is set to home-agent, the Agent-Type parameter specifies whether the MAX TNT reaches the home network as a gateway or a router. The default is gateway.

When it is set to gateway-home-agent, the home agent delivers tunneled data to the home network without routing. The tunneled data does not bring up a connection to the home network, so the connection between the home agent and home network must already be up, as for example in a nailed or direct connection.

When the Agent-Type parameter is set to router-home-agent, the home agent routes tunneled data to the home network.

Specifying a UDP port for the tunnel

The UDP-Port parameter sets the UDP port the unit uses locally to manage the tunnel. The default value is 5150. Both ends of a tunnel must agree on its value.

Setting the home agent password

When the MAX TNT operates as a home agent, the Home-Agent-Password parameter sets the password a foreign agent must supply to establish a tunnel with this box. You can specify up to 21 characters.

Specifying retry limits

The Retry-Timeout parameter controls the time to wait between retries when attempting to establish a tunnel. The default value is 3 seconds, which is appropriate for most sites.

The Retry-Timeout parameter controls the maximum number of attempts to establish a tunnel before switching to an alternate home agent. You can specify a number between 1 and 100. The default is 10.

Example of setting up a foreign agent

Following is an example of an ATMP profile for a foreign agent:

```
admin> new atmp
ATMP read

admin> set agent-mode = foreign-agent

admin> write
ATMP written
```

For a more detailed example, see “Example of a foreign agent configuration” on page 73.

Example of setting up a home agent

Following is an example of an ATMP profile for a home agent in gateway mode:

```
admin> new atmp
ATMP read
admin> set agent-mode = home-agent
admin> set agent-type = gateway-home-agent
admin> set home-agent-password = my-password
admin> write
ATMP written
```

For a more detailed example, see “Example of a home agent configuration in gateway mode” on page 74.

Connection profile parameters

When the MAX TNT is acting as a home agent in gateway mode, a Connection profile defines the connection to the home network. When it is operating as a foreign agent, Connection profiles authenticate mobile client connections that will be tunneled to a home agent.

Following are the parameters for configuring either of these ATMP-related connections, shown with their default values:

```
CONNECTION station
   tunnel-options
      profile-type = disabled
      max-tunnels = 0
      primary-home-agent = ""
      secondary-home-agent = ""
      udp-port = 5150
      home-agent-password = ""
      home-network-name = ""
```

Table 5. Connection profile tunnel-option parameters

Tunnel parameter	Mobile client	Gateway to home network
Profile-Type	Required	Required
Max-Tunnels	N/A	Optional
Primary-Home-Agent	Required	N/A
Secondary-Home-Agent	Optional	N/A
UDP-Port	Optional	N/A
Home-Agent-Password	Required if specified in home agent	N/A
Home-Network-Name	Required if home agent is in gateway mode	N/A

Specifying a profile type

The Profile-Type parameter specifies the type of connection. Three options are available:

- Disabled

  The connection is not used for ATMP.

- Mobile-Client

  Specifying this setting if the Connection profile is used to authenticate a mobile client.

- Gateway-Profile.

  Specifying this setting if the Connection profile defines a gateway-mode connection to a home network.

Although you cannot set the profile type in RADIUS, it is effectively set to Mobile-Client if the RADIUS profile specifies a home agent address attribute (Ascend-Home-Agent-IP-Addr, Ascend-Primary-Home-Agent or Ascend-Secondary-Home-Agent).

Specifying a tunnel maximum

When the Connection profile defines a gateway-mode connection to a home network, the Max-Tunnels parameter controls the maximum number of mobile clients that can use the connection, all at the same time, to tunnel into that home network. A value of 0 sets no limit. The default value is 0.

Specifying a primary and secondary home agent

For a Mobile-Client connection, the Primary-Home-Agent parameter specifies the IP address or host name of the primary home agent, and the Secondary-Home-Agent parameter specifies the IP address or host name of a secondary home agent.

If you specify a host name (up to 31 characters), the MAX TNT attempts to look up the host IP address in DNS. If the home agent requires a UDP port number different than the value specified in the UDP-Port parameter, you can specify a port value by appending a colon character (:) and the port number to the host name. For example:

```
admin> set primary-home-agent=10.11.22.33:8877
admin> set primary-home-agent=home-agent.company.com:6969
admin> set secondary-home-agent=11.56.12.128:4000
```

The home agent IP address should be the system address, not the IP address of the interface on which it receives tunneled data. (For more detail, see “Setting the system address” on page 68.)

You can specify this information in RADIUS by means of the Ascend-Primary-Home-Agent, Ascend-Secondary-Home-Agent, or Ascend-Home-Agent-IP-Addr attribute.

Specifying a UDP port for the tunnel

For a Mobile-Client connection, the UDP-Port parameter sets the default UDP port to use when communicating with a home agent. You can override this value by specifying the port in the home agent address string, as described in the preceding section.The default value is 5150. Both ends of a tunnel must agree on the value.

You can specify this information in RADIUS by means of the Ascend-Home-Agent-UDP-Port attribute.

Specifying the home agent password

For a Mobile-Client connection, the Home-Agent-Password parameter specifies the password required by the home agent (up to 20 characters). For related information, see “Setting the home agent password” on page 69.

You can specify this information in RADIUS by means of the Ascend-Home-Agent-Password attribute.

Specifying the home network name

For a Mobile-Client connection when the home agent is running in gateway mode, the Home-Network-Name parameter refers to the name of the home network connection. (If the home agent is operating in router mode, leave this parameter blank.) The name of the home network connection is specified in the station parameter of that Connection profile on the home agent.

You can specify this information in RADIUS by means of the Ascend-Home-Network-Name attribute.

Example of a mobile client Connection profile

Following is an example of a procedure that configures the ATMP aspects of a Connection profile for a mobile client:

```
admin> read connection rachel
CONNECTION/rachel read

admin> list tunnel-options
profile-type = disabled
max-tunnels = 0
primary-home-agent = ""
secondary-home-agent = ""
udp-port = 5150
home-agent-password = ""
home-network-name = ""

admin> set profile-type = mobile-client

admin> set primary-home-agent = jupiter.xyz.com

admin> set home-agent-password = jupiter-password

admin> write
CONNECTION/rachel written
```

Example of a home agent gateway mode Connection profile

Following is an example of a procedure that configures the ATMP aspects of a Connection profile for a home agent in gateway mode, supporting a maximum of 120 tunnels to the home network at xyz.com:

```
admin> read connection xyz
CONNECTION/xyz read
```

```
admin> list tunnel-options
profile-type = disabled
max-tunnels = 0
primary-home-agent = ""
secondary-home-agent = ""
udp-port = 5150
home-agent-password = ""
home-network-name = ""

admin> set profile-type = gateway-profile

admin> set max-tunnels = 120

admin> write
CONNECTION/xyz written
```

Example ATMP configurations

This section contains several examples that show how to set up foreign agent, mobile client, and home agent profiles.

Example of a foreign agent configuration

The foreign agent is responsible for authenticating mobile clients and requesting a tunnel across an IP connection to a home agent. Following is an example of a procedure that configures the MAX TNT to operate as a foreign agent, and configures two Connection profiles for mobile clients—one to a home agent in gateway mode, and one to a home agent in router mode.

1 Open the ATMP profile (create it if necessary):

```
admin> new atmp
ATMP read
```

2 Specify foreign-agent mode, and then write the profile:

```
admin> set agent-mode = foreign-agent

admin> write
ATMP written
```

3 Create an IP-routing Connection profile to the home agent.

For information about configuring IP routing connections, see the *MAX TNT Network Configuration Guide*. For information about configuring connections in RADIUS, see the *MAX TNT RADIUS Configuration Guide.*

4 Create a Connection profile for each mobile client that will tunnel to the home agent. The following example shows the ATMP configuration of a Connection profile to a home agent in router mode:

```
admin> read connection ca1
CONNECTION/ca1 read

admin> list tunnel-options
profile-type = disabled
max-tunnels = 0
primary-home-agent = ""
secondary-home-agent = ""
udp-port = 5150
home-agent-password = ""
home-network-name = ""
```

```
admin> set profile-type = mobile-client
admin> set primary-home-agent = home-agent.abc.com
admin> set secondary-home-agent = 11.56.12.128
admin> set home-agent-password = abc-password
admin> write
CONNECTION/ca1 written
```

The following example shows the ATMP configuration of a Connection profile for connecting to a home agent in gateway mode:

```
admin> read connection pac
CONNECTION/pac read
admin> list tunnel-options
profile-type = disabled
max-tunnels = 0
primary-home-agent = ""
secondary-home-agent = ""
udp-port = 5150
home-agent-password = ""
home-network-name = ""
admin> set profile-type = mobile-client
admin> set primary-home-agent = home-agent.xyz.com
admin> set secondary-home-agent = 12.66.56.120
admin> set home-agent-password = abc-password
admin> set home-network-name = homenet
admin> write
CONNECTION/pac written
```

In the last set of commands, *homenet* is the name of the home agent's Connection profile to the home network.

Example of a home agent configuration in gateway mode

A home agent in gateway mode receives GRE-encapsulated IP packets from the foreign agent, strips off the encapsulation, and passes the packets across a WAN connection to the home network. The WAN connection must already be up, because tunneled data does not bring up a connection.

To enable hosts and routers on the home network to reach the mobile client, you must configure a static route in the Customer Premise Equipment (CPE) router on the home network (not in the home agent). The static route must specify the home agent as the route to the mobile client. That is, the route's destination address specifies the address of the mobile client, and its gateway address specifies the IP address of the home agent.

Following is an example of a procedure that configures the MAX TNT to operate as a home agent in gateway mode and configures a Connection profile for connecting to the home network:

1 Open the ATMP profile (create it if necessary):

```
admin> new atmp
ATMP read
```

2 Configure the home agent parameters, and then write the profile:

```
admin> set agent-mode = home-agent

admin> set agent-type = gateway-home-agent

admin> set home-agent-password = my-password

admin> write
ATMP written
```

3 Create an IP-routing Connection profile to the foreign agent.

For information about configuring IP routing connections, see the *MAX TNT Network Configuration Guide*. For information about configuring connections in RADIUS, see the *MAX TNT RADIUS Configuration Guide.*

4 Create a Connection profile for connecting to the home network.

The following example shows the ATMP configuration of a Connection profile for connecting to the home network. For more information about configuring nailed connections, see the *MAX TNT Hardware Installation Guide* and *MAX TNT Network Configuration Guide*.

```
admin> read connection homenet
CONNECTION/homenet read

admin> list tunnel-options
profile-type = disabled
max-tunnels = 0
primary-home-agent = ""
secondary-home-agent = ""
udp-port = 5150
home-agent-password = ""
home-network-name = ""

admin> set profile-type = gateway-profile

admin> set max-tunnels = 50

admin> write
CONNECTION/ca1 written
```

The profile for the connection to the home network must be a local profile (it cannot be specified in RADIUS), and the name of this Connection profile must be specified in the Home-Network-Name parameter or Ascend-Home-Network-Name attribute in the mobile client's configured profile.

Example of a home agent configuration in router mode

A home agent in router mode receives GRE-encapsulated IP packets from the foreign agent, strips off the encapsulation, and then routes the packets to the home network in the usual way. It also adds to its routing table a host route to the mobile client.

If you enable RIP on the home agent's local interfaces, other hosts and networks can route to the mobile client. Enabling RIP is particularly useful if the home network is one or more hops away from the home agent's Ethernet. If RIP is turned off, other routers require static routes that specify the home agent as the route to the mobile client.

Note: If the home agent's local interface is the home network (a direct connection), you should turn on proxy ARP in the home agent to enable local hosts to ARP for the mobile client.

Following is an example of a procedure that configures the MAX TNT to operate as a home agent in router mode:

1 Open the ATMP profile (create it if necessary):

```
admin> new atmp
ATMP read
```

2 Configure the home agent parameters, and then write the profile:

```
admin> set agent-mode = home-agent
admin> set agent-type = router-home-agent
admin> set home-agent-password = my-password
admin> write
ATMP written
```

3 Create an IP-routing Connection profile to the foreign agent.

For information about configuring IP routing connections, see the *MAX TNT Network Configuration Guide*. For information about configuring connections in RADIUS, see the *MAX TNT RADIUS Configuration Guide.*

Example of a home-and-foreign agent configuration

In Figure 13, the MAX TNT operates as a home agent for network B and as a foreign agent for network A. For each tunnel, it meets all of the same requirements described in previous sections for a home agent or foreign agent.

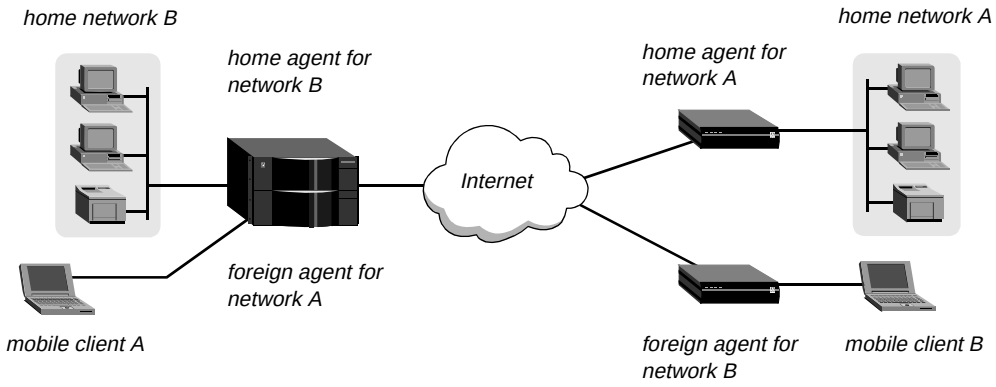

Figure 13. MAX TNT acting as both home agent and foreign agent

Following is an example of a procedure that configures the MAX TNT in Figure 13 to operate as a home agent in router mode for network B, and as a foreign agent for network A:

1 Open the ATMP profile (create it if necessary):

```
admin> new atmp
ATMP read
```

2 Specify home-and-foreign-agent mode:

```
admin> set agent-mode = home-and-foreign-agent
```

3 Configure the home agent parameters, and then write the profile:

```
admin> set agent-type = router-home-agent
admin> set home-agent-password = my-password
admin> write
ATMP written
```

4 Configure a Connection profile for mobile client A:

```
admin> read connection mclientA
CONNECTION/mclientA read

admin> list tunnel-options
profile-type = disabled
max-tunnels = 0
primary-home-agent = ""
secondary-home-agent = ""
udp-port = 5150
home-agent-password = ""
home-network-name = ""

admin> set profile-type = mobile-client

admin> set primary-home-agent = max.home-network-A.com

admin> set home-agent-password = homenetA-password

admin> write
CONNECTION/mclientA written
```

ATMP support for connecting to a GRF switch

Two parameters, MTU-Limit and Force-Fragmentation, have been added to the ATMP profile to enable the MAX TNT to operate as an ATMP foreign agent tunneling to a GRF switch configured as home agent. Figure 14 shows the MAX TNT tunneling to a GRF across a 100-BaseT Ethernet segment:

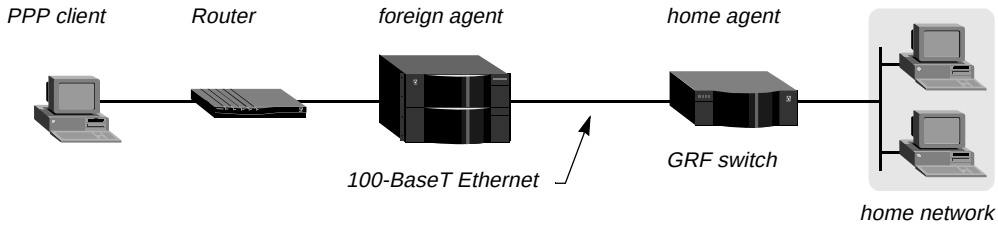

Figure 14. ATMP tunnel to GRF switch

The MAX TNT can receive packets that are larger than the Ethernet Maximum Transmission Unit (MTU) from a PPP client that logs in through a remote access router, such as the Pipeline router in Figure 14. In addition, some clients might send frames larger than the negotiated Maximum Receive Unit (MRU) with the Don't Fragment (DF) bit set, a behavior intended to discover the path's MTU.

In either case, the unit dialing in negotiates an MRU that is large enough to make fragmentation unnecessary. But when the MAX TNT encapsulates the packet in GRE, it adds an 8-byte GRE header and a 20-byte IP header, which can make the packet size larger than the MTU of the tunneled link. So, the packet must either be fragmented or rejected. However, having a very high aggregate forwarding rate at the ATMP gateway requires that the foreign agent fragment the IP packets before encapsulation rather than after.

The following parameters (shown with their default values) enable the MAX TNT to fragment the IP packets before encapsulating them in GRE:

```
ATMP
   mtu-limit = 0
   force-fragmentation = no
```

Prefragmentation is required when the cost of reassembling IP packets should be shifted to the end clients instead of loading the gateway, as is the case with the GRF.

Setting the MTU limit

To determine the maximum size packet it can send to the home agent without prefragmentation, the MAX TNT checks the value of the MTU-Limit parameter. If the parameter is set to zero, the MAX TNT does not perform prefragmentation. If the parameter is set to a non-zero value, it fragments packets at the specified size. For example, to enable the MAX TNT to send full 1500-byte frames on Ethernet, set the MTU-Limit parameter as shown in the following example:

```
admin> read atmp
ATMP read

admin> set mtu-limit = 1472

admin> write
ATMP written
```

Forcing fragmentation

Typically, if the incoming frame has the DF bit set and is too large to tunnel to the ATMP home agent, the MAX TNT returns an ICMP message that informs the client that the host is unreachable without fragmentation. This standard, expected behavior improves end-to-end performance by enabling the client to determine the path's MTU and thereby avoid unnecessary fragmentation and reassembly.

The Force-Fragmentation parameter changes this standard behavior. When prefragmentation is enabled (MTU-Limit parameter set to a non-zero value), and the Force-Fragmentation parameter is set to Yes, the MAX TNT ignores the DF bit and fragments the frame anyway. Following is an example that enables the MAX TNT to fragment and tunnel 1500-byte frames across 100-BaseT, even if the frames' DF bits are set:

```
admin> read atmp
ATMP read

admin> set mtu-limit = 1472

admin> set force-fragmentation = yes

admin> write
ATMP written
```

Note: The Force-Fragmentation parameter enables behavior that is *not standard* and might cause problems. In particular, setting this parameter to Yes disables MTU discovery mechanisms.

Home agent inactivity timers for ATMP tunnels

When an ATMP foreign agent restarts, tunnels that were established to a home agent are not normally cleared, because the home agent is not informed that the mobile clients are no longer connected. So, the home agent does not release the resources held by the unused tunnel. To enable the home agent to reclaim the resources held by unused tunnels, ATMP home agents can now set an inactivity timer using the following parameter, which is shown with its default value:

```
ATMP
   idle-timer = 0
```

The inactivity timer runs only on the home agent side.Its value specifies the number of minutes—from 0 to 65535— that the home agent maintains an idle tunnel before disconnecting it. A value of 0 disables the timer, which means that established tunnels remain connected forever. The setting affects only tunnels created after the timer was set. Existing tunnels are not affected.

In the following example, the timer is set to 30 minutes:

```
admin> new atmp
ATMP read

admin> list
agent-mode = tunnel-disabled
agent-type = gateway-home-agent
udp-port = 5150
home-agent-password = ""
retry-timeout = 3
retry-limit = 10
idle-timer = 0
mtu-limit = 0
force-fragmentation = no

admin> set idle-timer = 30

admin> write
ATMP written
```

New administrative features

A progress indication for Load or Save commands

Load and Save commands now provide an indication that they were progressing during the time required to complete the command. The progress indication is a spinning cursor that is rotated once for every few data packets processed.

Changes to Syslog output

The MAX TNT now reports additional session information about various errors logged via Syslog. The information should assist in identifying all messages associated with a session. It also supports a new Syslog message to provide the disconnect and progress codes for a given session. Error messages associated with a session can contain the following information:

Syntax	Description
`[shelf/slot/line/channel]`	a physical channel identifier
`[MBID xxx]`	a session identifier
`[name]`	the authenticated name
`[ calling -> called ]`	the calling number or the called number, or both

For a given session identifier, multiple physical channel identifiers are possible (for example, one identifier might be for the T1 line, and another for the HDLC channel or modem number). This is shown in the sample log below, in which messages include the MBID, DNIS, and CLID in brackets. Note that slot 1/2 is an 8T1 card, and slot 1/3 is a 48-modem card.

```
...: [1/2/1/2] [MBID 1; 9995551212 -> 7898] Incoming Call
...: [1/3/1/0] [MBID 1; 9995551212 -> 7898] Assigned to port
...: [1/2/1/2] [MBID 1; 9995551212 -> 7898] Call Connected
...: [1/3/1/0] [MBID 1] [balsup-pc] LAN session up: <balsup-pc>
...: [1/3/1/0] [MBID 1] [balsup-pc] LAN session down: <balsup-pc>
...: [1/3/1/0] [MBID 1; 9995551212 -> 7898] Call Terminated
...: [1/3/1/0] [MBID 1] [balsup-pc] : STOP: 'balsup-pc'; cause 45.; progress
60.; host 10.1.26.2
```

Fatal crash information on console

If the MAX TNT crashes, it now prints a stack trace to the console serial port at the bitrate defined in the Serial profile. The following information is included:

```
FE: N, Load: loadname, Version: version
Stack trace: 0xaddr-0 0xaddr-1 0xaddr-2 0xaddr-3 0xaddr-4 0xaddr-5
```

In the first line of output, *N* is a fatal error number, *loadname* is the name of the load (for example, tntsr or tntmdm56k), and *version* is the software version (for example, 2.0.0).

The second line of output displays the top six program counter addresses from the execution stack active at the time of the crash.

Finger (RFC 1288) support and Userstat enhancements

The MAX TNT now supports additional options for displaying user session information. If Finger is enabled in the IP-Global profile, the MAX TNT can return user information to a remote Finger query, such as UNIX client. In addition, the native Userstat command now includes additional options for displaying information in a 140-character-wide table format and includes additional fields.

Finger user information protocol

Finger is described in RFC 1288. To enable it in the MAX TNT, set the following parameter to Yes:

```
IP-GLOBAL
   finger = yes
```

The default value for this parameter is No, which causes the MAX TNT to reject queries from Finger clients with the following message:

```
Finger online user list denied.
```

Setting the Finger parameter to Yes enables the MAX TNT to accept Finger queries and return the requested active session details to a remote client. The client can ask for short or wide format; for example, a UNIX client can request the wide format by using the –l option. The following command:

```
# finger @tnt1
```

displays the narrow (80-character wide) format, and the following command

```
# finger -l @tnt1
```

displays a wide (140-character-wide format of session information for the system named "tnt1." The client can also request the details of all sessions, or of a single session. For example, to request information about a single user named Gavin:

```
# finger gavin@tnt1
```

The Finger forwarding service, which uses the hostname format "@host1@host2", is not supported. If the remote client uses the forwarding request format, the client sees the following message:

```
Finger forwarding service denied.
```

Userstat options

The MAX TNT Userstat command displays session status information. In previous releases, the information was always 80 characters wide, for example:

```
admin> userstat -s
SessionID  Line/Chan  Slot:Item     Rate     Svc Address       Username
228687860  1.01.02/01 1:03:01/01    56K      PPP 10.100.0.1    barney
228687861  1.02.03/02 1:04:02/00    28800    PPP 10.168.6.24   jake
<end user list>  2 active user(s)
```

For information on the fields shown in the output immediately above, see the *MAX TNT Reference Guide*. In this release, the Userstat command supports two new options. The –l option produced a 140-character-wide format with additional fields, and the –d option to cause the output to be "dumped" to the display rather than being shown one page at a time. When the Userstat command line includes the –l option, the following additional fields are reported:

Table 6. Additional fields displayed for Userstat –l

Field Name	Description
Dialed#	The number dialed to initiate this session.
ConnTime	The amount of time in hours:minutes:seconds format since the session was established.
IdleTime	The amount of time in hours:minutes:seconds format since data was last transmitted across the connection.

Userstat –k to terminate user sessions

The Userstat command can now terminate a user session that uses one of the following service types: PPP, SLIP, MPP, Telnet, Telnet binary, Raw TCP, or terminal server. The command cannot terminate Frame Relay or DTPT service types. To terminate a user session, include the –k option on the command line; for example:

```
admin> userstat
SessionID  Line/Chan  Slot:Item   Rate  Svc Address         Username
246986325  1.01.02/01 1:13:01/000 33600 PPP 100.100.8.2     100.100.8.2
<end user list>  1 active user(s)
```

```
admin> userstat -k 246986325
Session 246986325 cleared
```

Slot card and system uptime information

A new command, Uptime, reports how long the system and its individual cards have been up. To enable network management stations to obtain uptime information, a new MIB object named slotLastChange has been added to the Ascend Enterprise MIB.

The Uptime command uses the following syntax:

```
admin> help uptime
uptime usage: uptime [ [ -a ] | [ [ shelf ] slot ] ]
uptime                  display the TNT system uptime.
uptime slot             display the TNT slot card uptime.
uptime shelf slot       display the TNT slot card uptime.
uptime -a               display the uptime for all TNT slot cards.
```

Without an argument, the command displays system uptime. But in the following example, the command displays the uptime for all slot cards in the UP state (cards that are not in the UP state are not reported):

```
admin> uptime -a
13:26:54
{ shelf-1 slot-1 }                 8t1-card    0 days 00:07:04
{ shelf-1 slot-2 }              48modem-card    0 days 00:06:00
{ shelf-1 slot-4 }              128hdlc-card    0 days 00:05:20
{ shelf-1 slot-5 }               4ether-card    0 days 00:06:38
```

Uptime displays the current time (13:26:54 in the preceding example), identifies the slot card, and then displays the length of time the system has been up in days followed by hours:minutes:seconds. The following example shows how long a modem card in slot 2 has been up:

```
admin> uptime 1 2
13:26:39 { shelf-1 slot-2 }         48modem-card 0000 days 00:05:53
```

The slotLastChange object has the following definition in the Ascend Enterprise MIB:

```
slotLastChange   OBJECT-TYPE
     SYNTAX      TimeTicks
     ACCESS      read-only
     STATUS      mandatory
     DESCRIPTION "The value of sysUpTime at the time the TNT slot card
                  entered its current state. For non-TNT systems 0 is
                  always reported."
     ::= { slotEntry 9 }
```

The slotLastChange variable reports the value of sysUpTime at the time the slot card entered its current state.

Frame Relay information reported on SWAN cards

The Ascend Enterprise MIB eventGroup, callStatusGroup, and sessionStatusGroup now contain information for Frame Relay profiles on SWAN slot cards. As a result, the status window in the command-line interface now shows the name of Frame Relay profiles on SWAN lines. For example:

```
5 Connections, 5 Sessions       |tnt-ma Status
0001 frswan1 FRY 13/01/1 64000  |Serial number: 7021016        Version: 2.0.0
0002 frswan2 FRY 13/02/1 64000|
0005 fr2      FRY 15/07/2 1984K| Rx Pkt:       2002
0006 fr1      FRY 15/01/2 1984K| Tx Pkt:       1844
0007 fr4      FRY 15/05/2 1984K|     Col:          4
                                |
                                |-------------------------------------------
                                | 1/13/01 LA n
                                | 1/13/02 LA n
                                | 1/15/01 TE .nnnnnnn nnnnnnnn nnnnnnnn
                                | 1/15/02 TE .------- -------- s------- ------
                                | 1/15/04 RA ........ ........ ........
                                | 1/15/05 TE .nnnnnnn nnnnnnnn nnnnnnnn
                                | 1/15/07 TE .nnnnnnn nnnnnnnn nnnnnnnn
--------------------------------------------------------------------------------
```

New disconnect code (210—slot card down)

Previously, when a slot card went down, the disconnect reason was reported as either 185 (remote end hung up) or 11 (modem loss carrier). For the MAX TNT, the 210 (slot card down) disconnect code has been added to indicate calls terminated due to a slot card that is down or entering the DOWN state. The 210 disconnect code is retrievable from the Ascend accounting events MIB, but it is not reported in RADIUS accounting Stop packets at this release.

It may take up to a minute to detect a card that is down or entering the DOWN state. If the caller terminates the call due to lack of response during that time, the disconnect code 185 or 11 will be reported.

The 210 disconnect code is likely to be reported under the following conditions:

- The card is administratively downed. This may take 2 to 6 seconds to detect.
- The card goes from the UP to the DOWN state. This may take 2 to 6 seconds to detect. If the card fails POST, no calls should be active, so the 210 code will not appear.
- The card reports diagnostics while dumping core. A transition from UP to DIAG indicates a card that is dumping core. This may take 2 to 6 seconds to detect.
- The card dies but stays in UP state. If the remote end doesn't disconnect first, the watchdog timer expires and the shelf resets the card automatically. The 210 is reported a few seconds after card is reset. The total time before 210 is reported is under 70 seconds.

Separate transmit and receive data rates reported

In previous releases, reported data rates reflected the receive rate only, and the transmit rate was not reported. Now, separate transmit (xmit) and receive (recv) data rates are supported in the Ascend enterprise call.mib and event.mib, in Userstat command output, and in RADIUS accounting. In other cases, where a single data-rate field is reported, the data rate still represents the receive rate.

For ISDN calls, the transmit rate shows the transmit data rate. For analog calls, it shows the modem baud rate at the time of the initial connection.

Ascend Call and Event MIBs

The Call MIB now includes callStatusXmitRate (callStatusEntry 14) and callActiveXmitRate (callActiveEntry 14). Following are the new object definitions:

```
callStatusXmitRate   OBJECT-TYPE
     SYNTAX          INTEGER
     ACCESS          read-only
     STATUS          mandatory
     DESCRIPTION     "The transmit rate for ISDN calls or the baud rate
                     for modem calls. A value of 0 is returned if entry
                     is invalid."
     ::= { callStatusEntry 14 }

callActiveXmitRate   OBJECT-TYPE
     SYNTAX          INTEGER
     ACCESS          read-only
     STATUS          mandatory
     DESCRIPTION     "The transmit rate for ISDN calls or the baud rate
                     for modem calls."
     ::= { callActiveEntry 14 }
```

The Event MIB now includes eventXmitRate (eventEntry 23). Following is the object definition:

```
eventXmitRate        OBJECT-TYPE
     SYNTAX          INTEGER
     ACCESS          read-only
     STATUS          mandatory
     DESCRIPTION     "The transmit data rate for ISDN calls or the baud
                rate for modem calls. Rate is given as bits-per-second.
                Applicable for all 'eventType's except callCleared(3).
                For callCleared(3), 0 will be returned. For modem
                calls, value will be 0 for callAnswered(2) events
                since rate is unknown at the time incoming call is
                detected."
     ::= { eventEntry 23 }
```

Userstat output

The Userstat command now reports both the transmit and receive rates. The following example output shows an active session from a 56K modem user:

```
admin> userstat
SessionID  Line/Chan  Slot:Item   Tx/Rx  Rate Svc Address         Username
247351303  3.01.08/12 3:04:05/000 48000/31200 PPP 11.168.6.124    pc-3
<end user list>  1 active user(s)
```

RADIUS accounting

RADIUS accounting now reports the receive date rate in the Ascend-Data-Rate attribute and the transmit data rate in the Ascend-Xmit-Rate attribute. For example, the following RADIUS record was generated by a 56K modem user session:

```
Sat Nov 8 08:11:37 1997
        User-Name = "pc-3"
        NAS-Identifier = 11.168.6.124
        NAS-Port = 33003
```

```
NAS-Port-Type = Async
Acct-Status-Type = Stop
Acct-Delay-Time = 0
Acct-Session-Id = "247351302"
Acct-Authentic = RADIUS
Acct-Session-Time = 36
Acct-Input-Octets = 659
Acct-Output-Octets = 457
Acct-Input-Packets = 18
Acct-Output-Packets = 14
Ascend-Disconnect-Cause = 100
Ascend-Connect-Progress = 60
Ascend-Xmit_rate = 48000
Ascend-Data-Rate = 31200
Ascend-PreSession-Time = 24
Ascend-Pre-Input-Octets = 464
Ascend-Pre-Output-Octets = 423
Ascend-Pre-Input-Packets = 12
Ascend-Pre-Output-Packets = 12
Ascend-Modem-PortNo = 4
Ascend-Modem-SlotNo = 4
Ascend-Modem-ShelfNo = 3
Caller-Id = "5108641846"
Client-Port-DNIS = "7835"
Framed-Protocol = PPP
Framed-Address = 11.168.6.124
```

IP-Pools command

To view the status of the IP address pools configured in the IP-Global profile, use the IP-Pools command, as shown in the following example:

```
admin> ip-pools
Pool#               Base         Count      InUse
1              10.154.3.50        50        0
3              10.154.3.150       50        1

Number of remaining allocated addresses:     99
```

The sample output shows two configured pools, with the base address, address count, and number of addresses in use for each pool.

Netstat command displays active sockets on slot cards

The Netstat command now displays the shelf/slot and socket numbers for active UDP and TCP sockets on slot cards. For example:

```
admin> netstat
udp:
-Socket-  Local Port  InQLen  InQMax    InQDrops    Total Rx
1/c    0        1023       0       1           0           0
1/c    1       route       0       0           0          25
1/c    2        echo       0      32           0           0
1/c    3         ntp       0      32           0           1
1/c    4        1022       0     128           0           0
1/c    5        snmp       0     128           0           0
1/1    0           1       0     256           0           0
```

```
1/1    1        1018       0     128           0           0
1/3    0           3       0     256           0           0
1/3    1        1021       0     128           0           0
1/5    0           5       0     256           0           0
1/5    1        1020       0     128           0           0
1/8    0           8       0     256           0           0
1/8    1        1019       0     128           0           0

tcp:
-Socket- Local                    Remote                State
1/c    0  lab-60.eng.ascen.telnet *.*                  LISTEN
```

In the preceding output, the Socket field shows the shelf/slot followed by the socket number.

Netstat reports TCP statistics collected from slot cards

The Netstat command now displays TCP statistics collected from slot cards, as well as the shelf-controller statistics reported in previous releases. In addition, several new statistics are included. The following command now displays the total of TCP statistics from the shelf-controller and all slot cards:

```
admin> netstat -s tcp
tcp:
0 active opens
3 passive opens
0 connect attempts failed
0 connections were reset
1 connections currently established
3565 segments received
0 segments received out of order
3620 segments transmitted
0 segments retransmitted
1 active closes
1 passive closes
0 disconnects while awaiting retransmission
```

Netstat command reports Finger service (port 79)

Netstat command output now reports the Finger service (port 79). For example, the following output reports both Finger and Telnet services:

```
admin> netstat tcp
tcp:
-Socket- Local                 Remote                       State
1/c    0 host1.eng.abc.finger *.*                         LISTEN
1/c    1 host1.eng.abc.telnet *.*                         LISTEN
1/c    2 host1.eng.abc.finger ridgeback.eng.abc..38040   TIME-WAIT
1/c    3 host1.eng.abc.finger ridgeback.eng.abc..38041 ESTABLISHED
1/c    4 host1.eng.abc.finger ridgeback.eng.abc..38042 ESTABLISHED
1/c    5 host1.eng.abc.finger ridgeback.eng.abc..38043 ESTABLISHED
1/c    6 host1.eng.abc.finger ridgeback.eng.abc..38044 ESTABLISHED
```

In the example output, the *name.service* field is fixed width, and the *name* (a hostname or full domain name) has been truncated to accommodate that width.

For TCP, Netstat now reports the following services:

Service	TCP port number
Telnet	23
TACACS+	49
Finger	79

For UDP, Netstat now reports the following services:

Service	UDP port number
Route	520
Echo	7
NTP	123
SNMP	161
SNMPTrap	162

As always, if the port being used is not found among these named services, it is printed as a number. Also, if the –n option is used on the Netstat command line, numeric addresses and port numbers are always reported instead of names.

Add DNIS and CLID to Syslog messages

DNIS and CLID information are displayed in Syslog messages that relate to a call, provided that the information is known. Following is an example that shows the DNIS 7895 in Syslog messages:

```
LOG info, Shelf 1, Controller, Time: 17:48:56--
† shelf 1, slot 1, line 1, channel 6, dnis 7895, Incoming Call, MBID 001

LOG info, Shelf 1, Controller, Time: 17:48:56--
† shelf 1, slot 2, dnis 7895, Assigned to port, MBID 001

LOG info, Shelf 1, Controller, Time: 17:48:57--
† shelf 1, slot 1, line 1, channel 6, dnis 7895, Call Connected, MBID 001

LOG warning, Shelf 1, Controller, Time: 17:49:20--
† shelf 1, slot 1, line 1, channel 6, dnis 7895, Call Disconnected

LOG info, Shelf 1, Controller, Time: 17:49:20--
† shelf 1, slot 2, Call Terminated
```

Call logging using the RADIUS accounting protocol

Call logging is a RADIUS-accounting based feature for logging call information from the MAX TNT. Its main purpose is to duplicate accounting information for sites that wish to keep accounting records separate from other groups that might need call-logging details to manage resources or troubleshoot call problems.

Once you have configured call logging, the MAX TNT sends Start Session, Stop Session, and Failure-to-Start Session packets to a call-log host. A call-log host is a local host that supports the RADIUS accounting protocol and is configured properly to communicate with the MAX

TNT (for example, a RADIUS accounting server or a host running NavisAccess). The call-log information is sent independently of RADIUS accounting records. If both call logging and RADIUS accounting are in use, the information is sent in parallel.

You set the following parameters, shown with their default values, to configure the MAX TNT to communicate with one or more call-log hosts:

```
CALL-LOGGING
   call-log-enable = no
   call-log-host-1 = 0.0.0.0
   call-log-host-2 = 0.0.0.0
   call-log-host-3 = 0.0.0.0
   call-log-port = 0
   call-log-key = ""
   call-log-timeout = 0
   call-log-id-base = acct-base-10
   call-log-reset-time = 0
   call-log-stop-only = yes
   call-log-limit-retry = 0
```

Following is an example of a procedure that enables call logging and specifies one call-log host on the local network:

```
admin> read call-logging
CALL-LOGGING read

admin> list
call-log-enable = no
call-log-host-1 = 0.0.0.0
call-log-host-2 = 0.0.0.0
call-log-host-3 = 0.0.0.0
call-log-port = 0
call-log-key = ""
call-log-timeout = 0
call-log-id-base = acct-base-10
call-log-reset-time = 0
call-log-stop-only = yes
call-log-limit-retry = 0

admin> set call-log-enable = yes

admin> set call-log-host-1 = 10.2.3.4

admin> write
CALL-LOGGING written
```

The parameters shown have the following functions:

Parameter	Function
Call-Log-Enable	Enables call logging. If set to No, none of the other call-logging parameters apply. If set to Yes, you must specify the IP address of at least one call-log host in the Call-Log-Host-*N* parameters

Parameter	Function
Call-Log-Host-*N*	Each specifies the IP address of one call-log host. The MAX TNT first tries to connect to server #1 for call-logging. If it receives no response, it tries to connect to server #2. If it receives no response from server #2, it tries server #3. If the MAX TNT connects to a server other than server #1, it continues to use that server until it fails to service requests, even if the first server has come online again.
Call-Log-Port	Specifies the UDP destination port to use for call-logging requests. The default value of 0 (zero) indicates any UDP port. If you specify a different number, the call-log host must specify the same port number (the numbers must match).
Call-Log-Key	A shared secret that enables the server to receive data from the MAX TNT. The value must match the configured shared secret on the call-log host.
Call-Log-Timeout	Specifies the number of seconds the MAX TNT waits for a response to a call-logging request. It can be set to a value of from 1 to 10. The default value is 0 (zero), which disables the timer.
Call-Log-ID-Base	Specifies whether the MAX TNT presents a session ID to the call-log host in base 10 or base 16. The default is base 10.
Call-Log-Reset-Time	Indicates the number of seconds that must elapse before the MAX TNT returns to using the primary call-log host (Call-Log-Host-1). The default value of 0 (zero) disables the reset to the primary call-log host.
Call-Log-Stop-Only	Specifies whether the MAX TNT should send an Stop packet with no user name. The MAX TNT typically sends Start and Stop packets to record connections. Authentication is required to send a Start packet. There are situations that the MAX TNT will send an Stop packet without having sent an Start packet in which case the Stop packets have no user name. The default value for Call-Log-Stop-Only is Yes. You can set it to No to prevent the unit from sending Stop packets with no user name.

Call-Log-Limit-Retry	If the server does not acknowledge a Start or Stop packet within the number of seconds specified in Call-Log-Timeout, the MAX TNT tries again, resending the packet until the server responds or the packet is dropped because the queue is full. The Call-Log-Limit-Retry parameter sets the maximum number of retries for these packets. The default value of 0 (zero) indicates an unlimited number of retries. There is minimum of 1 retry. The following example limits the number of retries to 10. There will be a total of 11 attempts: the original attempt plus 10 retries.
	`admin> **read call-logging**`
	`CALL-LOGGING read`
	`admin> **set call-log-limit-retry = 10**`
	`admin> **write**`
	`CALL-LOGGING written`

Configurable session listing output

Administrators can display a list of active sessions by using the Userstat command or sending a Finger query to a MAX TNT that has enabled Finger in the IP-Global profile. Both methods of displaying the list of sessions use the same output format, for example:

```
SessionID  Line/Chan  Slot:Item   Tx/Rx Rate  Svc Address       Username

250808749  1.01.01/02 1:04:03/018 56000/56000 PPP 100.100.1.8     max8
250808750  1.01.01/01 1:04:03/019 56000/56000 PPP 100.100.1.9     max9
<end user list>  2 active user(s)
```

This output format can now be customized. Following is the relevant parameter, which is shown with its default value:

```
SYSTEM
   userstat-format = %i %l %s %r %d %a %u %c %t %n
```

The Userstat-Format parameter specifies a series of conversion strings (a character preceded by a percent-sign), which are described in Table 1. You can enter up to 72 characters. The maximum width of the output string depends on the width of the fields present in the session listing output (see Table 1). If you enter a character without a percent-sign, it is printed as a literal character in the session listing output.

Table 1. Userstat-Format conversion strings

String	Field Width	Output Text	Meaning
%i	10	SessionID	Unique ID assigned to the session
%l	10	Line/Chan	Physical address (shelf.slot.line/chan)
%s	11	Slot:Item	shelf:slot:item/logical-item of the host port
%r	11	Tx/Rx Rate	Transmit and receive rates
%d	3	Svc	A three-letter code showing the type of service
%a	15	Address	IP address
%u	14	Username	Connection profile name
%c	10	ConnTime	Amount of time connected in hours:minutes:seconds
%t	10	IdleTime	Amount of time idle in hours:minutes:seconds
%n	24	Dialed#	Number dialed if known

The default value of the Userstat-Format parameter causes the standard session listing output format for both the Userstat command and responses to Finger queries. For example:

```
admin> get system userstat-format
userstat-format = %i %l %s %r %d %a %u %c %t %n
```

```
admin> userstat

SessionID  Line/Chan  Slot:Item   Tx/Rx Rate  Svc Address       Username

250808749  1.01.01/02 1:04:03/018 56000/56000 PPP 100.100.1.8     max8
250808750  1.01.01/01 1:04:03/019 56000/56000 PPP 100.100.1.9     max9
<end user list>  2 active user(s)
```

The following example customizes the session listing output to include only the Username, Svc, and ConnTime information, with an at-sign appearing between the service and connection-time for each session:

```
admin> read system
SYSTEM read

admin> set userstat-format = %u (%d) @ %c

admin> write
SYSTEM written

admin> userstat

Username       Svc    ConnTime

joeb          (PPP) @ 1:22:34
jimmyq        (PPP) @ 3:44:19
sallyg        (PPP) @ 5:12:56

<end user list>  3 active user(s)
```

External authentication and accounting features

Conflicts between RADIUS and local configurations resolved

Because external RADIUS and local profiles are integrated into the same database, the MAX TNT platform does not allow external RADIUS profiles and local profiles with the same name. If the administrator executes a Write command to save a local profile, and there is already an external RADIUS profile with the same name, the following error message appears:

```
error: Cannot overwrite duplicate external CONNECTION/testdlci profile
```

When external RADIUS profiles are retrieved, if the profile index is a duplicate of a local profile of the same type, the following message is logged at the Error level:

```
"Could not add duplicate RADIUS CONNECTION profile 'testdlci'"
```

In addition, external RADIUS profiles now correctly report the time they were last refreshed. Note that configurations saved via TFTP or the serial console do not include profiles that were retrieved from an external database or that are read-only.

Setting the number of RADIUS accounting retries

When the MAX TNT is configured for RADIUS accounting, it sends accounting Start and Stop packets to the RADIUS server to record connections. If the server does not acknowledge a packet within the number of seconds in Acct-Timeout, the MAX TNT tries again, resending the packet until the server responds or the packet is dropped because the queue is full. The following new parameter specifies the maximum number of retries for accounting packets:

```
EXTERNAL-AUTH
   rad-acct-client
      acct-limit-retry = 0
```

The default value of 0 (zero) indicates an unlimited number of retries, which means that the MAX TNT resends the packet until the server responds or the packet is dropped because the queue is full. There is minimum of 1 retry. The following example limits the number of retries to 10. There will be a total of 11 attempts: the original attempt plus 10 retries.

```
admin> read external-auth
EXTERNAL-AUTH read

admin> list rad-acct-client
acct-server-1 = 10.2.3.56
acct-server-2 = 10.7.8.62
acct-server-3 = 10.5.6.11
acct-port = 1646
acct-src-port = 0
acct-key = *******
acct-timeout = 5
acct-sess-interval = 0
acct-id-base = acct-base-10
acct-reset-time = 0
acct-checkpoint = 0
acct-limit-retry = 0
acct-stop-only = yes

admin> set acct-limit-retry = 10

admin> write
EXTERNAL-AUTH written
```

NAS-port type added to RADIUS accounting

The MAX TNT bit-encodes the 16-bit NAS-Port number sent to RADIUS accounting daemons to indicate the shelf, slot, line, and channel on which a call was received. It now also includes a NAS-Port-Type value to indicate whether the established session uses asynchronous or synchronous transmission.

The NAS-Port-Type may be one of the following values:

- NAS_Port_Type_Sync (1) for synchronous sessions.
- NAS_Port_Type_Async (0) for asynchronous sessions.

RADIUS accounting for failed authentication

RADIUS accounting keeps records of sessions, which are typically used for billing and security tracking. When RADIUS accounting is in use, the MAX TNT sends a Stop packet to the RADIUS server when a session terminates. RADIUS accounting Stop packets are normally sent for authenticated connections, connections that are dropped before authentication, and connections that fail authentication.

You can configure the MAX TNT not to send Stop packets for connections that fail authentication by changing the setting of the following parameter:

```
EXTERNAL-AUTH
   radius-acct-client
      acct-drop-stop-on-auth-fail = no
```

The default value is No. If the parameter is set to Yes, RADIUS accounting Stop packets are not sent for connections that fail authentication. The commands in the following example configure the MAX TNT not to send Stop packets for connections that fail authentication:

```
admin> read external-auth
EXTERNAL-AUTH read

admin> set radius-acct-client acct-drop-stop-on-auth-fail = yes

admin> write
EXTERNAL-AUTH written
```

For more information about RADIUS accounting, see the *MAX TNT RADIUS Configuration Guide*.

Preventing accounting Stop packets with no user name

When the MAX TNT is configured for RADIUS accounting, it sends accounting Start and Stop packets to the RADIUS server to record connections. Authentication is required to send a Start packet. There are situations in which the MAX TNT will send an accounting Stop packet without having sent an accounting Start packet, in which case the Stop packets have no user name.

The following parameter has been added to specify whether the MAX TNT should send an accounting Stop packet with no user name:

```
EXTERNAL-AUTH
   rad-acct-client
      acct-stop-only = yes
```

The default value is Yes. You can set this parameter to No to prevent the unit from sending Stop packets with no user name to the RADIUS server. For example:

```
admin> read external-auth
EXTERNAL-AUTH read

admin> list rad-acct-client
acct-server-1 = 10.2.3.56
acct-server-2 = 10.7.8.62
acct-server-3 = 10.5.6.11
acct-port = 1646
acct-src-port = 0
acct-key = *******
acct-timeout = 5
acct-sess-interval = 0
acct-id-base = acct-base-10
acct-reset-time = 0
acct-checkpoint = 0
acct-limit-retry = 0
acct-stop-only = yes

admin> set acct-stop-only = no

admin> write
EXTERNAL-AUTH written
```

Distinct ID sequences for RADIUS authentication and accounting

RADIUS uses an ID value to aid in request-response matching. By default, the MAX TNT uses a single sequence space for the RADIUS ID number in all RADIUS messages. A single space limits the number of IDs available for assignment to 256. In this release, you can configure distinct ID sequence spaces for RADIUS accounting and authentication packets.

When you configure the MAX TNT to use distinct ID sequence spaces, the RADIUS server must perform additional checks to detect duplicates. The server should check the RADIUS ID value as well as the service type and destination UDP port in each packet. The service type can be determined by sorting all values of the code field into two classes—auth and acct—and then comparing the received code value to determine to which class it belongs. The destination UDP port can be the same for both services when a single RADIUS server performs both services.

To configure the MAX TNT to use distinct ID sequence spaces, use the following parameter (shown with its default setting):

```
EXTERNAL-AUTH
   rad-id-space = unified
```

When rad-id-space is set to Unified, RADIUS authentication and accounting packets share the same ID sequence space, so a combined total of 256 authentication and accounting packets are sent before the ID sequence rolls over. The commands in the following example configure the MAX TNT to use distinct sequence spaces:

```
admin> read external
EXTERNAL-AUTH read

admin> set rad-id-space = distinct

admin> write
EXTERNAL-AUTH written
```

When RAD-ID-Space is set to Distinct, RADIUS authentication and accounting packets do not share the same ID sequence space. The MAX TNT can send a total of 256 authentication packets before the authentication ID sequence rolls over, and 256 accounting packets before the accounting ID sequence rolls over. Three sequence spaces are allocated: one for the Unified sequence space, and one each for the authentication and accounting ID sequences.

Unique RADIUS accounting IDs based on source port number

RADIUS ID values are used in request-response matching. For each unique accounting request (including retries, if a response is not received within the configured timeout period), an 8-bit ID value is assigned. The assigned value is freed when the request is no longer pending (when the request has been matched with a response or timed out).

On a high-capacity NAS, such as the MAX TNT, the RADIUS ID space can be used up, so that no unique values are available for the next accounting request. Recognizing this problem, the IETF RADIUS Working Group will propose a change to RFC 2138 (RADIUS), requiring use of the source UDP port in request-response matching. Each request will be identified by the UDP source port as well as the RADIUS ID value.

The proposed change will allow large NASes to have more than 256 outstanding requests. The Ascend RADIUS server supports the use of the source UDP port in request-response matching, as do several other RADIUS server implementations. To configure the MAX TNT to send the

source UDP port number in RADIUS request-to-response matching, use the following parameter (shown with its default setting):

```
EXTERNAL-AUTH
   rad-id-source-unique = system-unique
```

The RAD-ID-Source-Unique parameter controls whether ID values must be unique system-wide or per requesting port. Unique IDs per-system is the default behavior. To include the source UDP port number in response matching, set the parameter to port-unique, as shown in the following example:

```
admin> read external
EXTERNAL-AUTH read

admin> set rad-id-source-unique = port-unique

admin> write
EXTERNAL-AUTH written
```

Note that if the RADIUS server does not distinguish requests on the basis of port and ID values, this configuration has no effect.

RADIUS accounting Start packet includes user's IP address

RADIUS accounting sends a Start packet to a server when a user is authenticated. The RADIUS accounting Start packet has been extended to include the following fields:

Field	Value
Framed-Address	The IP address assigned to the user. This field is included when RADIUS authentication for a scripted user returned a Framed-User attribute.
Framed-Protocol	The encapsulation protocol (PPP/SLIP) in use. This field is included when RADIUS authentication for a scripted user returned a Framed-Protocol attribute.

ATMP attributes in RADIUS accounting Stop records

Previously, RADIUS Stop packets for ATMP-tunneled connections contained the Tunneling-Protocol attribute (127). RADIUS accounting now reports more information about ATMP connections with the addition of three attributes in accounting Stop packets generated by ATMP foreign agents for terminated mobile-client connections. The new attributes are shown in Table 7:

Table 7. ATMP-related attributes in RADIUS Stop packets

Attribute Name	Number	Type
Ascend-Home-Agent-IP-Addr	183	integer
Ascend-Home-Agent-UDP-Port	186	integer
Ascend-Home-Network-Name	185	string

The Ascend-Home-Agent-IP-Addr attribute indicates the IP address of the home agent used for the mobile client whose tunneled connection has terminated.

The Ascend-Home-Agent-UDP-Port attribute indicates the UDP port used for communicating with the home agent for the terminated connection.

The Ascend-Home-Network-Name attribute indicates the name of the mobile client's home network for this mobile client. This attribute is not present in the Stop record if the home agent is configured in ATMP router mode.

For details about these aspects of ATMP configuration, see "Ascend Tunnel Management Protocol (ATMP)" on page 67 of this Release Note.

Following is an example of a Stop record that contains these new attributes:

```
Mon Oct 27 02:41:38 1997
User-Name = "Jacob"
NAS-Identifier = 1.1.1.1
NAS-Port = 10105
Acct-Status-Type = Stop
Acct-Delay-Time = 0
Acct-Session-Id = "111111111"
Acct-Authentic = RADIUS
Acct-Session-Time = 0
Acct-Input-Octets = 215
Acct-Output-Octets = 208
Acct-Input-Packets = 10
Acct-Output-Packets = 10
Ascend-Disconnect-Cause = 1
Ascend-Connect-Progress = 60
Ascend-Data-Rate = 56000
Ascend-PreSession-Time = 1
Ascend-Pre-Input-Octets = 215
Ascend-Pre-Output-Octets = 208
Ascend-Pre-Input-Packets = 10
Ascend-Pre-Output-Packets = 10
Framed-Protocol = PPP
Framed-Address = 2.2.2.2
Tunneling-Protocol = ATMP
Ascend-Home-Agent-IP-Addr = 3.3.3.3
Ascend-Home-Agent-UDP-Port = 5150
Ascend-Home-Network-Name = homenet
```

New Local-Profiles-First setting

The Local-Profiles-First parameter in the External-Auth profile specifies what action to take if the MAX TNT fails its first attempt to authenticate a connection, whether locally or by using an external server. Prior to this release, the parameter could be set to either Yes or No to set the authentication sequence for the TNT. In this release, a new RNAK setting has been added. With this setting, if a NAK is returned from the remote server, the MAX TNT terminates the connection without proceeding to a local profile. Following is a synopsis of the system behavior associated with each setting of the Local-Profiles-First parameter:

When Local-Profiles-First = Yes, the MAX TNT checks the local (NVRAM) profile first.

- If the profile exists and the password matches, it allows the connection.

- If the profile exists and the password does not match, it checks external authentication.
- If the profile does not exist, it proceeds with external authentication.

If Local-Profiles-First = No, the MAX TNT checks external authentication first.

- If the server ACKs the request, it allows the connection.
- If the server doesn't respond, it checks for a matching local profile.
- If the server NAKs the request, it checks for a matching local profile.

If Local-Profiles-First = RNAK, the MAX TNT checks external authentication first.

- If the server ACKs the request, it allows the connection.
- If the server doesn't respond, it checks for a matching local profile.
- If the server NAKs the request, it terminates the connection.

The SNMP sysAuthPreference variable has also been updated to include the new RNAK setting, as shown below:

```
sysAuthPreference   OBJECT-TYPE
                    INTEGER {
                    no-op(1),
                    local-first(2),
                    remote-first(3),
                    remote-first-no-local-if-naked(4)
                    }
ACCESS              read-write
STATUS              mandatory
DESCRIPTION
"An incoming call can be authenticated using a local profile or one from
an authentication server such as RADIUS or TACACS. Local-first means
authenticate from a local profile first, and if that fails, try the
authentication server. Remote-first means get a profile from the authen-
tication server and authenticate from that and if that fails, try to
authenticate from a local profile. Remote-first-no-local-if-naked is
similar to remote-first, except if the external authentication server
NAK the request, than the external authentication server NAK the
request, than the connection will be denied, i.e. no search of the local
profiles will be made."
::= { systemStatusGroup 10 }
```

Configurable cause element in ISDN Disconnect packets

When CLID or DNIS authentication fails, the MAX TNT can now return either User Busy (decimal 17) or Normal Call Clearing (decimal 16) as the Cause Element in ISDN Disconnect packets. The default behavior is to send Normal Call Clearing (16).

The administrator can configure the Cause Element value by setting the following parameters, shown with their default values:

```
EXTERNAL_AUTHENTICATION
   rad-auth-client
      auth-id-fail-return-busy = no
      auth-id-timeout-return-busy = no
```

If these parameters are set to No (the default), Normal Call Clearing (16) is sent when CLID or DNIS authentication fails or times out.

If you set the Auth-ID-Fail-Return-Busy parameter to Yes, User Busy (17) is sent when CLID or DNIS authentication fails.

If you set the Auth-ID-Timeout-Return-Busy parameter to Yes, User Busy (17) is sent when CLID or DNIS authentication times out.

Nailed connections retrieved from RADIUS

The MAX TNT now brings up nailed connections retrieved from a RADIUS server. Previously, when the MAX TNT retrieved a nailed Connection or Frame Relay profile from a RADIUS server, it did not take the action required to bring the connection up. Now, the retrieved profiles are handled as if they were local profiles, and are visible when a user executes a Dir command. For example, in the following Dir command output, the sample profile named "rad" is a nailed connection retrieved from a RADIUS server:

```
admin> dir conn
    44  07/30/1997 14:59:25  don
    67  09/05/1997 13:32:23  Miami
    52  10/30/1997 12:16:22  rad
```

Nailed RADIUS profiles are displayed along with the local profiles, and you can open them. However, the profiles retrieved from RADIUS are read-only in the command-line interface.

Shelf-controller proxy RADIUS accounting

In earlier releases, RADIUS accounting records were handled solely by the host card that performed the packet-handling functions, such as a modem or HDLC card. If the card went down with open sessions, no Stop records were sent to the accounting server for those sessions. (A RADIUS accounting checkpoint feature provides periodic session information to enable session billing even if the RADIUS accounting server didn't receive a Stop record, but this is not considered an ideal solution.)

In this release, the master shelf-controller keeps track of all accounting Start records sent by host cards. If the shelf-controller determines that a host card has gone down for any reason, it acts as proxy for the card and sends the accounting server a fail-safe Stop record for each of the card's open sessions. The host card may be brought down administratively (Slot –d), removed from the system, or it may go down due to an error condition.

How proxy accounting works

When RADIUS accounting is in use, the usual situation occurs as shown in Figure 15:

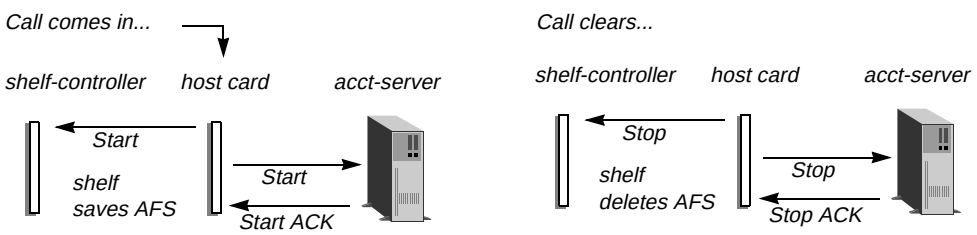

Figure 15. Normal RADIUS accounting (no proxy necessary)

When a call comes in, the host card first sends a Start record to the shelf-controller, which stores it as an Accounting Fail-Safe (AFS) record. The host card then sends one or more Start records to the RADIUS accounting server, repeating until it receives an ACK from the server. Similarly, when the call clears, the host card sends a Stop record to the shelf-controller, which causes it to delete the AFS record for that session. The host card then sends the accounting server Stop records until it receives an ACK from the server.

When RADIUS accounting is in use and the host card goes down for any reason, proxy accounting occurs as shown in Figure 16:

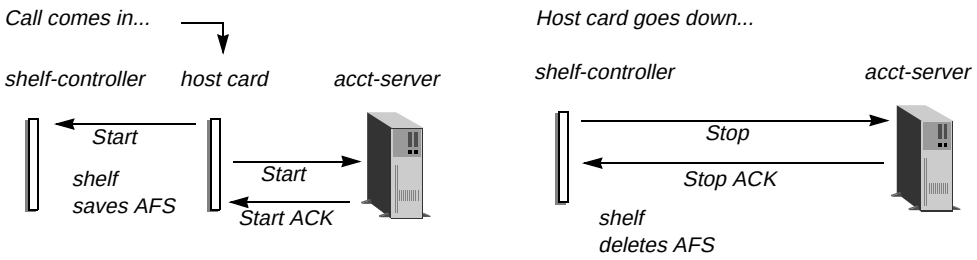

Figure 16. Proxy accounting (host card goes down)

In this case, when the shelf-controller notes that the host card is down, it uses its own information about the host card and the stored AFS record to send a Stop record directly to the RADIUS accounting server, repeating until it receives a Stop ACK from the server. The shelf-controller then deletes the AFS record for that session.

Note that if the accounting server is accessible only via the host card that goes down, Stop records cannot be delivered successfully in any case.

Contents of the Stop record sent by proxy

The AFS Stop record does not contain all of the information it would have if the host card had sent it; in particular it does not contain the input/output octet count fields or any other dynamic information related to the session. In Table 2, Yes means that the attribute is included in the Stop record if applicable. For example, a modem port number is not applicable to a non-modem call. No means that the attribute is either not included in the record or is set to null, as appropriate.

Table 2. Accounting attributes included in proxy Stop records

Attribute in regular Stop record	In proxy Stop record:
Acct-Authentic	Yes
Acct-Delay-Time	Yes
Acct-Input-Octets	No
Acct-Input-Packets	No
Acct-Output-Octets	No
Acct-Output-Packets	No
Acct-Session-Id	Yes

Table 2. Accounting attributes included in proxy Stop records

Attribute in regular Stop record	In proxy Stop record:
Acct-Status-Type	Yes
Acct-Session-Time	Yes. (The session time is accurate to within a few seconds.)
Ascend_Called_Number	No
Ascend-Connect-Progress	Yes
Ascend-Data-Rate	Yes
Ascend-Disconnect-Cause	Yes. (The Disconnect reason is always 210, slot card down.)
Ascend-First-Dest	No
Ascend-Home-Agent-IP-Addr	Yes
Ascend-Home-Agent-UDP-Port	Yes
Ascend_Modem_PortNo	Yes
Ascend_Modem_ShelfNo	Yes
Ascend_Modem_SlotNo	Yes
Ascend-Multilink-ID	Yes
Ascend-Num-In-Multilink	Yes
Ascend-Pre-Input-Octets	No
Ascend-Pre-Input-Packets	No
Ascend-Pre-Output-Octets	No
Ascend-Pre-Output-Packets	No
Ascend-PreSession-Time	Yes
Caller-Id	No
Class	No
Framed-Address	Yes
Framed_IPX_Network	Yes
Framed-Protocol	Yes
Login-Host	Yes
Login-Service	Yes

Table 2. Accounting attributes included in proxy Stop records

Attribute in regular Stop record	In proxy Stop record:
Login-TCP-Port	Yes
NAS-Identifier	Yes
NAS-Port	Yes
NAS-Port-Type	Yes
Tunneling-Protocol	Yes
User-Name	Yes
per-user-rad-acct	Yes. (This information is used to choose an accounting server when a connection is authenticated, and is not part of accounting logging.)

Example of a Stop record sent by proxy

Following is an example of a shelf-controller accounting proxy for an HDLC call:

```
Wed Nov 5 14:50:21 1997
        User-Name = "joel-mhp"
        NAS-Identifier = 206.65.212.199
        NAS-Port = 2272
        NAS-Port-Type = Sync
        Acct-Status-Type = Stop
        Acct-Delay-Time = 0
        Acct-Session-Id = "246212864"
        Acct-Authentic = RADIUS
        Acct-Session-Time = 4
        Acct-Input-Octets = 0
        Acct-Output-Octets = 0
        Acct-Input-Packets = 0
        Acct-Output-Packets = 0
        Ascend-Disconnect-Cause = 210
        Ascend-Connect-Progress = 67
        Ascend-Data-Rate = 0
        Ascend-PreSession-Time = 0
        Ascend-Pre-Input-Octets = 0
        Ascend-Pre-Output-Octets = 0
        Ascend-Pre-Input-Packets = 0
        Ascend-Pre-Output-Packets = 0
        Framed-Protocol = PPP
        Framed-Address = 192.168.6.66
```

Debugging proxy accounting

The Acct-Failsafe debug command is available on the master shelf or the slot host cards for verifying correct accounting proxying. (Slot host cards do not include the "-d" option.) Slave shelf controllers and slot line cards do not support this command. With debug permissions enabled, you can display the usage statement by typing:

```
admin> acct-failsafe
usage: acct-failsafe -option [ params ]
       -d <shelf> <slot>
           (d)isplay AFS info for <shelf> <slot>
       -d (d)isplay AFS info for all relevant slots
       -t (t)oggle module debug level
       -? display this summary
```

To display information about the calls on any slot which are candidates for proxy accounting.:

```
admin> acct-failsafe -d
Slot 1/8:
HashTable @ 10542160, bucketCount: 192, callCount: 23, hashName <afs-1:8>
Slot 2/5:
HashTable @ 10585730, bucketCount: 48, callCount: 7, hashName <afs-2:5>
```

To display the same information for a single slot card in shelf 1, slot 8:

```
admin> acct-failsafe -d 1 8
Slot 1/8:
HashTable @ 10542160, bucketCount: 192, callCount: 23, hashName <afs-1:8>
```

To specify which level of debug to use for the command, use the –t option. A debug level of zero indicates none (no messages). A level of 7 is fairly verbose. For example:

```
admin> acct-failsafe -t 7
acct-failsafe debug output at level 7
```

NAS port identifier optionally reported in new format

By default, the MAX TNT reports the NAS port value in a format that is appropriate to the multishelf and multislot architecture of the system. For details on this default format, see the *MAX TNT RADIUS Configuration Guide*. Now, administrators can specify that the NAS port value must be represented in five digits, as follows:

tllcc

In this format, the following definitions apply:

- *t* = 1 = digital call or 2 = analog call
- *ll* = line number
- *cc* = channel number

For this definition to make sense with the MAX TNT architecture, the following requirements must be met:

- Multishelf is not supported.
- All line cards must be placed contiguously in the first slots of the standalone system. Lines must be numbered 1 through 8 for slot 1, 9 through 16 for slot 2, and so on.

Following is the new parameter that configures the MAX TNT to use the five-digit NAS port value:

```
SYSTEM
   new-nas-port-id-format = yes
```

If set to Yes (the default), the MAX TNT uses the NAS port identifier format that matches its architecture. For details on this format, see the *MAX TNT RADIUS Configuration Guide*. If

New-NAS-Port-ID-Format is set to No, the MAX TNT uses the five-digit NAS port values. Note that the system must be standalone.

The following example configures the MAX TNT to use five-digit NAS port values:

```
admin> read system
SYSTEM read

admin> set new-nas-port-id-format = no

admin> write
SYSTEM written
```

Note: Do not set the New-NAS-Port-ID-Format parameter while the system has active sessions if external accounting is in use.

Enhanced SNMP support

SNMP agent on multishelf system now reports on slave cards

In a multishelf system, the master shelf-controller keeps status information about all slots in the system. The SNMP agent on the master shelf-controller reports status information on the slots in the Ascend Enterprise MIB Slots group. The slotIndex for the cards in each shelf in a multishelf system is shown below:

Multishelf slots	slotIndex value
Shelf 1 Slots 1 – 18	1 – 18
Shelf 2 Slots 1 – 18	19 – 36
Shelf 3 Slots 1 – 18	37 – 54
Shelf 4 Slots 1 – 18	55 – 72
Shelf 5 Slots 1 – 18	73 – 90
Shelf 6 Slots 1 – 18	91 – 108
Shelf 7 Slots 1 – 18	109 – 126
Shelf 8 Slots 1 – 18	127 – 144
Shelf 9 Slots 1 – 18	145 – 162

Slots 1-16 represent the actual removable slot cards. Slots 17 represents the shelf controller. Slot 18 is reserved for future use.

For example, for a multishelf MAX TNT with master shelf 4 and slave shelves 3 and 7, the slotIndex range would be 37-54 for slave shelf 3, 55-72 for the master shelf, and 109-126 for slave shelf 7.

Ability to disconnect user via SNMP request

An SNMP Set request can now terminate a user session by setting one of the following SNMP objects in the session MIB to invalid(1):

`ssnStatusValidFlag` as part of `sessionStatusTable`

`ssnActiveValidFlag` as part of `sessionActiveTable`

Additional information about SNMP-initiated transfers

The MAX TNT supports the following MIB variable, which is used to ascertain the status of a TFTP download or upload initiated via SNMP:

```
ascend.systemStatusGroup.sysConfigTftp.sysConfigTftpStatus
```

In the past, the reported status was limited to Passed or Failed. It can now be used to determine the following states:

```
ok( 1 ),                    -- tftp operation succeeded
notFound( 2 ),              -- file not found
access( 3 ),                -- access violation
noSpace( 4 ),               -- no disk space to write file
badTid( 6 ),                -- unknown transfer ID
badOp( 5 ),                 -- bad tftp operation
exists( 7 ),                -- file already exists
noSuchUser( 8 ),            -- no such user
parameter( 9 ),             -- parameter error
busy( 10 ),                 -- tftp server cannot handle request
noResources( 11 ),          -- no memory for request
timeout( 12 ),              -- timed out
unrecoverable( 13 ),        -- unrecoverable error
tooManyRetries( 14 ),       -- too many retries
createFile( 15 ),           -- create file
openFile( 16 ),             -- open file
inProgress( 17 )            -- get/put request in progress
```

Ability to enable and disable modems via SNMP

An SNMP Set request can now enable or disable a modem by setting the following SNMP object in the slots group to invalid(1):

```
ascend.slots.slotMdmTable.slotMdmEntry.slotMdmItemConfig
```

SNMP support for TNT IDSL slot card

Line status, line traps, and user connection statistics for the IDSL slot card are now retrievable via SNMP.

SNMP advanced.mib now supported

This release supports the Ascend Advanced MIB, previously called the WAN MIB.The Advanced MIB defines objects related to WAN lines, channels, and ports.

DTPT sessions to the ZGR identified in Session MIB

In this release, outgoing DTPT sessions to the ZGR are identified in the sessionStatusTable and sessionActiveTable as part of session.mib. To use this feature, administrators must compile the new session.mib file into their management stations.

The following variables have been added to the two tables in the session.mib to identify DTPT sessions to the ZGR:

- ssnStatusCurrentService

- ssnActiveCurrentService

These variables support the following new values:

- virtualConnect(16),-- Virtual Connect to a modem
- dchannelX25(17), -- D Channel X.25
- dtpt(18) -- DTPT session to ZGR.

Multishelf traps enabled by default

In previous releases, the value of the SNMP MIB object multiShelfStateTrapState (multiShelf 6) was set to Disabled by default. Now, it is set to Enabled by default, as shown in the following object definition:

```
multiShelfStateTrapState OBJECT-TYPE
SYNTAX  INTEGER { enabled(1), disabled(2) }
ACCESS   read-write
STATUS   mandatory
DESCRIPTION
"This variable indicates whether the master system
produces the multiShelfStateChange trap."
DEFVAL { enabled }
::= { multiShelf 6 }
```

This object determines whether a trap is generated when a multishelf link is down (if one of the shelves is down.) If it is set to Disabled (2), the trap is not sent, regardless of Trap profile configurations. If it is set to Enabled (the default), the Slot-Enabled parameter in a Trap profile determines whether the specified host receives the multishelf trap. Only if Slot-Enabled is set to Yes in the Trap profile does the specified host receives multishelf traps.

In the following example, Host-A (10.2.3.4) receives multishelf traps and Host-B (10.5.6.7) does not:

```
admin> new trap host-a
TRAP/host-a read

admin> list
host-name* = test
community-name = ""
host-address = 0.0.0.0
alarm-enabled = yes
security-enabled = no
port-enabled = no
slot-enabled = no

admin> set host-address = 10.2.3.4

admin> set slot-enabled = yes

admin> write
TRAP/host-a written

admin> new trap host-b
TRAP/host-b read

admin> set host-address = 10.5.6.7

admin> write
TRAP/host-b written
```

If the administrator sets the multiShelf.multiShelfStatTrapState object to 2 (disabled), neither host receives multishelf traps.

Slave shelves generate trap when multishelf link is down

In this release, both the master shelf and slave shelf can forward SNMP traps if the multishelf link between them is down. If the link is down because the master shelf is powered down or reset, the slave shelf forwards a trap. If the link is down because the multishelf cables are disconnected, both the master and slave shelves can forward a trap.

If traps are enabled on both the master and slave shelf controllers, a trap with the following OID may be generated to indicate multishelf link conditions:

`.1.3.6.1.4.1.529.19.5.1.2.`*X*

A trap is reported by both the master and slave shelf-controllers when a multishelf cable between the master and slave is disconnected. In this case, *X* in the OID is the number of the shelf that lost communication, and the trap value is 1 (idle).

A trap is reported by either the master or slave shelf-controller if one of them is reset.In this case, *X* in the OID is the number of the shelf that lost communication, and the trap value is 1 (idle).

A trap is reported by the master shelf-controller when the link is back up again. In this case, *X* in the OID is the destination shelf number, and the trap value is 4 (up). This trap is reported only by the master shelf to indicate that the entire multishelf system is up.

Customized features: T-Online

This section describes features that apply only to the Deutsche Telekom T-Online service. T-Online support was introduced in release 1.3A. Throughout this section, DIRDO stands for Dial-In Redirect Dial-Out.

T-Online: PRI-PRI switching for E1

PRI-PRI switching for T-Online provides a network side implementation of NET-5 to support switching calls from the Deutsche Telekom public network to a T-Online server. If T-Online is enabled in the System profile, the MAX TNT compares the phone number and subaddress number it obtains from the call Setup and Info messages to the DIRDO info stored in RADIUS. It switches the inbound call to the T-Online server if it finds any of the following matches in RADIUS:

- The phone number and subaddress of the incoming call match a phone number and subaddress entry in RADIUS.
- The phone number matches a phone number entry in RADIUS and there is no subaddress.
- The subaddress matches a subaddress entry in RADIUS and there is no phone number.
- There is no incoming-call phone number or subaddress.

The MAX TNT begins collecting the subaddress information, and for each call Setup message from the switch that does *not* include “Sending Complete Information Element,” it starts the T302 timer (the Setup Ack timer).

The MAX TNT stops the timer when it receives a message that includes “Sending Complete Information Element.” The MAX TNT assumes there are no more subaddress digits to collect when the T302 timer stops or expires.

Setting up PRI-PRI switching in the System profile

To support PRI-PRI switching for T-Online, the following parameters have been added to the System profile (shown with their default values):

```
SYSTEM
   t-online = no
   t-online-most-avail-chan = no
```

To set up PRI-PRI switching for T-Online, you enable T-Online and specify how the system chooses which NT-configured line to choose for redirecting the call.

The T-Online parameter enables the MAX TNT to route calls to a T-Online server.

Note: Trunk group 8 is reserved for DTPT calls when the T-Online parameter is set to Yes.

The T-Online-Most-Avail-Chan parameter specifies the channel allocation algorithm. Set it to Yes to choose the link with the most available channels. Set it to No to choose a link by the round-robin method.

The commands in the following example enable T-Online and specify that the system should use the link with the most available channels:

```
admin> read system
SYSTEM read

admin> set t-online = yes

admin> set t-online-most-avail-chan = yes

admin> write
SYSTEM written
```

Configuring the E1 lines

To enable PRI-PRI switching on the E1 lines, the following parameters have been added to the E1 line profile (shown with their default values):

```
E1 {shelf-N slot-N N}
   line-interface
      t-online-type = none
      t302-timer = 1500
```

The T-Online-Type parameter is set to None by default. For PRI-PRI switching, you can set it to TE or NT. If you set it to TE, the E1 line should connect to the switch. If you set it to NT, the line should connect to the ZGR server. One TE-configured line can switch calls to one or more NT-configured lines.

The T302-Timer parameter sets the number of milliseconds the MAX TNT waits before assuming that there are no more subaddress digits to collect. You can specify a value range from 100 to 30000 (.1 sec to 30 sec).

Following is an example of a procedure that configures both the TE and NT lines:

```
admin> read e1 {1 16 7}
E1/{ shelf-1 slot-16 7 } read
```

```
admin> set t-online-type = te
admin> set t302-timer = 3132
admin> write
E1/{ shelf-1 slot-16 7 } written
admin> read e1 {1 16 8 }
E1/{ shelf-1 slot-16 8 } read
admin> set t-online-type = nt
admin> set t302-timer = 3132
admin> write
E1/{ shelf-1 slot-16 8 } written
```

With this configuration, when both lines are up, the Line status window will display TE for line 7 and NT for line 8.The window displays information about the channels for these lines, based on the call type, as follows:

Normal calls display the standard characters that indicate call status:

Character	Meaning
d	dialing
* (asterisk)	connected call
r	ringing

PRI-PRI calls display the following characters to indicate call status:

Character	Meaning
N	dialing
= (equal-sign)	connected call
R	ringing

DTPT outgoing calls display the following characters to indicate call status:

Character	Meaning
D	dialing
% (percent-sign)	connected call

Pseudo-tunneling PPP for T-Online

The Deutsche Telekom T-Online service enables individuals with single-channel PPP connections to contact a ZGR. The ZGR is itself highly restrictive, requiring a one-to-one correspondence of packets' IP source address to the B channel on which the packets are received. A call to the ZGR via the MAX TNT has the same interface-to-address restriction as a call made directly to the ZGR.

So that the MAX TNT can handle PPP connections in the usual way, and yet comply with the ZGR restrictions, it makes use of a new DTPT encapsulation protocol for connections to a ZGR. WAN traffic that is not routed to a ZGR undergoes no special handling. PPP

authentication, bringing up and tearing down sessions, security, and the IP routing engine all operate normally.

WAN -to-WAN routing to the ZGR

In the usual case, a PPP connection comes in and the MAX TNT authenticates it as usual, builds a session, and then routes the data out over the WAN to the ZGR via a single-channel dedicated call, as shown in Figure 17. Each B-channel connection to the ZGR is associated with one source IP address.

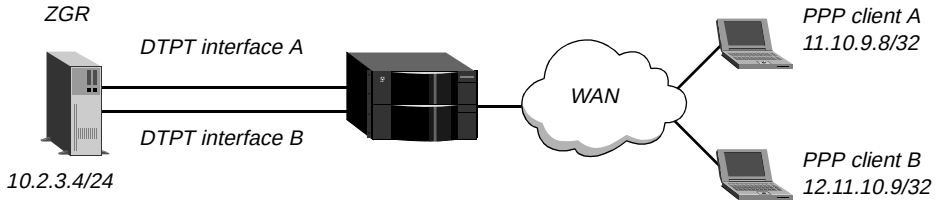

Figure 17. Dial-in PPP connections connecting to a ZGR

IP packets received from a PPP client and destined for the ZGR must all have the same source address. If the dial-in equipment is a router, the first IP address that generates a packet destined for the ZGR owns the resulting ZGR connection, and no other IP addresses behind the same router can connect to the ZGR.

The dial-in client must use PPP encapsulation or one of its multi-channel variants for the initial connection. The actual connection to the T-Online ZGR server is limited to a single B channel, but the user can access the ZGR server and other destinations in the same session.

A call from the MAX TNT to the ZGR can be disconnected for any of the reasons applicable to an ordinary PPP connection, including termination by the ZGR. If the connection to the end-user terminates for any reason, the MAX TNT immediately terminates the associated call to the ZGR.

Local-to-WAN routing to the ZGR

If the packet bringing up the call is originated by the MAX TNT itself, its associated source interface is the loopback interface. Call handling is similar to the WAN-to-WAN routing case, except that, because there is no inbound call, the outbound call is not actively torn down when the source traffic stops. Instead, it behaves like an ordinary outbound PPP call in that it will eventually time out, assuming it has not been terminated by the remote end.

LAN-to-WAN routing to the ZGR

IP packets from the LAN can also be routed to the ZGR via the MAX TNT, as shown in Figure 18:

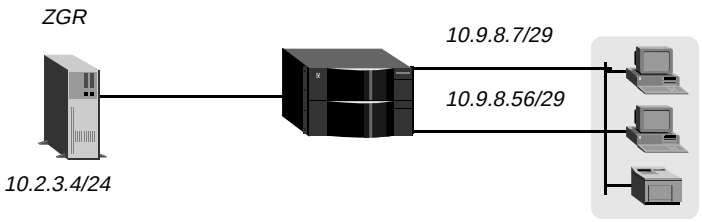

Figure 18. Local clients connecting to a ZGR

Packets from many source addresses might arrive over a single LAN interface. However, there is still a one-to-one correspondence between source IP addresses and outgoing channels to the ZGR. Because the number of source addresses on the LAN can vary, up to a number larger than the number of interfaces on a MAX TNT, it is impossible to guarantee that there will be a B channel available for every attempted access to the ZGR. If LAN-initiated calls to the ZGR are allowed, access might fail for LAN users, WAN users, or both.

Calls placed in response to packets received over the LAN do not automatically terminate when the source of the packets stops sending them, unless you set a value for the Idle-Timer parameter in the Connection Session-Options subprofile. If you set the Idle-Timer parameter, some users might obtain a connection to the ZGR on an initial attempt, but not on a reconnect attempt that occurs after the timer has expired.

Because the LAN clients do not require authentication for sending packets to the MAX TNT, you should ensure security on the LAN by some other means.

Configuring a connection to a ZGR

The following parameters in a Connection profile enable pseudo-tunneled PPP to the ZGR server. The parameters are shown with sample values.

```
CONNECTION station
   encapsulation-protocol = dtpt

   ppp-options
      send-auth-mode = pap-ppp-auth
      send-password = zgr-password
      link-compression = stac
      MRU = 1524

   ip-options
      vj-header-prediction = yes
      remote-address = 10.2.3.4/24
```

These parameters are not new for the DTPT encapsulation type. With the exception of the Encapsulation-Protocol parameter, they do not support new settings for DTPT. However, they are required for optional settings that apply to the DTPT connection to the ZGR.

DTPT encapsulation sets up the call management required for managing multiple discrete B-channel connections to the same destination (the ZGR server). Although in some respects the set of calls to the ZGR resembles an MP or MP+ bundle, they are separate PPP calls, and no bundling or dynamic bandwidth allocation applies.

DTPT encapsulation differs from PPP in that when the MAX TNT uses DTPT to call the remote end, it reports its IP address during NCP negotiations as the IP source address of the packet that caused it to place the call, instead of reporting its own (router) address.

The parameters in Table 8 are not applicable when the DTPT encapsulation type is specified.

Table 8. Parameters that are not applicable for DTPT-encapsulated calls

Parameter that does not apply	Explanation
Telco-Options Answer-Originate	Calls to the ZGR are always outbound.
IP-Options Private-Route	Because the ZGR server has a single IP address, the routing table contains a single route to the ZGR, regardless of how many calls are active. That route is never advertised.
IP-Options RIP	The route to the ZGR is never advertised.

Following is a sample Connection profile for connecting to a ZGR server:

```
admin> new connection zgr-1
CONNECTION/zgr-1 read

admin> set encaps = dtpt

admin> set ppp send-auth-mode = pap-ppp-auth

admin> set ppp send-password = zgr-password

admin> set ip remote-address = 10.2.3.4/24

admin> write
CONNECTION/zgr-1 read
```

Note that PAP authentication is recommended for connecting to the ZGR. If the MAX TNT places calls to the ZGR without proper authentication, the ZGR connection is subject to denial-of-service attacks.

Interface and route handing for DTPT interfaces

For DTPT, the MAX TNT creates a pseudo-interface and aims the routes to any destination with the encapsulation type DTPT at that interface. The pseudo-device name for the DTPT pseudo-interface is dtpt*nnn* where *nnn* is the number of the interface in the interface table. WAN traffic that is not routed to a DTPT interface is not handled any differently than usual.

When the MAX TNT receives a packet whose destination has a Connection profile specifying DTPT encapsulation, it routes the packet (by conventional operation of the routing engine) to the DTPT pseudo-interface, where the source address is examined. The MAX TNT then finds the corresponding output session (with a real outbound interface) and forwards the packet on that interface. No DTPT session is ever used to carry outbound packets from more than one source address.

If the MAX TNT receives a packet for which there is not already a connection to the ZGR, it creates a new dialout interface, associates it with the inbound call, and places the call. The MAX TNT deactivates the interface if the attempt to open the new session fails or if the interface fails a validation test. Because the interface is not permanent, deactivating it removes it from the table.

Introduction

What is in this addendum

The documentation that came with the MAX TNT unit describes how to install the hardware and configure the system. However, since the documentation was published, new system software has been released that contains features that are not yet included in the product documentation. This addendum describes those new features.

Related publications

Additional information is available in the MAX TNT documentation set, which consists of the following manuals:

- *The Ascend Command-Line Interface*. Shows you how to use the MAX TNT command-line interface effectively.
- *MAX TNT Hardware Installation Guide*. Describes how to install the MAX TNT hardware and use the command-line interface to configure its slot cards for a variety of supported uses. Describes how calls are routed through the system. Includes the MAX TNT technical specifications.
- *MAX TNT Network Configuration Guide*. Describes how to use the command-line interface to configure WAN connections and other related features.
- *MAX TNT RADIUS Configuration Guide.* Describes how to use RADIUS to configure WAN connections and other related features.
- *MAX TNT Reference Guide*. An alphabetic reference to all MAX TNT profiles, parameters, and commands.

New features in release 2.0.0

This section describes all features introduced in 2.0.0.

Rockwell code version

Release 2.0.0 supports Rockwell K56flex code version 1.160T.

Modified profiles

This section describes parameters that have been moved or deleted from the profiles in which they resided in previous releases.

Ethernet profile

The read-only MAC-Address and Link-State parameters have been moved from the Ethernet profile to the Ether-Info profile, which is not written to NVRAM. Ether-Info profiles are created when the card is in active state, and are deleted when the slot is brought down.

Following is an example that lists the contents of an Ether-Info profile:

```
admin> dir ether-info
     0  08/13/1997 13:34:26  { shelf-1 slot-2 1 }
     0  08/13/1997 13:34:26  { shelf-1 slot-2 2 }
     0  08/13/1997 13:34:26  { shelf-1 slot-2 3 }
     0  08/13/1997 13:34:26  { shelf-1 slot-2 4 }
     0  08/13/1997 13:34:30  { shelf-1 controller 1 }

admin> read ether-info { 1 2 1}
ETHER-INFO/{ shelf-1 slot-2 1 } read

admin> list
interface-address* = { shelf-1 slot-2 1 }
mac-address = 00:c0:7b:68:ef:98
link-state = up
```

Note: The MAC-Address and Link-State parameters no longer appear in the Ethernet profile. Because these parameters are read-only, there are no backward compatibility issues.

Frame-Relay profile

The FR-Link-Up parameter has been deleted from Frame-Relay profiles, and the corresponding RADIUS attributes Ascend-FR-LinkUp (157) and supporting values Ascend-LinkUp-Default and Ascend-LinkUp-AlwaysUp) are deprecated. Attribute 157 will be ignored in the current release, and may be reused for other purposes in a future release.

Administrators are encouraged to remove Ascend-FR-LinkUp (157) from their users file as soon as possible to avoid conflicting with other uses in the future.

New PCMCIA flash format

In releases earlier than 2.0.0, the system had to be running new shelf-controller code before you could successfully load a Tar file that contained images for new cards that were not supported in the earlier software version. To remove this requirement and facilitate future upgrades, the PCMCIA flash card format has been modified to store loads by a card type's hardware identifier.

Note: The new flash format is incompatible with the format used previously, so flash cards must be reformatted when upgrading to 2.0.0. This means that before reformatting, you must save your configured profiles to an external location.

The most recent upgrade instructions are available on the Ascend FTP server.

New dual-stage boot procedure

In previous releases, the shelf-controller code image resided in on-board flash and slot-card code resided in PCMCIA flash. In this release, both code images reside on the PCMCIA card, and a boot-loader program resides in on-board flash memory. For details about how to upgrade the system to versions including the boot-loader software, see the most recent upgrade instructions, which are available on the Ascend FTP server.

.How the boot-loader operates

The sole purpose of the boot-loader program is to load the operational code images. When the system comes up, the boot-loader initializes the shelf-controller by running POST, invoking a command-line interface on the local serial port, and enabling IP networking on the shelf-controller's built-in Ethernet port. At this point, it the boot-loader reads the System profile to determine whether it is running on a master (or standalone) shelf, or on a slave shelf.

If it is running on a master or standalone shelf, the boot-loader attempts to load the operational code for the shelf-controller by reading from a local flash card. If the attempt succeeds, it executes the operational code and brings up all slot cards. The master shelf then satisfies all code requests from slot cards and slave shelves via the Image Transfer Protocol (ITP).

If the boot-loader is running on a slave shelf, it requests the operational code for the shelf-controller from the master shelf via ITP. (If the master shelf does not have the code at that point, the ITP transfer fails and the slave does not retry until it is reset.)

What happens if the dual-stage boot fails

Generally, the dual-stage boot works so that the administrator might not even be aware that it takes place. However, if the boot procedure fails for any reason—for example, if the flash card containing the code has been removed or is corrupted, or if the master shelf is not available—the system comes up in the boot-loader state. In the boot-loader state, the system prompt is:

```
BOOT>
```

The command-line interface in this state is accessible from the shelf-controller's onboard Ethernet and Serial ports. It supports a subset of commands and profiles, so standard command such as Show and Status are not available. The following profiles are used by the boot-loader to configure the remote management communication:

- System profile
- IP-Global profile
- IP-Interface profile
- Log profile
- Serial profile

Note: Some of the parameters contained in these profiles may not be visible in the boot-loader state. Although it is possible to edit the profiles while the system is in the boot-loader state, this is not recommended. Settings for parameters supported in the operational but not the boot-loader environment would be lost if you write the profile.

In addition, in the boot-loader state, slot cards neither boot nor operate. If the failure occurred on the master of a multishelf system, all cards in the system are in the RESET state. If the failure occurred a slave shelf, only the cards on that shelf are in the RESET state.

In the boot-loader state, the following functionality is still available to the administrator:

- Coredump
- DNS
- IP routing (no OSPF)
- Syslog
- SNTP
- Fatal error log
- Telnet (inbound and outbound)
- Ping (inbound and outbound)
- TFTP and serial code load

Troubleshooting the dual-stage boot procedure

If the dual-stage boot procedure fails and the system is in the boot-loader state, first try to manually initiate the second stage of the boot procedure by resetting the system with the following command:

```
BOOT> reset
```

Note: The boot-loader does not support the –a option to the Reset command. On a multishelf system, you must manually reset each shelf, beginning with the master shelf. If the problem that caused the boot failure is corrected by resetting the master shelf, the slave shelves should not require any further action—they will come up when the master shelf is operational.

If this fails to resolve the difficulty and the system is a master shelf (or a standalone system), verify that the flash card containing the code load has not been removed from the system. If the system is a slave shelf, verify that the master shelf is available by checking the status of the master shelf and the intershelf cabling. If you have ruled out these possibilities, follow this procedure:

1 Use the Dircode command to verify that the flash card is present and contains shelf-controller operational code. For example, the Dircode output should include a line such as this:

```
BOOT> dircode
Flash card code directory:
```

```
Card 1, directory size 16
          4ether-card reg   good   141521 Jan 6 18:49      2.0.0
          32idsl-card reg   good   511317 Jan 6 18:51      2.0.0
              t3-card reg   good   205111 Jan 6 18:49      2.0.0
         48modem-card reg   good   547614 Jan 6 18:49      2.0.0
         capadsl-card reg   good   409774 Jan 6 18:50      2.0.0
           4swan-card reg   good   366848 Jan 6 18:50      2.0.0
    10-unchan-t1-card reg   good   459811 Jan 6 18:50      2.0.0
             8t1-card reg   good   177351 Jan 6 18:49      2.0.0
     shelf-controller reg   good   940282 Jan 6 18:51      2.0.0
         192hdlc-card reg   good   543436 Jan 6 18:49      2.0.0
     48modem-56k-card reg   good   556918 Jan 6 18:50      2.0.0
            sdsl-card reg   good   382271 Jan 6 18:50      2.0.0
```

2 If the shelf-controller image is *not* listed in the Dircode output, reload the Tar file and then reset. For example:

```
BOOT> load tar network host1 /vol/src/tntrel.tar

BOOT> reset
```

3 If shelf-controller code *is* listed in the Dircode output, use the Fsck command to check the flash file system. For example:

```
BOOT> fsck 1
ffs check in progress for card 1...

Dir 1 has magic, size 16, sequence 0x1d5
Dir 2 not in use

Using dir entry: 1, total data blocks: 0x40, directory size: 16

4ether-card:(0x08)
    reg   good 141521 (0x0228d1) Jan 6 18:49
32idsl-card:(0x14)
    reg   good   511317 (0x07cd55) Jan 6 18:51
t3-card:(0x0d)
    reg   good   205111 (0x032137) Jan 6 18:49
48modem-card:(0x06)
    reg   good   547614 (0x085b1e) Jan 6 18:49
capadsl-card:(0x11)
    reg   good   409774 (0x0640ae) Jan 6 18:50
4swan-card:(0x09)
    reg   good   366848 (0x059900) Jan 6 18:50
10-unchan-t1-card:(0x0b)
    reg   good   459811 (0x070423) Jan 6 18:50
8t1-card:(0x04)
    reg   good   177351 (0x02b4c7) Jan 6 18:49
shelf-controller:(0x02)
    reg   good   940282 (0x0e58fa) Jan 6 18:51
192hdlc-card:(0x07)
    reg   good   543436 (0x084acc) Jan 6 18:49
48modem-56k-card:(0x0e)
    reg   good   556918 (0x087f76) Jan 6 18:50
sdsl-card:(0x10)
    reg   good   382271 (0x05d53f) Jan 6 18:50
flash card 1 fsck: good.
```

4 If errors are reported in the Fsck output, use the Format command to reformat the flash card, and then reset the system. For example:

```
BOOT> format 1
format will erase existing card 1 data; confirm: [y/n] y
format in progress...
format complete.

BOOT> reset
```

5 When the system has reset, Telnet back into the system and load the code again. For example:

```
BOOT> load tar network host1 /vol/src/tntrel.tar
```

6 Reset again.

```
BOOT> reset
```

Multishelf features

Loadslave command for updating slave shelf code

The Loadslave command enables the administrator to update slave shelves from the master shelf interface. It uses the following syntax:

```
admin> help loadslave
LoadSlave usage: LoadSlave shelf [image (1 or 2)]

- shelf: The Slave Shelf Controller to be loaded.
- image: The low (1) or high (2) boot image of the Master.
default is image2
```

The first argument is the shelf number of a slave shelf. The second argument specifies which of two load images to use to update the specified shelf. Both load images are maintained in the master shelf-controller's NVRAM. For example, the following command updates slave shelf 3 with the code image in the master shelf-controller's high-address section of NVRAM (the default):

```
admin> loadslave 3
```

When you load a binary to the master shelf-controller via TFTP or a serial connection, the compressed image is stored in the high-address section of NVRAM, referred to as *image2* in Figure 1. When you then reset the system to execute the new shelf-controller software, the system first verifies that the compressed image is good and copies it into the low-address section of memory. The copy is referred to as *image1*. The system then decompresses image1, loads it into memory, and boots from image1.

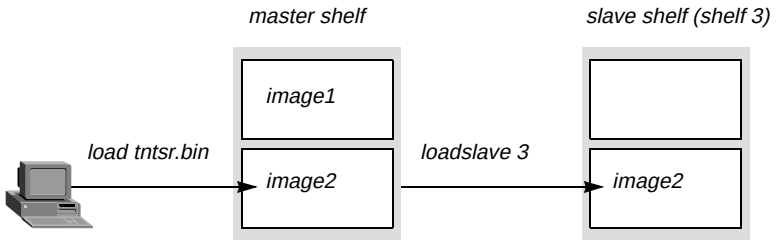

Figure 1. Loading new shelf-controller software

The slave shelf always stores the code image in the high-address section of its NVRAM (image2). However, you can specify in the Loadslave command whether you want it to load the

binary from image1 or image2 in the master shelf. The default is image2. After you reset the master shelf, both images are identical.

Reset –a to reset the mutishelf system

To reset the master shelf and all slaves in a multishelf system, append the –a flag to the Reset command. For example, while logged into the master shelf, the following command resets the entire multishelf system:

```
admin> reset –a
```

The system prompts for confirmation:

```
Reboot the entire system, dropping all connections? [y/n] y

Please stand by. System reset in progress...
```

The –a flag is not valid on slave shelves.

MP and MP+ bundled channels span HDLC cards and shelves

The MAX TNT can now bundle channels for an MP or MP+ connection across multiple HDLC cards, which may reside in different shelves of a multishelf system. The behavior of the Call-Routing-Sort-Method parameter in the System profile has been modified to enable bundling channels across HDLC cards transparently. In addition, a new TNTMP –i command enables the administrator with debug permissions to check bundles.

Changes to the Call-Routing-Sort-Method

When the system resets, the MAX TNT creates its call-routing database by sorting the list of all installed devices. (During active use, the sort order depends on system activity, but the initial sort determines the order in which the MAX TNT first uses host channels.)

If Call-Routing-Sort-Method is set to Item-First in the System profile, as shown in the following example, calls are supposed to be distributed across multiple host cards:

```
admin> read system
SYSTEM read

admin> set call-routing-sort-method = item-first

admin> write
SYSTEM written
```

However, previously, calls were not distributed evenly on multishelf systems. For example, if a multishelf system had an HDLC card in slots 1/15 and 9/2, the system created the following call-routing database after a reset:

```
admin> callroute -a
 1:15:01/1   0 0:00:00/0 digital-call-type 0 0
 1:15:01/2   0 0:00:00/0 digital-call-type 0 0
 1:15:01/3   0 0:00:00/0 digital-call-type 0 0
 ...
 1:15:01/32 0 0:00:00/0 digital-call-type 0 0
 ...
 9:02:01/1   0 0:00:00/0 digital-call-type 0 0
 9:02:01/2   0 0:00:00/0 digital-call-type 0 0
```

```
 9:02:01/3   0 0:00:00/0 digital-call-type 0 0
 ...
 9:02:01/32 0 0:00:00/0 digital-call-type 0 0
```

In this case, the first 32 calls were routed to 1/15, the next 32 calls were routed to 9/2, the next 32 calls were routed to 1/15, and so forth. In this release, if Call-Routing-Sort-Method is set to Item-First in the System profile and the system has an HDLC card in slots 1/15 and 9/2, the system creates the following call-routing database after a reset:

```
admin> callroute -a
 1:15:01/1   0 0:00:00/0 digital-call-type 0 0
 9:02:01/1   0 0:00:00/0 digital-call-type 0 0
 1:15:01/2   0 0:00:00/0 digital-call-type 0 0
 9:02:01/2   0 0:00:00/0 digital-call-type 0 0
 1:15:01/3   0 0:00:00/0 digital-call-type 0 0
 9:02:01/3   0 0:00:00/0 digital-call-type 0 0
 ...
 1:15:01/32 0 0:00:00/0 digital-call-type 0 0
 9:02:01/32 0 0:00:00/0 digital-call-type 0 0
```

The new order distributes the calls evenly across the two HDLC cards in different shelves.

The TNTMP –i command

The following example requires that debug permissions be enabled for the current User profile. It shows how the TNTMP –i command displays information about MP and MP+ bundles and their channels:

```
admin> tntmp -i
mpBundleID=13 masterSlot=1/15 masterMpID=2 ifCount=2 rtIf=1/17:6
           routeID     slot    ifNum localIfNum localMpID
                32     1/15        1          1          2
                33     9/ 2      193          1          2
```

This command works on HDLC cards as well, as shown in the following example:

```
admin> open 1 15

hdlc-1/15> tntmp -i
mpBundleID=13 masterSlot=1/15 masterMpID=2 ifCount=2 rtIf=1/17:6
           routeID     slot    ifNum localIfNum localMpID
                32     1/15        1          1          2
                33     9/ 2      193          1          2
```

In both examples, the output shows a two-channel MP or MP+ bundle with the first channel in slot 1/15 and the second (slave) channel in slot 9/2. The fields in TNTMP –i command output are explained below.

Field	Description
`mpBundleID`	The bundle ID known to the whole system. If the connection adds channels for additional bandwidth on demand, the call for those channels is compared to the current bundle and assigned the same bundle ID as the other channels of the call.

Field	Description
masterSlot	The master slot shows the channel that was established as the base channel of the connection. When the MAX TNT receives a call that is not part of an existing bundle, it authenticates the call and establishes the base channels of the connection. That channel becomes the master of the multilink connection.
masterMpID	The MpID is the bundle ID known locally to the slot card. The masterMpID field shows the MpID at the master slot card. (The masterMpID is always the same as the localMpID for channels on the master slot card.)
ifCount	The interface count shows the number of channels in the bundle.
rtIf	The rtIf field shows the shelf/slot:ID for the Route Logical Interface.
routeID	The routeID column shows the globally known ID for each call.
slot	The slot column shows the shelf/slot numbers of the channels in the MP or MP+ bundle.
localIfNum	The localIfNum is the channel number on the slot card. For HDLC cards, the channels are numbered 1–128. In the sample output, the master slot (1/15) shows channel number 1. The interface number for the slave slot (9/2) is also 1, meaning the first channel on that card. However, at the master slot card, the slave interface number is mapped to a pseudo-interface number greater than 128, to prevent its being confused with channels on the master slot.
localMpID	The MpID is the bundle ID known locally to the slot card. The localMpID field shows the MpID for channels on the local slot card.

Multishelf Show and Open commands

Changes have been made in shelf-to-shelf communications that affect how master and slave shelves communicate. In particular, the Open command now works for slot 17 (the controller) on slave shelves, and the Show command displays information about slot 17 on slave shelves.

Using the Show command

On the master shelf, the Show command output now includes slave shelf-controllers that are UP. For example:

```
Shelf 1 ( master ):
    { shelf-1 slot-1 0 }       UP       8t1-card
    { shelf-1 slot-4 0 }       UP       128hdlc-card
    { shelf-3 slot-1 0 }       UP       128hdlc-card
    { shelf-3 slot-2 0 }       UP       4ether-card
    { shelf-3 slot-3 0 }       UP       8t1-card
    { shelf-3 slot-4 0 }       UP       48modem-56k-card
    { shelf-3 slot-5 }         OCCUPIED
    { shelf-3 controller 0 }   UP       shelf-controller
```

You cannot change the state of a slave shelf-controller by using the Slot –u or Slot –d commands. If you do execute attempt to bring the slave shelf up or down by using one of these commands, the following error message appears:

```
can’t force slot 3/17 state change
```

Using the Open command

On the master shelf of a multishelf system, you can open a session with a slave shelf (for example, shelf 3) as follows:

```
admin> open 3 17
```

You can then execute commands on the slave shelf as usual, except that you cannot use the Open command from the slave shelf. If you do execute the Open command, the following error appears:

```
Can’t use open command on a slave shelf.
```

Slave shelves log reset to master’s fatal log

Slave shelves now log an Operator Reset message both to the master shelf fatal error log and to their own log. The message uses the following format:

```
OPERATOR RESET: Index: 99 Revision: 2.0.0 Shelf 9 (tntsr)
Date: 11/07/1997.       Time: 17:56:06
Reset from unknown, user profile super.
```

The shelf number indicates which slave shelf was reset. In addition, if the shelf is reset locally (from a Telnet or console session) the following message is displayed in that session:

```
Please stand by. System reset in progress...
```

If the slave shelf is reset indirectly from the master shelf, the message appears in all debug-enabled sessions.

Features for T1, E1, and T3 cards

R2 signaling support for E1

R2 signaling is an ITU-T standardized signaling protocol, which can be used on E1 digital trunks for establishing and clearing 64Kbps switched circuits. It uses a combination of A/B bit manipulation in channel 16 of the E1 frame (line signaling), and in-band MF tone generation and detection (register signaling). The relevant specifications are in ITU-T recommendations Q.400 to Q.490.

R2 signaling is widely implemented in international markets where ISDN PRI is not yet available. Ascend supports this protocol on the MAX TNT E1 platform. The following parameters in an E1 profile, shown with sample values, are relevant to R2 signaling:

```
E1 {shelf-N slot-N N}
   line-interface
      signaling-mode = e1-r2-signaling
      switch-type = switch-cas
      number-complete = 1-digits
      group-b-signal = signal-b-6
```

```
group-ii-signal = signal-ii-1
answer-delay = 200
caller-id = no-caller-id
```

Following is an example that sets these parameters for R2 signaling:

```
admin> read e1 {1 7 1}
E1/{ shelf-1 slot-7 1 } read

admin> set line signaling-mode = e1-r2-signaling

admin> set line switch-type = switch-cas

admin> set line group-b-signal = signal-b-6

admin> set line group-ii-signal = signal-ii-1

admin> write
E1/{ shelf-1 slot-7 1 } written
```

The parameters and their effects are shown below:

Parameter	Effect
Signaling-Mode	Specifies the type of signaling used on the E1 line. For R2 signaling, specify E1-R2-Signaling (R2 signaling), E1-Korean-Signaling (a version of the R2 signaling protocol specified for use in Korea), E1-P7-Signaling (P7 signaling), E1-Chinese-Signaling (a version of the R2 signaling protocol specified for use in China), or E1-Metered-Signaling (metered R2 signaling protocol, used in Brazil and South Africa).
Switch-Type	Specifies the type of network switch. For E1 R2 signaling, it should always be set to Switch-CAS.
Number-Complete	Specifies how many digits must be received of the dialed number of an incoming call using R2 signaling. You can indicate up to 10 digits of a phone number that must be received, or specify the End-of-Pulsing signal (to signify that the full number has been received). In all cases, the digits received before the call is answered are considered the called number for call-routing purposes.
Group-B-Signal Group-II-Signal	The Group B signal is the signal sent immediately before answering an incoming call. See “E1_R2 Israeli signaling” on page 16 for updated information about this parameter. The Group II signal is the signal sent in the course of an outgoing call, immediately after acknowledgment by the called end that all necessary address digits have been received. You can set the Group-B-Signal parameter to a value from B-1 through B-15, and the Group-II-Signal to a value from II-1 through II-15. For systems in Mexico and Korea, set these values to Signal-B-1 and Signal-II-2, respectively. For systems in Argentina, set these values to Signal-B-6 and Signal-II-1, respectively. For information about the proper settings for other countries, please contact your carrier.
Answer-Delay	Specifies the number of milliseconds to delay before answering a call. Use this parameter if the MAX TNT answers calls too quickly. You can set it to a number from 100 to 3000.

Parameter	Effect
Caller-ID	Used only with E1-Chinese-Signaling, to request the Calling Line ID (CLID) from the switch.

PRIdisplay command enhancements

PRIdisplay command output for a T1, E1, or T3 card now includes a time stamp relative to the time the card booted, and also includes PRI messages that have bad CRCs or are too long. In addition, the administrator can now specify a single line to be monitored.

To use the PRIdisplay command, open a session with a T1, E1, or T3 card. For example, the following command opens a session with a card in shelf 1, slot 15:

```
admin> open 1 15
```

The PRIdisplay command uses the following syntax:

```
t3-1/15> help pridisplay
priDisplay <n> [ <line> ]
     where <n> is the number of octets to display
     set <n> to zero to turn display off
     where <line> is the line whose d-channel to display
     set <line> to zero specify any line
```

In the following example, the PRIdisplay command displays the first 160 bytes of PRI messages. Notice the time stamp in each message:

```
t3-1/15> pridisplay 160
Display the first 160 bytes of PRI messages
PRI-XMIT-24: 01:38:53: 3 of 3 octets
1010A850: 00 01 7f                                              ...
PRI-RCV-24: 01:38:55: 3 of 3 octets
10112C10: 00 01 7f                                              ...
PRI-RBAD-22: 01:38:53: 2 of 2 octets
1010A850: 00 01
```

In the following example, the first PRIdisplay command displays the first 32 bytes of PRI messages for line 12 only. The second command enables display of the first 32 bytes of messages for any line on the card. The third command turns off the message display:

```
t3-1/15> prid 32 12
Display the first 32 bytes of PRI messages for line 12

t3-1/15> prid 32 0
Display the first 32 bytes of PRI messages

t3-1/15> prid 0
PRI message display terminated
```

To close the session with the card and return to the shelf-controller:

```
t3-1/15> quit
admin>
```

DS3 –i for internal loopback

The DS3link command on a T3 card now supports the –i option for performing an internal loopback for diagnostic purposes. The –i option connects the DS3 receive path to the DS3 transmit path at the D3MX. The transmitted DS3 signal is still sent to the network as well.

To use the DS3link command, open a session with a T3 card. For example, to open a session with a T3 card in shelf 1, slot 15:

```
admin> open 1 15
```

The DS3link command uses the following syntax:

```
t3-1/15> ? ds3link
ds3link get information about the DS3 interface
usage: ds3link [ - a|b|c|d|i|l|s|t|? ]
     -a display current DS3 line alarms
     -b < on | off > transmit DS3 Alarm Indication Signal
     -c display and clear line error statistics
     -d < 1 - 7 > display current DS2 line state
     -i < on | off > loop the DS3 inwards
     -l < on | off > loop the DS3 outwards
     -s display line error statistics without clearing
     -t toggle debug output
     -? display this summary
```

For example, the following command activates a DS3 internal loopback:

```
t3-1/15> ds3link -i on
DS3 internal loopback activated
```

To deactivate the DS3 internal loopback:

```
t3-1/15> ds3link -i off
DS3 internal loopback deactivated
```

Clock-Source command enhancements

The Clock-Source command on T1, E1, or T3 slot cards now lists only currently eligible local clock sources. Sources with layer 2 up, which are preferred, are marked with an asterisk. In addition, a message is now logged whenever the system clock source changes.

To use the Clock-Source command, first open a session with a T1, E1, or T3card. For example, if the T3 card is in shelf 1, slot 15:

```
admin> open 1 15
```

Then enter the Clock-Source command. The following example shows the new Clock-Source output:

```
t3-1/15> clock-source
Master line: 1
Source List:
        Source: line 1 Available*       priority: 2
        Source: line 3 Available        priority: 2
```

Following are examples of log messages generated for clock source transitions:

```
LOG notice, Shelf 1, Controller, Time: 19:44:39--
 Master clock source changed to slot-1/8 line 1
LOG notice, Shelf 1, Controller, Time: 10:34:56--
 Master clock source changed to local oscillator
```

Support for 64K and 56K calls over channels using R2 signaling

The default bandwidth for data calls coming in over E1 channels using R2 signaling is now 64K. To configure a connection to use 56K instead, use the following parameter (shown with its default setting):

```
CONNECTION station
   telco-options
      force-56kbps = no
```

Following is an example of a procedure that specifies 56K for a call coming in over channels using R2:

```
admin> read connection test
CONNECTION/test read

admin> set telco force-56kbps = yes

admin> write
CONNECTION/test written
```

Multiple NFAS groups on T1 and T3 cards

In earlier releases, the MAX TNT supported NFAS (Non-Facility Associated Signaling), a service which allows a single D-channel on one DS1 (with an optional backup D-channel) to control all B channels in some number of additional DS1s (up to 8 on a T1 card and 28 on a T3 card, subject to the limitations of the switch). Each card was limited to a single NFAS group. However, some sites require multiple NFAS groups on a single card to enabled grouped DS1s for different applications.

The MAX TNT now supports NFAS groups. An NFAS group contains a minimum of two PRIs, so a T1 card supports up to four NFAS groups, and a T3 card supports up to 14 NFAS groups. To support this configuration, the T1 profile contains the following new parameter, shown with its default value:

```
T1 { shelf-N slot-N line-N }
   nfas-group-id = 0
```

For a T1 card, you can set the NFAS-Group-ID to a value of from 0 to 3. For a T3 card, valid values are from 0 to 13. Lines with the same NFAS-Group-ID value are in the same NFAS group.

To configure multiple NFAS groups, you must set both the NFAS-Group-ID parameter and the NFAS-ID parameter for each DS1. Within the group, all PRIs share the same NFAS-Group-ID value and have different, unique NFAS-ID values.

In the following example, two NFAS groups are configured on a T1 card. Each group contains four DS1s. The example uses the NFAS group IDs 1 and 2, but you can assign any valid NFAS-Group-ID values.

Note: You must first obtain an NFAS ID for each DS1 from the Telco. Within an NFAS group, each DS1 must have a unique NFAS-ID. Telcos often use NFAS-ID=0 for the PRI with the

primary D-Channel, and NFAS-ID=1 for the PRI with the secondary D-Channel. They then assign to each PRI that does not have a D channel a unique numeric value.

Following is an example of configuring two NFAS groups on a T1 card, with each group containing four PRIs. The next group of commands configure NFAS group 1, which contains lines 1 through 4:

```
admin> read t1 {1 2 1}
T1/{ shelf-1 slot-2 1 } read
admin> set line signaling-mode = isdn-nfas
admin> set line nfas-id = 0
admin> set line nfas-group-id = 1
admin> set channel 24 channel = nfas-primary
admin> write
T1/{ shelf-1 slot-2 1 } written
admin> read t1 {1 2 2}
T1/{ shelf-1 slot-2 2 } read
admin> set line sig = isdn-nfas
admin> set line nfas-id = 1
admin> set line nfas-group-id = 1
admin> set line channel 24 channel = nfas-secondary
admin> write
T1/{ shelf-1 slot-2 2 } written
admin> read t1 {1 2 3}
T1/{ shelf-1 slot-2 3 } read
admin> set line sig = isdn-nfas
admin> set line nfas-id = 2
admin> set line nfas-group-id = 1
admin> write
T1/{ shelf-1 slot-2 3 } written
admin> read t1 {1 2 4}
T1/{ shelf-1 slot-2 4 } read
admin> set line sig = isdn-nfas
admin> set line nfas-id = 3
admin> set line nfas-group-id = 1
admin> write
T1/{ shelf-1 slot-2 4 } written
```

The following commands configure NFAS group 2, which contains lines 5 through 8:

```
admin> read t1 {1 2 5}
T1/{ shelf-1 slot-2 5 } read
admin> set line signaling-mode = isdn-nfas
admin> set line nfas-id = 0
admin> set line nfas-group-id = 2
admin> set channel 24 channel = nfas-primary
```

```
admin> write
T1/{ shelf-1 slot-2 5 } written
admin> read t1 {1 2 6}
T1/{ shelf-1 slot-2 6 } read
admin> set line sig = isdn-nfas
admin> set line nfas-id = 1
admin> set line nfas-group-id = 2
admin> set line channel 24 channel = nfas-secondary
admin> write
T1/{ shelf-1 slot-2 6 } written
admin> read t1 {1 2 7}
T1/{ shelf-1 slot-2 7 } read
admin> set line sig = isdn-nfas
admin> set line nfas-id = 2
admin> set line nfas-group-id = 2
admin> write
T1/{ shelf-1 slot-2 7 } written
admin> read t1 {1 2 8}
T1/{ shelf-1 slot-2 8 } read
admin> set line sig = isdn-nfas
admin> set line nfas-id = 3
admin> set line nfas-group-id = 2
admin> write
T1/{ shelf-1 slot-2 8 } written
```

E1_R2 Israeli signaling

The relevant specifications for this signaling type are in ITU-T recommendations Q.400 to Q.490 and Israeli MFC-R2 Register Signaling documentation. The following parameters support configuration for E1_R2 Israeli signaling, shown with their default values, which work for systems in Israel:

```
E1 {shelf-N slot-N N}
   line-interface
      group-b-answer-signal = signal-b-6
      group-b-busy-signal = signal-b-3
```

For example:

```
admin> read e1 {1 7 1}
E1/{ shelf-1 slot-7 1 } read
admin> set line group-b-answer-signal = signal-b-6
admin> set line group-b-busy-signal = signal-b-3
admin> write
E1/{ shelf-1 slot-7 1 } written
```

Group-B-Answer-Signal replaces the Group-B-Signal parameter found in earlier releases. It specifies the group-B signal that the MAX TNT sends before answering a call, and can be set

to a value from Signal-B-1 to Signal-B-15. The default is Signal-B-6, which is the recommended setting for E1_R2 Israeli signaling.

Group-B-Busy-Signal specifies the group-B signal that the MAX TNT sends before sending a busy signal, and can be set to a value from Signal-B-1 to Signal-B-15. The default is Signal-B-3, which is the recommended setting for E1_R2 Israeli signaling.

Note: When the MAX TNT does not have sufficient resources to handle the call correctly (for example, if all of its modems are busy), it sends the group-B signal specified by the Group-B-Busy-Signal parameter.

Features for modem and HDLC cards

See also “MP and MP+ bundled channels span HDLC cards and shelves” on page 7.

V.34 setting for 56k modem cards

This release supports V.34 modem modulation for the 56K modem cards. You configure this feature by setting the following parameter, which is shown with its default value:

```
TERMINAL-SERVER
   modem-configuration
      modem-mod = k56-modulation
```

To support the ITU standard V.8bis (Voice Call Ready), a 56K modem in the MAX TNT normally sends a tone at the beginning of modem training. This is commonly referred to as CRe and is a dual tone (1375Hz + 2002 Hz) followed by a single tone at 400Hz with a combined duration of approximately 500 ms. Although V.8bis is designed not to interfere with V.32bis (which supports a maximum rate of 14.4 Kbps) modem negotiation, some V.32 and V.34 modems do not successfully complete modem training after reception of the V.8bis tone.

The Modem-Mod parameter has two settings: K56-Modulation and V34-Modulation. When it is set to V34-modulation, 56K modem cards never exceed the speeds used by V.34 modems (33.6k) and do not send the V.8bis tone.

For example, the following commands configure V.34 modulation for calls coming in to 56K modem cards:

```
admin> read terminal-server
TERMINAL-SERVER read

admin> set modem-configuration modem-mod = v34-modulation

admin> write
TERMINAL-SERVER write
```

Command for displaying WAN session data

The wanDisplay and wanOpening commands already supported on host cards (modem or HDLC cards) enable the administrator to turn on or turn off a display of WAN session data as it is received and transmitted. The wanDisplay command shows WAN data for all sessions; wanOpening shows WAN data only during connection establishment. This release provides a new command, wanDSess, that shows WAN data for particular user sessions.

All of the commands display the WAN session data in the following format:

```
RECV-93:: 58 of 58 octets
1017FD44: 7e ff 7d 23 c0 21 7d 21 7d 21 7d 20 7d 3b 7d 21    ~.}#.!}! }!}
1017FD54: 7d 24 7d 25 f4 7d 22 7d 26 7d 20 7d 2a 7d 20 7d    }$}%.}"} &}
1017FD64: 20 7d 27 7d 22 7d 28 7d 22 7d 33 7d 29 7d 23 7d     }'}"}(} "}3
1017FD74: 20 c0 7b 6d 28 20 c4 7d 2c 7e                      .{m( .} ,~

XMIT-93:: 58 of 58 octets
100173F8: 7e ff 7d 23 c0 21 7d 21 7d 21 7d 20 7d 3b 7d 21    ~.}#.!}! }!}
10017408: 7d 24 7d 25 f4 7d 22 7d 26 7d 20 7d 2a 7d 20 7d    }$}%.}"} &}
10017418: 20 7d 27 7d 22 7d 28 7d 22 7d 33 7d 29 7d 23 7d     }'}"}(} "}3
10017428: 20 c0 7b 63 42 cf 7d 23 9b 7e                      .{cB.}# .~
```

The wanDSess command uses the following syntax:

```
wandsess session-name octets
```

where *session-name* is the name of a local Connection profile or a RADIUS profile, and *octets* specifies the maximum number of octets per frame. For example:

```
modem-1/7> wandsess tlynch 16

RECV-95:: 16 of 89 octets
101808DC: 7e 21 45 00 00 54 6f 2c 00 00 ff 01 81 a3 01 02    ~!E..To,
XMIT-95:: 16 of 90 octets
100138B8: 7e 21 45 00 00 54 31 f9 00 00 ff 01 be d6 c0 a8    ~!E..T1.
RECV-95:: 16 of 89 octets
1017FD44: 7e 21 45 00 00 54 71 2c 00 00 ff 01 7f a3 01 02    ~!E..Tq,
XMIT-95:: 16 of 89 octets
100138B8: 7e 21 45 00 00 54 33 f9 00 00 ff 01 bc d6 c0 a8    ~!E..T3.
```

Terminal server login timeout

When a user logs into the terminal server in terminal mode, a login prompt appears. If the user does not proceed any further than the login prompt within 300 seconds, the login times out. The administrator can now configure this timeout value by setting the following parameter, shown with its default value:

```
TERMINAL-SERVER
   terminal-mode-configuration
      login-timeout = 300
```

For example, to set the timeout to 60 seconds:

```
admin> read terminal
TERMINAL-SERVER read

admin> set terminal login-timeout = 60

admin> write
TERMINAL-SERVER written
```

If you set the Login-Timeout parameter to zero, the login never times out.

New call-routing sort method for digital calls

The following new parameter has been added to the System profile to enable the administrator to use different call-routing sort methods for digital and analog calls:

```
SYSTEM
   digital-call-routing-sort-method = slot-first
```

When Digital-Call-Routing-Sort-Method is set to Slot-First (the default), the MAX TNT sorts digital calls by shelf and slot number, and then by item number. This improves system performance for MP or MP+ calls by concentrating the channels of a call on one HDLC card.

When Digital-Call-Routing-Sort-Method is set to Item-First, the MAX TNT sorts the calls by item number, then shelf, and then slot number. This setting distributes incoming calls evenly across multiple HDLC cards. Distributing calls across cards for bundled channels creates extra processing overhead.

The Call-Routing-Sort-Method parameter in the System profile now specifies the sort method for analog calls only. It is set to Item-First by default, which means that analog calls are distributed evenly across multiple host cards.

Direct-access modem dialout over 56K modems

Direct-access dialout enables users to dial out over the MAX TNT 56K modems. You can configure the feature by setting the following parameters, shown with their default values:

```
TERMINAL-SERVER
   dialout-configuration
      enabled = no
      direct-access = no
      port-for-direct-access = 5000
      password-for-direct-access = ""
      security-for-direct-access = none
```

Following are descriptions of the Dialout-Configuration parameters:

Parameter	Description
Enabled	Controls whether modem dialout is allowed. If set to No, none of the other parameters in the Dialout-Configuration subprofile apply.
Direct-Access	Enables the direct-access dialout feature. If set to Yes, users can Telnet to a particular port on the MAX TNT to get immediate dialout service. The port number configured as the Port-for-Direct-Access tells the MAX TNT that all Telnet sessions to port want immediate modem access. If set to No, the remaining parameters in the Dialout-Configuration subprofile do not apply.
Port-for-Direct-Access	Specifies the TCP port for immediate dialout service. Must be set to an integer from 5000 to 32767 if Direct-Access is enabled. The default setting is 5000.
Security-for-Direct-Access	Specifies the type of security used for direct-access dialout. If set to Global, the Password-for-Direct-Access parameter must specify a password, which will be required from users Telneting to the specified TCP port. If Security-for-Direct-Access is set to None, no password is required for dialing out. If it is set to User, a local Connection or RADIUS profile must be configured to allow dialout.

Parameter	Description
Password-for-Direct-Access	The password (up to 64 characters) used for Global mode authentication. If Security-for-Direct-Access is not set to Global, this parameter is ignored.

Example of direct-access dialout with global password

Following is an example of setting up direct-access dialout, in this case on TCP port 5028 with a single global password (“pizza”) required for modem access:

```
admin> read terminal-server
TERMINAL-SERVER read

admin> list dialout-configuration
enabled = no
direct-access = no
port-for-direct-access = 5000
password-for-direct-access = ""
security-for-direct-access = none

admin> set enabled = yes

admin> set direct-access = yes

admin> set port = 5028

admin> set password = pizza

admin> set security = global

admin> write
TERMINAL-SERVER written
```

Example of direct-access dialout with user passwords

The commands entered in the following example set up direct-access dialout on TCP port 5000 and specify that Connection or RADIUS profiles are required for modem access:

```
admin> read terminal-server
TERMINAL-SERVER read

admin> list dialout-configuration
enabled = no
direct-access = no
port-for-direct-access = 5000
password-for-direct-access = ""
security-for-direct-access = none

admin> set enabled = yes

admin> set direct-access = yes

admin> set security = user

admin> write
TERMINAL-SERVER written
```

Example of a Connection profile for modem dialout

You can configure a Connection profile to allow modem dialout by setting the following parameters, shown with their default values:

```
CONNECTION station
   telco-options
      data-service = 56k-clear
      dialout-allowed = no
```

Following are descriptions of the Connection Telco-Options parameters related to modem dialout:

Parameter	Description
Data-Service	Specifies the bandwidth and service of outgoing calls. When the MAX TNT initiates a dialout, it requests a data service and bandwidth rate, and the answering end rate-adapts. The Modem data service setting is related only to dialouts, it is not required for inbound modem calls.
Dialout-Allowed	Specifies whether the profile may be used for dialing out on one of the MAX TNT unit's digital modems. Only if it is set to Yes will the local user be allowed to dial out on a modem to the destination specified in the Connection profile.

The following example shows how to configure the Telco-Options subprofile to allow dialout:

```
admin> read connection test
CONNECTION/test read

admin> set telco data-service = modem

admin> set telco dialout-allowed = yes

admin write
CONNECTION/test written
```

How a user dials out with direct-access

A user executes the following steps to dial out over one of the MAX TNT 56K modems when direct-access has been configured with global security:

1. Telnet to the MAX TNT from a workstation, specifying the direct-access port number on the command line. For example:

   `telnet tnt01 5000`

   Where the first argument, tnt01, is the system name of the MAX TNT and the second argument, 5000, is the direct-access port.

2. When prompted for a password, enter the Password-for-Direct-Access.

3. Use the standard Rockwell AT commands to dial out on the modem, just as if using a modem connected directly to a workstation. For example:

   `ATDT 555-1212` `^M`

4. To terminate the session with the modem, terminate the Telnet session.

Note: Direct-access dialout uses the Telnet protocol, rather than a raw TCP connection, for communicating with client processes. This means that any client process wishing to use this service to transmit or receive binary data must, at a minimum, escape outgoing IAC (0xFF) characters, handle escaped incoming IAC characters, and strip out incoming Telnet options. For a description of the Telnet protocol and how it differs from a raw TCP connection, see RFCs 854 and 855.

Performance enhancements for TCP-Clear calls

In this release, data from TCP-Clear dialup sessions that do not require V.120 processing can be buffered and transmitted as TCP packets rather than as a continuous data stream, which increases performance for these connections. In addition, unless V.120 processing is required, TCP-Clear WAN data is now sent directly to the HDLC interface rather than to the terminal-server subsystem. (If V.120 processing is required, the call is processed by the terminal server, as in previous releases.)

The system does not collect session statistics for TCP-Clear calls that make use of these enhancements.

Parameters for setting up packet buffering

Following are the parameters relevant to TCP-Clear packet buffering. The parameters are shown with their default values:

```
CONNECTION
   tcp-clear-options
      detect-end-of-packet = no
      end-of-packet-pattern = ""
      flush-length = 256
      flush-time = 20

ANSWER-DEFAULTS
   tcp-clear-answer
      detect-end-of-packet = no
      end-of-packet-pattern = ""
      flush-length = 256
      flush-time = 20
```

Detect-End-of-Packet specifies whether the MAX TNT buffers incoming WAN data. If set to Yes, after the dialup session has been authenticated, the MAX TNT begins buffering incoming WAN data until it receives the specified End-of-Packet-Pattern, or until it reaches the specified timeout (Flush-Time) or maximum packet length (Flush-Length), whichever comes first. If Detect-End-of-Packet is set to No (the default), none of the related parameters apply.

Flush-Length specifies the maximum number of bytes to buffer. Valid values are from 1 to 8192. The default value is 256. (Note that buffering large packets consumes more system resources.) If the system has buffered the specified number of bytes without matching the End-of-Packet-Pattern, it flushes the buffer by writing the data to TCP.

Flush-Time specifies a timer in milliseconds. Valid values are from 1 to 1000. The timer begins counting down on received of the first byte of buffered data. If the specified number of msecs has elapsed without matching the End-of-Packet-Pattern, the system flushes the buffer by writing the data to TCP.

End-of-Packet-Pattern defines a character pattern that signals the end of a packet. When the MAX TNT matches this pattern in the buffered data, it immediately flushes the buffer by writing all data up to and including the pattern out to TCP. Note that data may be written before a match occurs based on a specified timeout (Flush-Time) or packet length (Flush-Length).

The character pattern can be up to 64 characters long. It can contain both ASCII characters and other binary data using the backslash (\) as an escape mechanism. To insert a literal backslash in the pattern, escape it by entering two backslash characters (\\).

To insert a 1 to 3 digit octal number, escape the value using the single backslash. (To avoid confusion between the literal ASCII characters 0 through 7 and an octal value, you can pad the octal value with leading zeros.) For example, the following pattern represents a carriage return (octal 15):

```
\015
```

To insert a 1 or 2 digit hexadecimal number in the pattern, precede the number with \x. For example, the following pattern represents a carriage return (hex 0D):

```
\x0D
```

Other special escape sequences are shown below:

Escape Sequence	Description	Value
\a	Alarm	7
\b	Backspace	8
\f	Form feed	12
\n	New line	10
\r	Carriage return	13
\t	Tab	9
\v	Vertical tab	11
\\	Backslash	92
\’	Apostrophe	44
\"	Double Quote	34
\?	Wildcard	Matches any single character

Example configuration for packet buffering

The following procedure shows one example of how to set up a Connection profile to buffer WAN packets. This is a test configuration to the echo generator server of a host (port 7). The End-of-Packet-Patter is set to three hex numbers.

```
admin> read connection jim
CONNECTION/jim read

admin> set encaps = tcp-raw

admin> list tcp-clear
host = sparky
port = 7
detect-end-of-packet = no
end-of-packet-pattern = ""
flush-length = 256
flush-time = 20

admin> set detect-end-of-packet = yes

admin> set end-of-packet-pattern = \xfe\xfd\xfe

admin> set flush-length = 16

admin> write
CONNECTION/jim written
```

Features for xDSL cards

POST and loopback test for xDSL cards

An all-channel loopback test is now part of POST for SDSL and ADSL-CAP cards. In addition, SDSL and ADSL-CAP cards now support the XDSLcmd command to activate a loopback test manually. For SDSL cards, the command uses the following syntax:

```
sdsl-1/6> xdslcmd -?
usage: xdslCmd [ - l|? ][ channel ][[count] [bufferSize]]
   -l  LoopBack on Channel
   -?  display this summary
   channel: channel to test (0-15, none = all)
```

For ADSL-CAP cards, the command uses the following syntax:

```
adsl-1/16> xdslcmd -?
usage: xdslCmd [ - l|? ][ channel ][[count] [bufferSize]]
   -l  LoopBack on Channel
   -?  display this summary
   channel: channel to test (0-5, none = all)
```

The command arguments are described below:

Argument	Description
–l	Initiates a loopback on the specified channel.
channel	The channel to test. If no channel is specified, all channels are tested.
Count	The number of looped frames. The default is 10.
Bufsize	The size of the looped frames. The default is 128 bytes.

The following example shows how to run a loopback test on channel 8 of an SDSL card in shelf 1, slot 6:

```
admin> open 1 6
sdsl-1/6> xdslcmd -l 8
```

New commands for checking the IDSL card

Tests have been added to the POST functions for the IDSL card, and two new commands are supported for testing the IDSL card. To use the commands, you must first open a session with the card. For example, to open a session with an IDSL card in shelf 1, slot 7:

```
admin> open 1 7
```

IDSL command for loopback and error tests

The new IDSLcmd command uses the following syntax:

```
idsl-1/7> idslcmd
usage: idslCmd [ - 1|2|f|z|c|u|r|n|? ][ channel ]
       -1  LoopBack on B1-channel
       -2  LoopBack on B2-channel
       -a  Analog loopBack on B1-channel
```

```
        -f  Fetch Block error counters
        -z  Clear Block error counters
        -c  corrupt CRC
        -u  cancel corrupt CRC
        -r  request corrupt CRC
        -n  cancel request corrupt CRC
        -?  display this summary
channel: channel to test (0-31, none = all)
```

EOC loopback over the B channels

The first two options, 1 and 2, initiate EOC (Embedded Operations Channel) loopback on the specified B channel. EOC refers to the out-of-band mechanism available in the BRI-U interface to implement maintenance functions. It is out-of-band in that it does not use either the D or B channels but uses the maintenance bits of U-interface superframe, so it is non-intrusive. Maintenance functions include test loopbacks as well as statistics gathering (block error counters) and request to generate errors (to check that the counters work).

When you use the 1 or 2 option, the command accepts additional arguments in the following syntax:

```
idsl-1/7> idslcmd -channel [EOC address] [count] [bufsize]
```

The command arguments are described below:

Argument	**Description**
channel	The channel may be 1 or 2, representing a B channel.
EOC address	A number from 0 to 7. The default is zero, which addresses the remote TA (NT). The numbers 1 to 6 address the nodes in between, with 1 being closest to the IDSL card.The number 7 broadcasts the EOC loopback to all the nodes.
Count	The number of buffers to be sent in the loopback. The default is 10.
Bufsize	The size of the buffer to be sent. The default is 128 bytes.

For example:

```
idsl-1/7> idslcmd -1 0 64 32
```

The preceding command requests the remote TA or BRI-U device (specified by the EOC address) to go into Loopback mode over the B1 channel and sends 64 frames of size 32. A message is displayed to report the number of frames received back from the TA in which the size and the payload matches what was sent.

Analog loopback over the B channels

With the –a option, the IDSLcmd command requests an analog loopback, testing data paths between components of the card itself. The –a option requires that you specify a channel number and accepts additional arguments in the following syntax:

```
idsl-1/7> idslcmd -a channel [count] [bufsize]
```

For example:

```
idsl-1/7> idslcmd -a 1 64 32
```

The preceding command puts the U interface echo canceller (IEC-2091) in analog loopback mode and sends 64 frames of size 32 over channel B1. A message is displayed to report the number of frames received back in which the size and the payload matches. This test is adequate to verify the path between the HDLC controllers and the IEC-2091 echo canceller.

Block error fetching and error clearing

The remote U interface/echo canceller provides internal counters for far-end and near end block errors. This allows comfortable surveillance of the transmission quality at the U-interface. A block error is detected each time when the calculated checksum of the received data does not correspond to the control checksum transmitted in the successive superframe. One block error thus indicates that one U-superframe has not been transmitted correctly. No conclusion with respect to the number of bit errors is possible from the block error counters.

A near-end block error (NEBE) indicates that the error has been detected in the receive direction. A far-end block error (FEBE) identifies errors in the transmission direction. These are all from the point of view of the remote TA, since the error counters we are fetching are those of the remote TA.

With the –f option, the IDSLcmd command fetches block error counters for the specified channel. For example:

```
idsl-1/7> idslcmd -f 1
```

Block error counters are cumulative and stop accumulating once the upper limit of 65534 has been reached. To clear the block error counters for a channel, use the –z option, as shown in the next command:

```
idsl-1/7> idslcmd -z 1
```

Performing CRC error tests

To test the NEBE and FEBE counters, you can simulate transmission errors by artificially corrupting CRCs. With the –c option, the ISDLcmd command inverts CRC to purposely generate CRC errors. The remote FEBE should then increment for every corrupt frame it receives. For example, to invert CRC on channel 2:

```
idsl-1/7> idslcmd -c 2
```

To cancel this command and return CRC to normal, use the –u option. For example:

```
idsl-1/7> idslcmd -u 2
```

Conversely, you can use the –r option to request the remote TA to invert CRC. Then, the remote NEBE should increment for every corrupt frame sent. For example:

```
idsl-1/7> idslcmd -r 1
```

To cancel this command and return CRC to normal, use the –n option. For example:

```
idsl-1/7> idslcmd -n 2
```

BRIdisplay command for displaying D-channel traffic

The new BRIdisplay command uses the following syntax:

```
idsl-1/7> bridisplay
bridisplay <count> [channel]
```

The command displays up to the specified *count* number of bytes of the specified D channel. The optional channel argument specifies one of the 32 D channels of an IDSL card—its valid range is 0–31. If you specify a channel number, only traffic on that D channel is displayed, otherwise, no filter on D channels is applied and all traffic on all D channels is displayed. For example, the following command displays up to 12 bytes of the traffic in every D channel on the card:

```
idsl-1/7> bridisplay 12
```

To turn off the display, set *count* to zero; for example:

```
idsl-1/7> bridisplay 0
```

New SDSL statistics reported

The HDLC-Rx-CRC-Error-Cnt parameter has been added to the SDSL-Statistics profile to show the number of CRC errors occurring on the line.

```
admin> get sdsl-statistics { 1 3 1}
physical-address* = { shelf-1 slot-3 1 }
line-up-timer = { 0 18 45 }
rx-signal-present = yes
line-quality = 15
up-dwn-cntr = 2
self-test = passed
far-end-db-attenuation = 4
firmware-startup-stage = normal-operation
hdlc-rx-crc-error-cnt = 0
```

The value of the HDLC-Rx-CRC-Error-Cnt parameter shows how many CRC errors have occurred. It is normal to show a few CRC errors, but the line is disconnected if 1500 errors occur within 2 second time period.

New maximum down-stream rate for ADSL-CAP

In this release, the Asymmetric Digital Subscriber Line (ADSL) Carrierless Amplitude Phase (CAP) card supports the 5120000 down-stream rate configuration. Following is the relevant parameter, which is shown with its default value:

```
ADSL-CAP { shelf-N slot-N N}
   line-config
      max-down-stream-rate = 2560000
```

The Max-Down-Stream-Rate parameter already supported settings of 7168000, 5120000, and 2560000. It now also supports the additional settings in the following list. These settings represent the maximum down-stream rates the transceiver supports.

- 640000
- 960000
- 1280000
- 1600000
- 1920000
- 2240000
- 2560000
- 2688000

- 3200000
- 4480000
- 5120000
- 6272000
- 7160000

Note: The CPE maximum down-stream rate defaults to 7160000. The COE maximum down-stream rate defaults to 2560000. To adjust the down-stream rates, configure the COE ADSL-CAP profile. The loop will train to the lower of the two rates.

If there is poor loop quality, the transceiver will choose a rate lower than the maximum rate, and a good loop quality causes the transceiver to choose a rate close to or at the Maximum-Down-Stream-Rate setting. If the loop quality is very poor, the transceiver will not train at all, and will be unable to connect to the remote side. In that case, the administrator must specify a lower maximum down-stream rate, because the transceiver does not cross rate boundaries.

For example, if the transceiver is configured for 7160000bps and the loop quality is very poor to the point that the transceiver will not connect to the remote side, the transceiver does not automatically adjust the down-rate into the 5120000bps range. The administrator needs to configure the Max-Down-Stream-Rate to the lower rate.

The following example shows commands that set the maximum down-stream rate to 5.12Mbps, and the system's responses:

```
admin> read adsl-cap {1 11 1}
ADSL-CAP/{ shelf-1 slot-11 1 } read

admin> set line-config max-down-stream-rate = 5120000

admin> write
ADSL-CAP/{ shelf-1 slot-11 1 } written
```

The status of the down-stream functionality is displayed as follows int he ADSL-CAP-Status profile. For example:

```
admin> read adsl-cap-status { 1 11 1}
ADSL-CAP-STATUS/{ shelf-1 slot-11 1 } read

admin> list
physical-address* = {  shelf-1 slot-11 1 }
if-group-index = 0
unit-type = coe
dev-line-state = port-up
up-stream-rate = 952000
down-stream-rate = 5120000
major-firmware-ver = 232
minor-firmware-ver = 0
hardware-ver = 0
up-stream-constellation = 256
down-stream-constellation = 256u
down-stream-operational-baud = 680
```

The Down-Stream-Operational-Baud parameter now displays 680 and Down-Stream-Rate displays 5120000.

Note: You can set the maximum down-stream rate for using SNMP utilities by writing the DownRate object in the AdslCapLineStatusEntry MIB. The DownRate object supports Read

and Write operations, and supports the same settings as the Maximum-Down-Stream-Rate parameter.

Nailed connections supported via IDSL

In previous releases, the IDSL card supported only switched connections. A terminal adapter (TA) device, such as a Pipeline, was configured for a switched line and assigned an arbitrary phone number. In this release, the IDSL card also supports nailed connections, and the TA connection can be configured for a nailed/leased line.

You can configure only one channel on an IDSL line for nailed usage. You must also assign that channel a group number. The Connection profile to the TA then references the assigned group number in its Nailed-Groups setting, to direct the connection to use the IDSL nailed channel.

The following parameters, shown with sample settings, configure an IDSL channel for nailed usage:

```
IDSL/{ shelf-N slot-N N }
   line-interface
      enabled = yes
      channel-config N
         channel-usage = nailed-64-channel
         nailed-group = 1
```

The Enabled parameter was previously named Line-Enabled. It enables the IDSL line for use.

The Channel-Usage parameter was previously Read-Only. It specifies the usage for the channel. Channel-Usage can specify one of the following values:

- Unused-Channel specifies that the channel is unused. The MAX TNT sends the single idle code defined for the channel.
- Switched-Channel (the default) specifies a switched channel.
- Nailed-64-Channel specifies a clear-channel 64k circuit. It does not require any setup information.

The Nailed-Group parameter is new. It specifies a group number for the nailed channel (from 0 to 65535, set to zero by default). To use the nailed/leased line, the Connection profile to the TA must reference the assigned group number in its Nailed-Groups setting.

The following example shows how to configure a nailed channel on the first channel of line 18 of an IDSL card in shelf 1, slot 7:

```
admin> read idsl {1 7 18}
IDSL/{ shelf-1 slot-7 18 } read

admin> set line enabled = yes

admin> list line channel 1
channel-usage = switched-channel
nailed-group = 0

admin> set channel-usage = nailed

admin> set nailed-group = 10

admin> write
IDSL/{ shelf-1 slot-7 18 } written
```

For information about using the BRIchannels command to verify channel usage, see the *MAX TNT Hardware Installation Guide*.

Following is an example of a Connection profile that makes use of the nailed IDSL channel:

```
admin> read connection pipeline
CONNECTION/pipeline read

admin> list telco
answer-originate = ans-and-orig
callback = no
call-type = off
nailed-groups = 0
ft1-caller = no
force-56kbps = no
data-service = 56k-clear
call-by-call = 0
billing-number = ""
transit-number = ""
expect-callback = no
dialout-allowed = no
delay-callback = 0

admin> set nailed-groups = 10

admin> write
CONNECTION/pipeline written
```

Frame-Relay circuit switching on xDSL cards

Frame-Relay-Circuit mode is now enabled in Connection profiles that are connected through ports on ADSL, SDSL, and IDSL cards. For details about configuring this feature, see “Frame Relay Switching” on page 61.

IP and OSPF routing features

OSPF does not support virtual IP interfaces

The Ascend OSPF implementation conforms with RFC 1583 and does not support virtual IP interfaces. A virtual IP interface is one created by the administrator and associated with a physical LAN interface in the MAX TNT. For example, in the following listing the first port on the Ethernet card in slot 15 (shelf-1, slot-15, port 1) has three virtual interfaces:

```
admin> dir ip-int
     8  01/14/1998 14:43:14  { { shelf-1 slot-15 2 } 0 }
     8  01/14/1998 14:43:14  { { shelf-1 slot-15 3 } 0 }
     8  01/14/1998 14:43:14  { { shelf-1 slot-15 4 } 0 }
    20  01/14/1998 14:57:48  { { shelf-1 controller 1 } 0 }
    11  01/14/1998 15:24:28  { { shelf-1 slot-15 1 } 0 }
    10  02/04/1998 11:56:47  { { shelf-1 slot-15 1 } 1 }
    10  02/04/1998 11:57:01  { { shelf-1 slot-15 1 } 2 }
    10  02/04/1998 11:57:09  { { shelf-1 slot-15 1 } 3 }
```

OSPF can be enabled on any one of the port’s IP interfaces, but not on more than one interface for the same port.

IP port caching

Although IP route-caching has been implemented in the MAX TNT for some time, the route caching mechanism does not affect traffic that is being directed to the MAX TNT itself at a higher protocol layer, such as the traffic in a TCP-Clear session.

In a TCP-Clear session, a TCP connection is established between the receiving slot card for the client dial-in (such as a modem card) and a server on the IP network, which is accessible through the destination card (such as an Ethernet card). TCP packets containing the client terminal byte stream are created by the modem card and sent to the server. In this example, the packets from modem card to server can be routed via IP route-caching directly to the Ethernet card. In the reverse direction, server to client, there is no IP route cache, because the packet is destined for the MAX TNT system itself. So the packet is delivered to the router, where it is forwarded to the modem card based on the destination port number.

The following parameter, shown with its default value, enables IP packet forwarding card-to-card based on the packet destination IP address and port:

```
IP-GLOBAL
   ip-port-cache-enable = yes
```

If you set this parameter to no, packets destined for the MAX TNT itself are routed from the receiving slot card to the destination slot card through the shelf-controller, rather than being forwarded directly from the receiving slot card.

TCP-timeout parameter enabled

In this release, the TCP-timeout parameter in the IP-Global profile is enabled. TCP-timeout applies to all TCP connections initiated from the MAX TNT, including Telnet, Rlogin, TCP-clear, and the TCP portion of DNS queries. It applies to established TCP connections as well as to initial attempts to connect. For example, when a user enters a hostname using client software in a terminal server session, and DNS returns a list of IP addresses for the host, if the first address proves unreachable and the timeout on each attempt is long, the client software often times out before finding a good address. The TCP-timeout parameter enables the administrator to adjust the TCP retry timer so that each unsuccessful connection attempt will terminate quickly, allowing more rapid progress through the list, to a good address if one is present.

Following are the relevant parameters, which are shown with their default values:

```
IP-GLOBAL
   tcp-timeout = 0
   dns-list-size = 6
```

Note that the maximum value for DNS-List-Size is 35 addresses.

Valid values for TCP-timeout are from 0 to 200 seconds. At the default value (0), the system attempts a fixed number of retries at escalating intervals, adding up to about 170 seconds total. (Other limits in the system terminate TCP retries after about 170 seconds, even if the parameter is set to a higher number.) If you set TCP-timeout to a non-zero value, the value is the number of seconds TCP retries persist. After the specified number of seconds, the retries stop and the connection is considered lost.

The optimal setting for the TCP-timeout parameter must be determined by experience, and depends on the characteristics of the TCP destination (server) hosts. For example, if the

destinations are all on a LAN under the same administrative control as the MAX TNT and are lightly loaded, then a short timeout (such as a few seconds) might be reasonable, because a host that does not respond within that interval is probably down. Conversely, if the environment includes servers with longer network latency times, such as those connected across the WAN, or load is high in the network or the router, or the characteristics of the remote hosts are not well known, a longer timeout is appropriate. Values of 30 to 60 seconds are common in UNIX TCP implementations.

OSPF global option for disabling ASBR calculations

Autonomous System Boundary Routers (ASBRs) perform calculations related to external routes. Normally, when the MAX TNT imports external routes from RIP (for example, when it establishes a WAN link with a caller that does not support OSPF) it performs the ASBR calculations for those routes. Now, you can prevent the MAX TNT from performing ASBR calculations by setting the following parameter, which is shown with its default value:

```
IP-GLOBAL
   ospf-global
      as-boundary-router = yes
```

If you set the AS-Boundary-Router parameter to No, the MAX TNT does not perform ASBR calculations.

Change in ASE route handling for NSSA configurations

For Not So Stubby Areas(NSSAs), all routes imported to OSPF must have the P-bit set (P stands for *propagate*). When the P-bit is enabled, Area Border Routers translate Type-7 LSAs to Type-5 LSAs, which can then be flooded to the backbone.

When the MAX TNT is configured to route OSPF in an NSSA, all external routes that are imported to OSPF now have the P-bit enabled in their respective link-state entry. These external routes are considered Type-7 ASE LSAs. They may be routes defined in local Connection profiles or RADIUS profiles, or static routes defined in IP-Route profiles.

Note: In previous releases, the ASE7-Adv parameter in IP-Route profiles provided a way to disable the P-bit for static routes imported to OSPF in an NSSA, to prevent those routes from being propagated to the backbone. This is no longer the case. The P-bit is now always enabled for ASE routes, so the MAX TNT disregards the setting of this parameter.

Local DNS table

The MAX TNT can now maintain a DNS table in RAM of up to 8 hostnames and their IP addresses. It consults the table in RAM for address resolution only if requests to the DNS server fail. The local table acts as a safeguard to ensure that the MAX TNT can resolve the local set of DNS names in case all DNS servers become unreachable or go down.

The local DNS table is propagated to RAM from a configured DNS-Local-Table subprofile in the IP-Global profile. At system startup, the values in the profile are copied to the table in RAM. If the administrator subsequently modifies the DNS-Local-Table subprofile, the changes are propagated to the table in RAM when the profile is written.

The DNS table in RAM has space for up to 35 IP addresses per hostname entry (the limit set by the maximum DNS-List-Size). The DNS-Local-Table subprofile allows a single IP address per hostname. For related information, see “Using the Auto-Update feature” on page 34.

To set up the local DNS table, the administrator configures the following parameters in the IP-Global profile, which are shown with their default values:

```
IP-GLOBAL
   dns-local-table
      enabled = no
      auto-update = no
      table-config [1]–[8]
         host-name = ""
         ip-address = 0.0.0.0
```

Parameter	Effect
Enabled	Specifies whether the local DNS table in RAM will be available if DNS queries fail. If set to No (the default), and a DNS query times out, the request fails. If set to Yes, the MAX TNT attempts to resolve the query by consulting the DNS table in RAM. If the hostname in the DNS query has an entry in the table in RAM, the system returns the associated IP address(es) to the requester.
Auto-Update	Specifies whether the local table is automatically updated by regular successful DNS queries. For details, see “Using the Auto-Update feature” on page 34.
Table-Config *N*	The Table-Config subprofiles, numbered 1 to 8, each contain a single hostname field and a single IP address field.
Host-Name	Specifies a hostname, which must be unique within the table. For details, see “Hostname matching” next.
IP-Address	Specifies a valid IP address for the Host-Name, or the null address.

Hostname matching

The hostname specified in the DNS-Local-Table subprofile must start with an alphabetic character, and must be less than 256 characters. Trailing periods are ignored in the comparison.

The hostname may be a host name or a fully qualified name that includes the domain name. If it does not contain a domain name, and the following parameters are set to valid domain-name values in the IP-Global profile:

```
IP-GLOBAL
 domain-name = eng.abc.com
 sec-domain-name = abc.com
```

Then the system appends the specified domain name when looking up the hostname. For example, for a DNS query on the following hostname:

```
host-name = wheelers
```

The MAX TNT searches for the hostname as well as the following domain names:

```
wheelers.eng.abc.com
wheelers.abc.com
```

Defining the local table

Following is an example of configuring a local table specifying three hosts:

```
admin> read ip-global
IP-GLOBAL read

admin> list dns-local
enabled = no
auto-update = no
table-config = [ { "" 0.0.0.0 } {"" 0.0.0.0 } {"" 0.0.0.0 } {"" 0.0.0.0+

admin> set enabled = yes

admin> list table 1
host-name = ""
ip-address = 0.0.0.0

admin> set host = host1.abc.com

admin> set ip = 10.1.2.3

admin> list ..
table-config[1] = { host1.abc.com 10.1.2.3 }
table-config[2] = { "" 0.0.0.0 }
table-config[3] = { "" 0.0.0.0 }
table-config[4] = { "" 0.0.0.0 }
table-config[5] = { "" 0.0.0.0 }
table-config[6] = { "" 0.0.0.0 }
table-config[7] = { "" 0.0.0.0 }
table-config[8] = { "" 0.0.0.0 }

admin> set 2 host = host2.xyz.

admin> set 2 ip = 11.1.2.3

admin> set 3 host = localhost

admin> set 3 ip = 10.0.0.1

admin> write
IP-GLOBAL written
```

If you specify an IP address without also specifying a hostname, a message such as the following is displayed when you write the profile:

```
error: dns-local-table: host-name missing (#3 1.2.3.4)
```

If you enter an invalid hostname, a message such as the following is displayed when you write the profile:

```
error: dns-local-table: host-name must start with alpha char (#5 11foo)
```

Using the Auto-Update feature

The Auto-Update parameter determines whether the local table is updated by regular successful DNS queries. If it is set to No (the default), the contents of the local table are not affected by successful DNS queries. If set to Yes, when a regular DNS query succeeds, a lookup on that hostname is made to the local table. If there is an entry for the hostname, the entry's IP address(es) is replaced by the query response. Note that you can use the Auto-Update parameter to build the local table.

The following parameters, which are shown with their default values, affect how the table is updated when Auto-Update is set to Yes:

```
IP-GLOBAL
 dns-list-attempt = no
 dns-list-size = 6
```

If DNS-List-Attempt is set to No, a successful DNS query returns a single address for a given hostname. In the DNS table in RAM, that address is stored and the remaining 34 addresses are cleared (set to zero).

If DNS-List-Attempt is set to Yes, a successful DNS query returns the number of addresses it finds for the host, up to DNS-List-Size. In the DNS table in RAM, those addresses are stored, overwriting the configured address or the addresses retrieved from earlier DNS queries. To prevent stale entries in the table in RAM, addresses greater than DNS-List-Size are cleared at each update.

Note: If the administrator modifies the DNS-Local-Table subprofile, assigning a single address to a host, the newly configured address is propagated to the table in RAM. The first address of the hostname entry is overwritten with the configured address, and all remaining addresses are cleared. If Auto-Update is set to Yes, the next successful DNS query overwrites the configured address and restores the multiple addresses (up to DNS-List-Size).

Following is an example that configures 8 hostnames with null addresses and then sets Auto-Update to Yes. The DNS-Local-Table changes will be propagated to RAM, and successful DNS queries to the specified hostnames will build the table with up to 14 addresses for each of the hosts.

```
admin> read ip-global
IP-GLOBAL read

admin> set dns-list-attempt = yes

admin> set dns-list-size = 14

admin> list dns-local
enabled = no
auto-update = no
table-config = [ { "" 0.0.0.0 } {"" 0.0.0.0 } {"" 0.0.0.0 } {"" 0.0.0.0+

admin> set enabled = yes

admin> set auto-update = yes

admin> list table
table-config[1] = { "" 0.0.0.0 }
table-config[2] = { "" 0.0.0.0 }
table-config[3] = { "" 0.0.0.0 }
table-config[4] = { "" 0.0.0.0 }
table-config[5] = { "" 0.0.0.0 }
table-config[6] = { "" 0.0.0.0 }
table-config[7] = { "" 0.0.0.0 }
table-config[8] = { "" 0.0.0.0 }

admin> set 1 host = mercury

admin> set 2 host = venus

admin> set 3 host = earth

admin> set 4 host = mars

admin> set 5 host = jupiter

admin> set 6 host = saturn

admin> set 7 host = uranus
```

```
admin> set 8 host = neptune

admin> write
IP-GLOBAL written
```

New command to display the local table

A new DNSTAB system-level command is provided for displaying the local DNS table. The new command uses the following syntax:

```
dnstab -s [<entry number>]
```

where entry-number is an optional argument specifying a number from one to eight indicating a host entry in the local table.

In the following example, DNS-List-Size is set to 14, and 11 addresses were resolved for the system named *neptune* (entry #8 based on the sample table configuration shown in the previous section).

```
admin> nslookup neptune
Resolving host neptune.
IP address for host neptune is 11.65.212.211.
IP address for host neptune is 10.168.10.9.
IP address for host neptune is 10.168.6.14.
IP address for host neptune is 10.168.6.144.
IP address for host neptune is 11.65.212.9.
IP address for host neptune is 10.168.6.141.
IP address for host neptune is 10.168.6.143.
IP address for host neptune is 11.65.212.182.
IP address for host neptune is 10.168.6.145.
IP address for host neptune is 10.168.6.86.
IP address for host neptune is 11.65.212.23.

admin> dnstab -s 8

Local DNS Table: enabled, AutoUpdate: enabled.
Local DNS Table
   Name                          IP Address      # Reads Time of last read
   ____________________________ _______________ _______ _________________
8: "neptune"                     11.65.212.211 *       1  ---
   10.168.10.9       10.168.6.14       10.168.6.144       11.65.212.9
   10.168.6.141      10.168.6.143      11.65.212.182      10.168.6.145
   10.168.6.86       11.65.212.23      -------            -------
   -------
```

This sample output indicates that DNS-List-Size = 14. Eleven addresses are displayed, with the remaining addresses (up to the DNS-List-Size) displayed as a row of hyphens.

The first line of output indicates that Enabled and Auto-Update are both set to Yes in the IP-Global profile. The asterisk following the first IP address indicates that the entry has been updated automatically by a DNS query. The number 1 in the # Reads column indicates that the hostname has been referenced once. The Time of Last Read column shows the time the entry is used, if SNTP is in use. If SNTP is not in use, this column will contain a row of hyphens, as shown in the output above.

Parameters for handling directed broadcasts

Two parameters have been added to help the administrator defend against Denial of Service (DoS) attacks that use directed broadcast traffic. Following are the relevant parameters, which are shown with their default values:

```
IP-INTERFACE {{shelf-N slot-N N} N}
   directed-broadcast-allowed = yes

IP-GLOBAL
   icmp-reply-directed-bcast = yes
```

Directed-Broadcast-Allowed specifies whether the MAX TNT forwards directed broadcast traffic onto the interface and its network. If set to No, the system drops directed broadcast traffic, preventing it from propagating onto intermediary networks. To protect all of the LAN interfaces against DoS attacks that use directed broadcast traffic, you must set the Directed-Broadcast-Allowed parameter to No in all IP-Interface profiles.

ICMP-Reply-Directed-Bcast specifies whether the MAX TNT responds to directed-broadcast ICMP echo requests. If set to No, the system does not respond to any directed-broadcast ICMP requests. The setting of this parameter is also shown in a new Directed-Broadcast field in the output of the Ifmgr debug command.

Following is an example that sets these parameters to No:

```
admin> read ip-int {{1 c 1} 0}
IP-INTERFACE/{ { shelf-1 controller 1 } 0 } read

admin> set directed-broadcast-allowed = no

admin> write
IP-INTERFACE/{ { shelf-1 controller 1 } 0 } written

admin> read ip-global
IP-GLOBAL read

admin> set icmp-reply-directed-bcast = no

admin> write
IP-GLOBAL written
```

IPX routing for the MAX TNT

A MAX TNT configured for IPX routing enables NetWare clients and distributed Novell networks to use NetWare services across the WAN. Figure 2 shows a MAX TNT that routes IPX between WAN interfaces (connections) and a local Novell network.

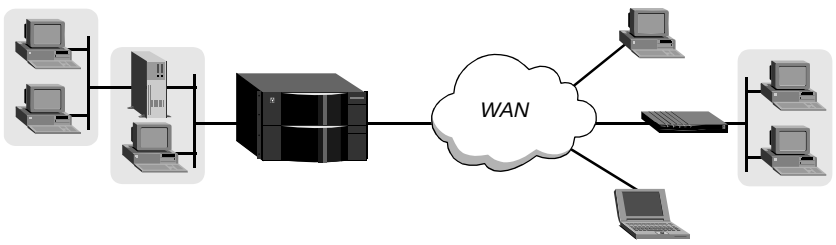

Figure 2. Routing IPX between LAN and WAN interfaces

Note: The NetWare version must be 3.11 or newer.

IPX routing on the WAN

Ascend has optimized IPX routing for the WAN, which required some modifications of standard IPX behavior as well as IPX extensions to enable the MAX TNT to operate as clients expect for NetWare LANs. This section discusses issues related to scaling LAN protocols to the WAN.

How Ascend units use IPX SAP

The MAX TNT follows standard IPX SAP behavior for routers when connecting to non-Ascend units across the WAN. However, when it connects to another Ascend unit configured for IPX routing, both ends of the connection exchange their entire SAP tables, so all remote services are immediately added to the MAX TNT unit's SAP table and vice versa.

When a NetWare client sends a SAP request to locate a service, the MAX TNT consults its SAP table and replies with its own hardware address and the internal network address of the requested server. This is analogous to proxy ARP in an IP environment. The client can then transmit packets whose destination address is the internal address of the server. When the MAX TNT receives those packets, it consults its RIP table. If it finds an entry for that destination address, it brings up the connection (unless it is already up) and forwards the packet.

How Ascend units use IPX RIP

The MAX TNT follows standard IPX RIP behavior for routers when connecting to non-Ascend units. However, when it connects to another Ascend unit configured for IPX routing, both ends of the connection immediately exchange their entire RIP tables. In addition, the MAX TNT maintains those RIP entries as static until the unit is reset or power cycled.

How IPX RIP works

IPX RIP is similar to the routing information protocol in the TCP/IP protocol suite, but it is a different protocol. IPX routers broadcast RIP updates periodically and when a WAN connection is established. The MAX TNT receives IPX RIP broadcasts from a remote device, adds 1 to the hop count of each advertised route, updates its own RIP table, and broadcasts updated RIP packets on connected networks in a split-horizon fashion.

The IPX RIP default route

The MAX TNT recognizes network number –2 (0xFFFFFFFE) as the IPX RIP default route. When it receives a packet for an unknown destination, the MAX TNT forwards the packet to the IPX router advertising the default route. If more than one IPX router is advertising the default route, the unit makes a routing decision based on Hop and Tick count. For example, if the MAX TNT receives an IPX packet destined for network 77777777 and it does not have a RIP table entry for that destination, the MAX TNT forwards the packet towards network number FFFFFFFE, if available, instead of simply dropping the packet.

Support for IPXWAN negotiation

The MAX TNT supports the IPXWAN protocol, which is essential for communicating with Novell software (such as NetWare Connect2) that supports dial-in connections, and with the

Multi-Protocol Router. For full specifications of the IPXWAN protocol, see RFC 1634 and *NetWare Link Services Protocol Specification—IPX WAN Version 2*.

When an IPX connection is brought up between two Ascend units, all options are negotiated during the IPXCP phase. IPXWAN negotiation never takes place between two Ascend units, because neither unit initiates the negotiation process by sending out an IPXWAN Timer_Request packet.

Connections with non-Ascend devices that use Novell software operating over PPP do not negotiate options during the IPXCP phase, so all options are negotiated during the IPXWAN phase of link establishment. The far-end device sends an IPXWAN Timer_Request packet, which triggers IPXWAN negotiation in the MAX TNT. The devices compare internal network numbers and assign the slave role to the unit with the lower number. The other unit becomes the master of this link for the duration of the IPXWAN negotiation. The slave unit returns an IPXWAN Timer_Response packet, and the master unit initiates an exchange of information about the final router configuration. The MAX TNT supports the following routing options:

- Ascend Routing (Unnumbered RIP/SAP without aging).
- Novell Routing (Unnumbered RIP/SAP with aging).
- None (The peer is a dial-in client. No RIP/SAP except on request and we may assign Net and Node Numbers.)

Header compression is rejected as a routing option. After IPXWAN negotiation is completed, transmission of IPX packets begins, using the negotiated routing option.

Extensions to standard IPX

NetWare uses dynamic routing and service location, so clients expect to be able to locate a server dynamically, regardless of where it is physically located. Because this scheme was designed to work in a LAN environment, not for WAN operations, Ascend provides extensions to standard IPX. The added features enhance WAN functionality, as shown in Table 1

Table 1. Ascend extensions to IPX

Ascend extension	Purpose
Virtual network for dial-in clients	To enable the MAX TNT to route IPX to non-routers (NetWare clients), it supports a virtual IPX network defined in the MAX TNT unit's IPX-Global profile. The unit can then assign a unique network address to the client. The client's connection must specify that it is a Dialin-Peer.
Accepting or rejecting RIP and SAP updates	The MAX TNT can transmit RIP and SAP updates, receive them, or both, or you can disable RIP or SAP updates for any IPX routing connection.

Table 1. Ascend extensions to IPX

Ascend extension	Purpose
Bringing up connections in response to a SAP query	The Dial-Query feature is designed for sites that support many clients and connections to only a few remote IPX networks. The MAX TNT brings up all connections that enable Dial-Query when it receives a SAP query for a file server (service type 0x04) and its SAP table has no entry for that service type.
Static routes to servers	Even though the MAX TNT learns its routes via RIP, it clears the entire RIP table when reset or powered down. Some sites configure a static IPX route to enable the MAX TNT to open a connection to that location and download the RIP table when the unit is powered up.
SAP filters	IPX SAP filters enable you to prevent the SAP table from becoming too large by explicitly including or excluding servers, services, or service types on any interface.

Recommendations for NetWare client software

NetWare clients on a WAN do not need special configuration in most cases. However, if the local network supports NetWare servers, you should configure NetWare clients with a preferred server on the local network, not at a remote site. If the local network does not support NetWare servers, configure local clients with a preferred server that is on the network with the lowest connection costs. For more information, see the NetWare documentation.

Due to possible performance issues, executing programs remotely is not recommended. For best results, put LOGIN.EXE on each client's local drive.

Both Macintosh and UNIX clients can use IPX to communicate with servers. However, both types of clients also support native protocols: AppleTalk (Macintosh) or TCP/IP (UNIX). If Macintosh clients must access NetWare servers across the WAN by using AppleTalk software (rather than MacIPX), the MAX TNT must support AppleTalk routing. Otherwise, AppleTalk packets will not make it across the connection. If UNIX clients access NetWare servers via TCP/IP (rather than UNIXWare), the MAX TNT must also be configured as an IP router. Otherwise, TCP/IP packets will not make it across the connection.

Note: Packet Burst lets servers send a data stream across the WAN before a client sends an acknowledgment. The feature is included automatically in server and client software for NetWare 3.12 or later. If local servers are running NetWare 3.11, they should have PBURST.NLM loaded. For more information, see your NetWare documentation.

IPX-Global profile settings

To create IPX-Interface profiles for routing on the MAX TNT LAN interfaces, you must enable IPX routing in the IPX-Global profile. In addition, to support dial-in NetWare clients

that are not routers, you must configure a virtual IPX network to be used for assigning IPX addresses to those clients. Following are the relevant parameters, shown with their default settings:

```
IPX-GLOBAL
   ipx-routing-enabled = yes
   ipx-dialin-pool = 12:34:56:78
```

Enabling IPX routing mode

The IPX-Routing-Enabled parameter enables IPX routing. When you write the profile, the MAX TNT comes up in IPX routing mode. At that time, it creates an IPX-Interface profile for each installed Ethernet port.

Defining a virtual IPX network for dial-in clients

When a NetWare client dials in, the MAX TNT negotiates a routing session with the client by assigning the client a node address on the virtual IPX network. The client must accept the network number defined in this pool. If it has its own node number, the MAX TNT uses that number to form the full network:node address. If the client does not have a node number, the MAX TNT assigns it a unique node address on the virtual network.

The IPX network number you assign must be unique within the entire IPX routing domain of the MAX TNT. The MAX TNT advertises the route to this virtual IPX network.

Example of an IPX-Global configuration

Following is an example of how to enable IPX routing mode and define a network for address assignment to dial-in clients that are not routers:

```
admin> read ipx-global
IPX-GLOBAL read
admin> set ipx-routing-enabled = yes
admin> set ipx-dialin = cccc1234
admin> write
IPX-GLOBAL written
```

When you write the profile, the MAX TNT comes up in IPX routing mode and creates IPX-Interface profiles for each Ethernet interface. Be sure that the network number you assign to the IPX-Dialin parameter is unique in the MAX TNT routing domain.

IPX-Interface profile settings

After you enable IPX routing in the IPX-Global profile, the system creates an IPX-Interface profile for each Ethernet interface in the system. *IPX-Interface profiles do not exist until you enable IPX routing globally.*

The IPX-Interface profiles contain the following parameters, which are shown with their default settings:

```
IPX-INTERFACE/{ { shelf-1 slot-6 2 } 0 }
   interface-address* = { { shelf-1 slot-6 2 } 0 }
   ipx-routing-enabled = no
   ipx-frame = None
```

```
ipx-net-number = 00:00:00:00
ipx-type-20 = no
ipx-sap-filter-name = ""
```

Note: IPX-Routing-Enabled must be set to Yes, and a valid IPX-Frame type must be specified, in the IPX-Interface profile for the shelf-controller Ethernet port.

Enabling IPX routing and spoofing on the interface

To enable the MAX TNT to route IPX on an Ethernet interface, you must set both the IPX-Routing-Enabled parameter and the IPX-Frame parameter. The IPX-Frame parameter specifies which IPX frame type the MAX TNT will route and spoof.

Note: The MAX TNT routes and spoofs only one IPX frame type. If some NetWare software transmits IPX in a frame type other than the type you specify, the MAX TNT drops those packets. If you are not familiar with the concept of packet frames, see the Novell documentation.

To see which frame type to use on a LAN interface, go to a NetWare server's console on that segment and type LOAD INSTALL to view the AUTOEXEC.NCF file. Following is an example AUTOEXEC.NCF line that specifies 802.3 frames:

```
Load 3c509 name=ipx-card frame=ETHERNET_8023
```

Assigning an IPX network number

If there are other NetWare routers (servers) on the LAN interface, the IPX number assigned to the MAX TNT for that interface must be consistent with the number in use by the other routers. The best way to ensure this is to leave the default null address in the IPX-Net-Number parameter. The null address causes the MAX TNT to learn its network number from another router on the interface, or from the RIP packets received from the local IPX server.

If you enter an IPX network number other than zero, the MAX TNT becomes a seed router, and other routers can learn their IPX network number from the MAX TNT. For details about seed routers, see the Novell documentation.

Propagating IPX type 20 packets on a LAN interface

Some applications, such as NetBIOS over IPX, use IPX Type 20 packets to broadcast names over a network. By default, these broadcasts are not propagated over routed links (as Novell recommends) and are not forwarded over links that have less than 1 Mbps throughput. However, if you are using an application such as NetBIOS over IPX, which requires these packets in order to operate, you can enable the router to propagate IPX Type 20 packets over a LAN interface by setting the IPX-Type-20 parameter to Yes.

Applying a SAP filter to the LAN interface

You can apply a SAP filter to a local interface by specifying the filter profile name as the value of the IPX-SAP-Filter-Name parameter. When applied to a LAN interface, a SAP filter includes or excludes specific services from the MAX TNT unit's SAP table and its responses to SAP queries on the interface. If the directory services feature is not supported, servers or services that are not in the MAX TNT table will be inaccessible to clients across the WAN. A filter applied to a LAN interface takes effect immediately.

For information about defining a SAP filter, see “IPX-SAP-Filter profile settings” on page 50. For an example that shows how to apply the filter, see “Example of applying an IPX SAP filter to a LAN interface” on page 51.

Example of an IPX-Interface configuration

Following is an example of input that enables the MAX TNT to route 802.3 IPX frames to and from the LAN interface and propagate IPX Type 20 packets:

```
admin> read ipx-int { {1 12 2 } 0 }
IPX-INTERFACE/{ { shelf-1 slot-12 2 } 0 } read

admin> set ipx-routing-enabled = yes

admin> set ipx-frame = 802.3

admin> set ipx-type-20 = yes

admin> write
IPX-INTERFACE/{ { shelf-1 slot-12 2 } 0 } written
```

Note that this example does not specify an IPX-Net-Number, which means the MAX TNT is a nonseed router that will learn its address from another IPX router on the network or from the RIP packets received from the local IPX server.

Answer-Defaults profile settings

You must set Enabled to Yes in the IPX-Answer subprofile of the Answer-Defaults profile to allow the MAX TNT to answer incoming IPX routing calls. In addition, because the MAX TNT does not have a built-in authentication mechanism (such as matching addresses) for IPX connections, they require PPP authentication.

Following are the relevant parameters, shown with their sample settings:

```
ANSWER-DEFAULTS
   ipx-answer
      enabled = yes
   ppp-answer
       receive-auth-mode = any-ppp-auth
```

Connection profile settings

Connection profiles for IPX routing connections typically use PPP authentication (described in the current *MAX TNT Network Configuration Guide*). In addition, they specify one or more of the following IPX options, which are shown with their default values:

```
CONNECTION station
   ipx-options
      ipx-routing-enabled = no
      peer-mode = router-peer
      rip = both
      sap = both
      dial-query = no
      net-number = 00:00:00:00
      net-alias = 00:00:00:00
      sap-filter = ""
      ipx-sap-hs-proxy = no
```

```
ipx-sap-hs-proxy-net = [ 0 0 0 0 0 0 ]
ipx-header-compression = no
```

Enabling IPX routing on a WAN interface

To enable IPX routing for this connection, set IPX-Routing-Enabled to Yes. If it is set to No, the MAX TNT will not route IPX packets on this interface.

Specifying whether the remote device is a router or dial-in client

The Peer-Mode parameter specifies whether the remote site is a dial-in NetWare client or another IPX router. When Peer-Mode is set to Dialin-Peer, the MAX TNT negotiates a routing session with the dial-in NetWare client by assigning the client a node address on the virtual IPX network defined in the IPX-Global profile. The client must accept the network number that is assigned. If the client has its own node number, the MAX TNT uses that number to form the full network address. If it does not have a node number, the MAX TNT assigns it a unique node address on the virtual network.

Note: When connecting to a Dialin-Peer, the MAX TNT does not send RIP and SAP advertisements across the connection, and it ignores RIP and SAP advertisements received from the far end. However, it does respond to RIP and SAP queries received from dial-in clients.

Controlling RIP and SAP updates to and from the remote router

When the remote end of the connection is a router (Peer-Mode), you can specify how RIP and SAP packets are handled across this WAN connection. Both parameters are set to Both by default, which means that the MAX TNT both sends updates across the WAN connection (informing other routers on the remote network of its routes or services), and receives updates from the remote router (including those routes or services in its RIP or SAP table).

You can set the RIP parameter to Send to cause the MAX TNT to send its routes to the remote router, but not to receive any updates on this interface. Or, you can set it to Recv to receive updates from the remote router, but not propagate the local IPX routes to the remote site. If you set it to Off, no routes are propagated in either direction.

The same settings apply to the SAP parameter. If SAP is set to both send and receive broadcasts on the WAN interface, the MAX TNT broadcasts its entire SAP table to the remote network and listens for SAP table updates from that network. Eventually, both networks have a full table of all services on the WAN. To control which services are advertised and where, you can disable the exchange of SAP broadcasts across a WAN connection, or specify that the MAX TNT will only send or only receive SAP broadcasts on that connection.

When to use net-number and net-alias

The Net-Number specifies the IPX network number of the remote-end router. This parameter, which is rarely needed, accommodates those remote-end routers that require the MAX TNT to know that router's network number before connecting.

The Net-Alias parameter may specify a second IPX network number, to be used only when connecting to non-Ascend routers that use numbered interfaces.

Applying a SAP filter to a WAN interface

You can apply a SAP filter to a WAN interface by specifying the filter profile name as the value of the SAP-Filter parameter. When applied to a WAN interface, a SAP filter includes or excludes specific services from the MAX TNT unit's SAP table and its responses to SAP queries on the interface. A filter applied to a WAN interface takes effect when the connection next becomes active.

For information about defining a SAP filter, see "IPX-SAP-Filter profile settings" on page 50. For an example that shows how to apply the filter, see "Example of applying an IPX SAP filter to a LAN interface" on page 51.

Using dial-query

Dial-Query configures the MAX TNT to bring up a connection when it receives a SAP query for service type 0x04 (File Server) and that service type is not present in the MAX TNT SAP table. If the MAX TNT has no SAP table entry for service type 0x04, it brings up every connection that has Dial Query set. For example, if 20 Connection profiles have Dial-Query set, the MAX TNT brings up all 20 connections in response to the query.

Note: If the MAX TNT unit has a static IPX route for even one remote server, it chooses to bring up that connection as opposed to the more costly solution of bringing up every connection that has Dial-Query set.

Home server proxy

For mobile NetWare clients, you can specify the network number of from one to six NetWare servers that should receive SAP queries across this connection. Without this feature, when the client is in a distant location and sends a Get Nearest Server Request query, the client receives responses from servers closer to that location, rather than the expected home server or servers. With the home-server proxy feature, mobile clients can bring up a connection to the server or servers they usually use.

To enable the home-server proxy, set the IPX-SAP-HS-Proxy parameter to Yes, and configure the IPX-SAP-HS-Proxy-Net parameter with from one to six IPX network numbers. The MAX TNT then directs the client's SAP queries only to the specified networks.

Using IPX header compression

The IPX-Header-Compression parameter specifies whether or not the MAX TNT should use IPX header compression on this connection if the specified encapsulation method supports it.

Example of a connection between two Novell LANs

Figure 3 shows a MAX TNT providing a connection between an IPX network, which supports both servers and clients, and a remote site that also supports both servers and clients, and an Ascend unit.

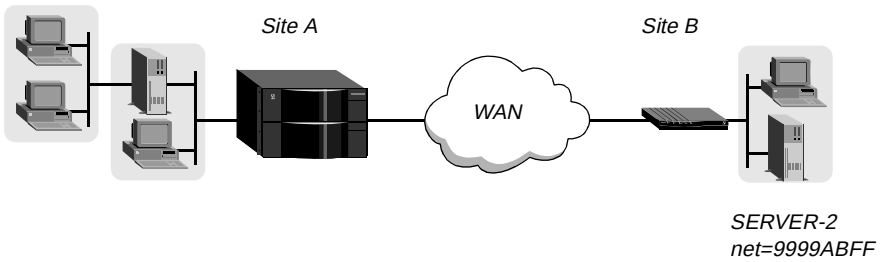

Figure 3. A connection with NetWare servers on both sides

In this example, the NetWare server at site B is configured with the following specifications:

```
Name=SERVER-2
internal net 013DE888
Load 3c509 name=net-card frame=ETHERNET_8023
Bind ipx net-card net=9999ABFF
```

Following is an example of specifying a connection to the Ascend unit at Site B:

```
admin> new conn sitebgw
CONNECTION/sitebgw read

admin> set active = yes

admin> set ppp recv-password = sitebpw

admin> list ipx
ipx-routing-enabled = no
peer-mode = router-peer
rip = both
sap = both
dial-query = no
net-number = 00:00:00:00
net-alias = 00:00:00:00
sap-filter = ""
ipx-sap-hs-proxy = no
ipx-sap-hs-proxy-net = [ 0 0 0 0 0 0 ]
ipx-header-compression = no

admin> set peer = router

admin> set rip = off

admin> write
CONNECTION/sitebgw written
```

Because RIP is turned off, you might want to create a static route to the server at the remote site, to ensure that the MAX TNT can bring up this connection, even immediately after a system reset. The following example shows how to configure a route to Server-2 at Site B:

```
admin> new ipx-route SERVER-2
IPX-ROUTE/SERVER-2 read

admin> set server-type = 0004

admin> set dest-network = 013DE888

admin> set server-node = 000000000001

admin> set server-socket = 0451

admin> set profile-name = sitebgw
```

```
admin> write
IPX-ROUTE/SERVER-2 written
```

Note: The destination network number is the server's internal network number. For more information about IPX routes, see "IPX-Route profile settings" on page 48.

Example of a connection to a dial-in client

Figure 4 shows a NetWare client dialing into a corporate IPX network using PPP software.

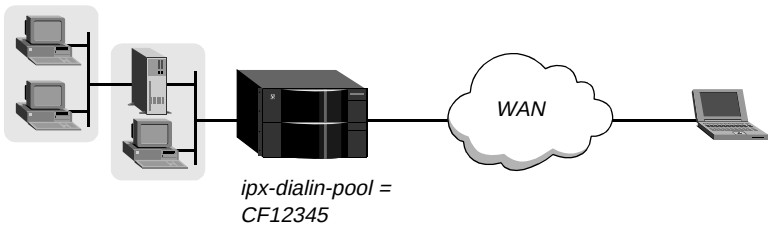

Figure 4. A dial-in NetWare client

Dial-in NetWare clients do not have an IPX network address. To have an IPX routing connection to the local network, the clients must dial in using PPP software, and the Connection profile must set Peer-Mode to Dialin-Peer. In addition, the MAX TNT must have a virtual IPX network defined for assignment to these clients. For information about defining a virtual IPX network, see "IPX-Global profile settings" on page 40.

Following is an example of input that configures an IPX routing connection for the client shown in Figure 4:

```
admin> new conn client
CONNECTION/client read

admin> set ppp recv-password = client-pw

admin> list ipx
ipx-routing-enabled = no
peer-mode = router-peer
rip = both
sap = both
dial-query = no
net-number = 00:00:00:00
net-alias = 00:00:00:00
sap-filter = ""
ipx-sap-hs-proxy = no
ipx-sap-hs-proxy-net = [ 0 0 0 0 0 0 ]
ipx-header-compression = no

admin> set ipx-routing = yes

admin> set peer = dialin

admin> write
CONNECTION/client written
```

Example of enabling home-server proxy

Following is an example of how to enable the home-server proxy feature in an IPX-routing Connection profile:

```
admin> read conn ipxclient
CONNECTION/ipxclient read

admin> list ipx
ipx-routing-enabled = no
peer-mode = router-peer
rip = both
sap = both
dial-query = no
net-number = 00:00:00:00
net-alias = 00:00:00:00
sap-filter = ""
ipx-sap-hs-proxy = no
ipx-sap-hs-proxy-net = [ 0 0 0 0 0 0 ]
ipx-header-compression = no

admin> set ipx-sap-hs-proxy = yes

admin> set ipx-sap-hs-proxy-net 1 = ccff1234

admin> write
CONNECTION/ipxclient written
```

Setting IPX-SAP-HS-Proxy to Yes enables the feature. You must then specify at least one (and up to six) IPX network addresses to which SAP broadcasts will be directed.

IPX-Route profile settings

When the MAX TNT is reset or power cycled, it clears its RIP and SAP tables from memory. Static routes create entries in new RIP and SAP tables as the unit initializes. The static routes enable the MAX TNT to reach a NetWare server and download more complete tables from there.

In the case where a MAX TNT is connecting to another Ascend unit, you might choose not to configure any static routes. However, that means that after a power-cycle or reset, you must dial the initial IPX routing connection manually. After that connection is established, the MAX TNT downloads the RIP table from the other Ascend unit and maintains the routes as static until its next power-cycle or reset.

The disadvantage of static routes is that they require manual updating whenever the specified server is removed or has an address change. Their advantages are that they ensure that the MAX TNT can bring up the connection in response to clients' SAP requests, and they help to prevent timeouts when a client takes a long time to locate a server on the WAN.

Note: You do not need to create IPX routes to servers that are on the local Ethernet.

Static IPX routes use the following parameters, which are shown with their default settings:

```
IPX-ROUTE name
   name* = name
   server-type = 00:00
   dest-network = 00:00:00:00
   server-node = 00:00:00:00:00:00
   server-socket = 00:00
   hops = 8
   ticks = 12
   profile-name = ""
   active-route = yes
```

Identifying the target

The service type is a number included in SAP advertisements. For example, NetWare file servers are SAP Service type 0x04.

The destination of an IPX route is the internal network of a server. For example, NetWare file servers are assigned an internal IPX network number by the network administrator and typically use the default node address of 000000000001. This is the destination network address for file read/write requests. (If you are not familiar with internal network numbers, see your NetWare documentation for details.)

Typically, Novell file servers use socket 0x451. The number you specify must be a well-known socket number. Services that use dynamic socket numbers may use a different socket each time they load, and will not work with IPX Route profiles. To bring up a connection to a remote service that uses a dynamic socket number, specify a master server with a well-known socket number on the remote network.

Specifying how to get to the server's network

To reach the remote server's network, the default hop count of 2 and tick count of 12 are usually appropriate, but you might need to increase these values for very distant servers. Ticks are IBM PC clock ticks (1/18 second). Note that best routes are calculated on the basis of on tick count, not hop count.

The Profile-Name parameter specifies the Connection profile to use. When the MAX TNT receives a query for the specified server or a packet addressed to that server, it finds the referenced Connection profile and dials the connection.

Activating the route

For the MAX TNT to use this route, the Active-Route parameter must be set to Yes.

Example of a static IPX route

The following example shows how to create a new IPX-Route profile for a remote server named Server-1.

```
admin> new ipx-route Server-1
IPX-ROUTE/Server-1 read
admin> set server-type = 0004
admin> set dest-network = cc1234ff
admin> set server-node 1 = 000000000001
admin> set server-socket = 0451
admin> set profile-name = sitebgw
admin> write
IPX-ROUTE/Server-1 read
```

IPX-SAP-Filter profile settings

IPX SAP filters include or exclude services from the MAX TNT SAP table. You can prevent the MAX TNT from sending its SAP table or receiving a remote site's SAP table by turning off IPX SAP in a Connection profile.

Each filter contains up to eight Input filters and Output filters, numbered from 1 to 8, which are defined individually and applied in order (1–8) to the packet stream.

The MAX TNT applies input filters to all SAP packets received by the MAX TNT. Input filters screen advertised services and exclude them from (or include them in) the MAX TNT service table as specified by the filter conditions.

The MAX TNT applies output filters to SAP response packets it transmits. If it receives a SAP request packet, the MAX TNT applies output filters before transmitting the SAP response, and excludes services from (or includes them in) the response packet as specified by the filter conditions.

Following are the subprofiles and parameters used to define a SAP filter, shown with their default values:

```
IPX-SAP-FILTER ipx-sap-filter-name
   ipx-sap-filter-name* = ipx-sap-filter-name
   input-ipx-sap-filters
      input-ipx-sap-filters [1–8]
         valid-filter = no
         type-filter = exclude
         server-type = 00:00
         server-name = ""
   output-ipx-sap-filters
      output-ipx-sap-filters [1–8]
         valid-filter = no
         type-filter = exclude
         server-type = 00:00
         server-name = ""
```

Each of the eight input and output filters include the same parameters.

Activating an input or output filter for use

The Valid-Filter parameter enables the input or output filter for use. If it is set to No, the MAX TNT skips the filter when it applies the entire IPX SAP filter to SAP data.

Specifying the action to take

The Type-Filter parameter specifies whether this filter will explicitly include the service in the SAP table or SAP response packets, or will explicitly exclude the service. The Include setting is typically used to include a specific service when previous input or output filters have excluded a general type of service. Exclude indicates a specific service to filter out of the SAP table or SAP response packets.

Identifying the service to be filtered

Server-Type specifies a hexadecimal number representing a type of NetWare service. For example, the number for file services is 0x04. In an output filter, the Server-Type parameter

specifies whether to include or exclude advertisements for this service type in SAP response packets. In an Input filter, the Server-Type parameter specifies whether to include or exclude services of this type in the MAX TNT service table.

If you specify a Server-Name, you can specify a local or remote NetWare server. If the server is on the local network and you are specifying an output filter, the Server-Name parameter specifies whether to include or exclude advertisements for this server in SAP response packets. If the server is on the remote IPX network and you are specifying an input filter, the Server-Name parameter specifies whether to include or exclude this server in the MAX TNT service table.

Example of defining an IPX SAP filter

For background information about SAP filters, see “IPX-SAP-Filter profile settings” on page 50. The following example shows how to create a SAP filter that prevents local NetWare users from accessing a remote NetWare server, by excluding it from the MAX TNT SAP table:

```
admin> new ipx-sap-filter server_1
IPX-SAP-FILTER/server_1 read

admin> list
ipx-sap-filter-name* = no-server1
input-ipx-sap-filters = [ { no exclude 00:00 "" } { no exclude 00:00 ""+
output-ipx-sap-filters = [ { no exclude 00:00 "" } { no exclude 00:00 "+

admin> list input 1
valid-filter = no
type-filter = exclude
server-type = 00:00
server-name = ""

admin> set valid-filter = yes

admin> set server-type = 0409

admin> set server-name = server_1

admin> list
valid-filter = yes
type-filter = exclude
server-type = 04:09
server-name = server_1

admin> write
IPX-SAP-FILTER/server_1 written
```

Example of applying an IPX SAP filter to a LAN interface

Following is an example of applying the SAP filter created in the previous section:

```
admin> read ipx-interface { {1 12 2 } 0 }
IPX-INTERFACE/{ { shelf-1 slot-12 2 } 0 } read

admin> set ipx-sap-filter-name = server_1

admin> write
IPX-INTERFACE/{ { shelf-1 slot-12 2 } 0 } written
```

For background information, see “IPX-Interface profile settings” on page 41.

Example of applying an IPX SAP filter to a WAN interface

Following is an example of applying the SAP filter created in the previous section:

```
admin> read conn client
CONNECTION/client read

admin> list ipx
ipx-routing-enabled = yes
peer-mode = dialin-peer
rip = both
sap = both
dial-query = no
net-number = 00:00:00:00
net-alias = 00:00:00:00
sap-filter = ""
ipx-sap-hs-proxy = no
ipx-sap-hs-proxy-net = [ 0 0 0 0 0 0 ]
ipx-header-compression = no

admin> set sap-filter = server_1

admin> write
CONNECTION/client written
```

AppleTalk routing and remote access

A MAX TNT configured for AppleTalk routing enables dial-in connections from AppleTalk Remote Access (ARA) client software, PPP dial-in software that supports AppleTalk, and AppleTalk-enabled Ascend units. Figure 5 shows a MAX TNT that routes AppleTalk between WAN interfaces (connections) and a local AppleTalk interface.

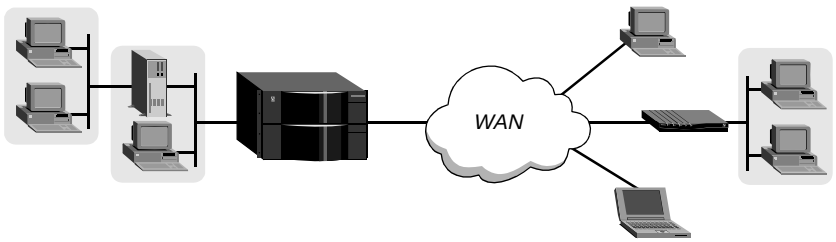

Figure 5. Routing AppleTalk between LAN and WAN interfaces

Note: AppleTalk routing be enabled on the shelf-controller to enable the system to forward AppleTalk packets from a host card to the shelf-controller. This is required for any kind of AppleTalk connection, even if the individual Connection profile to a remote device does not use routing.

Atalk-Global profile settings

When an ARA or AppleTalk PPP client dials in, the MAX TNT assigns the client an AppleTalk address on a virtual AppleTalk network. You define the virtual AppleTalk network in the Atalk-Global profile by setting the following parameters, which are shown with sample settings:

```
ATALK-GLOBAL
   atalk-dialin-pool-start = 100
   atalk-dialin-pool-end = 200
```

AppleTalk networks are assigned a network range, which is a contiguous range of integers between 1 and 65,199. Each network range must be unique, No two networks can use the same range, and no two network ranges can overlap.

Each number in the range can be associated with up to 253 nodes, so the range determines how many clients can dial in. For example, a network with a range 1000-1002 could support up to 2 x 253, or 506 clients. Following is an example of defining a virtual network. In this case, the network range is 1000–1002:

```
admin> read atalk-global
ATALK-GLOBAL read
admin> set atalk-dialin-pool-start = 1000
admin> set atalk-dialin-pool-end = 1002
admin> write
ATALK-GLOBAL written
```

Atalk-Interface profile settings

In the Atalk-Interface profile, you enable AppleTalk routing and specify whether the MAX TNT will operate as a seed or nonseed router on the interface. In this release, only the built-in Ethernet interface on the shelf-controller can be configured an AppleTalk interface. The Atalk-Interface profile contains the following parameters, which are shown with sample settings:

```
ATALK-INTERFACE { { shelf-1 controller 1 } 0 }
   interface-address* = { { shelf-1 controller 1 } 0 }
   atalk-routing-enabled = yes
   hint-zone = "SLC Engineering"
   atalk-Router = atlk-router-seed
   atalk-Net-Start = 1001
   atalk-Net-End = 1010
   atalk-Default-Zone = "SLC Engineering"
   atalk-Zone-List = [ "SLC Engineering" "SLC Test 1" "SLC Test
```

Configuring a seed router

A seed router has its own hard-coded network and zone configuration. You configure the MAX TNT as a seed AppleTalk router on a LAN interface by using the following parameters:

Parameter	Effect
Atalk-Routing-Enabled	Enables the MAX TNT to route AppleTalk on the shelf-controller Ethernet interface. If set to No, none of the other AppleTalk parameters applies.

Parameter	Effect
Atalk-Router	Specifies a routing mode. If set to Atlk-Router-Off, none of the remaining parameters applies. If set to Atlk-Router-Seed, the unit comes up with the specified zone and network configuration. If there are other AppleTalk routers on the LAN interface, you must be sure that the zone and network information you specify is completely consistent with the corresponding specifications for those routers.
Atalk-Net-Start Atalk-Net-End	Specify the network range for the shelf-controller Ethernet interface. The network range is a contiguous range of integers between 1 and 65,199. Each range must be unique. No two interfaces may use the same range, and no two network ranges may overlap. Each number in the range can be associated with up to 253 nodes, so the range determines how many clients the interface can support. For example, an interface with the range 1005-1010 could support up to 5 x 253, or 1265 clients.
Default-Zone	Specifies the default AppleTalk zone for the shelf-controller Ethernet interface. The default zone is the zone assigned to an AppleTalk service on this interface if the service does not select a zone in which to reside.
Zone-List	Specifies the zone list for the shelf-controller Ethernet interface. The zone list is a list of 1 to 32 AppleTalk zone names. Each name consists of from 1 to 33 characters, including embedded spaces. The characters must be in the standard printing character set, and must not include an asterisk (*).

In the following example, the MAX TNT is configured as a seed router on the shelf-controller Ethernet interface. The sample configuration defines the network range 1005–1010, three zones, and the default zone for the LAN interface.

```
admin> read atalk-int {{1 c 1} 0}
ATALK-INTERFACE/ { { shelf-1 controller 1 } 0 } read

admin> set atalk-routing = yes

admin> set atalk-router = atlk-router-seed

admin> set atalk-net-start = 1005

admin> set atalk-net-end = 1010

admin> set atalk-default-zone = engineering

admin> set atalk-zone-list 1 = admin

admin> set atalk-zone-list 2 = test

admin> set atalk-zone-list 2 = engineering

admin> write
ATALK-INTERFACE/ { { shelf-1 controller 1 } 0 } written
```

Configuring a nonseed router

A nonseed router acquires its network and zone configuration from another router on the network. You configure the MAX TNT as a nonseed AppleTalk router on a LAN interface by using the following parameters:

Parameter	Effect
Atalk-Routing-Enabled	Enables the MAX TNT to route AppleTalk on the shelf-controller Ethernet interface. If set to No, none of the other AppleTalk parameters applies.
Atalk-Router	Specifies a routing mode. If set to Atlk-Router-Off, none of the remaining parameters applies. If set to Atlk-Router-Nonseed, the unit learns its zone and network configuration from another AppleTalk router (a seed router) on the network. If the MAX TNT is configured in nonseed mode, a seed router must be available at startup time, or the MAX TNT cannot come up in AppleTalk routing mode. (If the MAX TNT comes up without AppleTalk routing enabled because no seed routers were available at startup, you must reset the system after a seed router is up.)
Hint-Zone	Specifies the zone in which the MAX TNT resides. The MAX TNT can include the specified zone name in the ZipGetNetInfo request packet it sends out to get its configuration from a seed router, and the router can return a valid network range for that zone.

In the following example, the MAX TNT is configured as a nonseed router:

```
admin> read atalk-int {{1 c 1} 0}
ATALK-INTERFACE/ { { shelf-1 controller 1 } 0 } read

admin> set atalk-routing = yes

admin> set atalk-router = atlk-router-non-seed

admin> set hint-zone = engineering

admin> write
ATALK-INTERFACE/ { { shelf-1 controller 1 } 0 } written
```

Answer-Defaults profile settings

To enable ARA client connections, you must enable ARA-Answer in the Answer-Defaults profile. In addition, if you intend to allow ARA Guest access set the Profiles-Required parameter to No (it is typically set to Yes for security purposes). These are the relevant parameters:

```
ANSWER-DEFAULTS
   profiles-required = no
   ara-answer
      enabled = yes
```

Following is an example of input that enables ARA-Answer and disables ARA Guest access:

```
admin> read answer
ANSWER-DEFAULTS read

admin> set ara-answer enabled = yes
```

```
admin> set profiles-required = yes

admin> write
ANSWER-DEFAULTS written
```

(Setting Profiles-Required to Yes disables ARA Guest access.)

Connection profile settings

PPP and ARA are the encapsulation protocols used for AppleTalk dialin on the MAX TNT. AppleTalk PPP and ARA Client software are available from Apple Computer (both ARA and PPP are supported in ARA 3.0) and from other vendors such as Netmanage Pacer PPP. Both AppleTalk PPP and ARA can be used over a modem or V.120 ISDN TA connection. AppleTalk PPP can also be used over sync-PPP when the calling unit is an Ascend router (Pipeline or MAX series).

You configure ARA or AppleTalk PPP connections by using the following parameters, which are shown with sample settings:

```
CONNECTION station
   encapsulation-protocol = ara
   ara-options
      recv-password = test
      ara-enabled = yes
      maximum-connect-time = 0
   appletalk-options
      atalk-routing-enabled = no
      atalk-static-ZoneName = ""
      atalk-static-NetStart = 0
      atalk-static-NetEnd = 0
      atalk-Peer-Mode = router-peer
```

Note: AppleTalk routing must be enabled for incoming PPP connections, but it is not necessary for ARA client connections.

Using these parameters, there are three ways to configure a Connection profile for AppleTalk connectivity:

- ARA client connection
- PPP dialin connection
- Synchronous PPP connection with an Ascend router

Configuring an ARA client Connection profile

An ARA client connection uses the ARA encapsulation protocol, and does not require AppleTalk routing in the Connection profile. You configure an ARA client connection by using the following parameters:

Parameter	Effect
Encapsulation-Protocol	Must specify ARA for ARA client connections.
ARA-Enabled	In the ARA-Options subprofile, the ARA-Enabled parameter turns on ARA processing for the connection.
Recv-Password	Specifies the password sent to the MAX TNT by the ARA client.

Parameter	Effect
Maximum-Connect-Time	Specifies the maximum number of minutes an ARA session can remain connected. The default setting, 0 (zero) disables the timer. If you specify a maximum connect time, the MAX TNT initiates an ARA disconnect when that time is up. The ARA link goes down cleanly, but remote users are not notified. Users will find out the ARA link is gone only when they try to access a device.

In Figure 6, the dial-in client is running ARA 3.0, with ARA encapsulation selected and with an internal modem. In this example, the client will be assigned a network address on the virtual 1000–1002 network, and a maximum ARA connection time of 60 minutes.

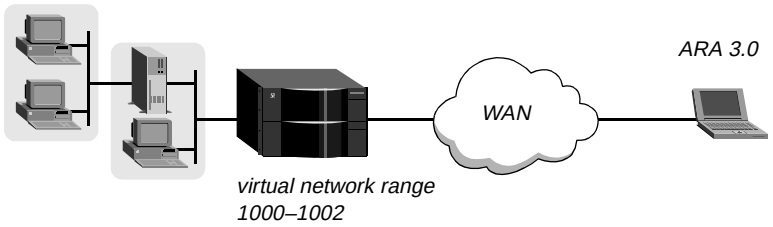

Figure 6. ARA Client dial-in

The administrator enters the following commands at the system prompt:

```
admin> read connection araclient
CONNECTION/araclient read
admin> set active = yes
admin> set encaps = ara
admin> set ara-enabled = yes
admin> set ara recv-password = ara-password
admin> set maximum-connect-time = 60
admin> write
CONNECTION/araclient written
```

Configuring a PPP dialin client connection for AppleTalk

An AppleTalk PPP dialin client connection uses the PPP encapsulation protocol. You configure an AppleTalk PPP client connection by using the following parameters:

Parameter	Effect
Encapsulation-Protocol	Must specify PPP for an AppleTalk PPP dialin client connection.
Recv-Password	In the PPP-Options subprofile, specifies the password sent to the MAX TNT by the PPP client.
Atalk-Routing-Enabled	In the AppleTalk-Options subprofile, enables AppleTalk routing for the connection. If AppleTalk routing has not been enabled in the Atalk-Interface profile, or if the Answer-Defaults profile does not enable ARA-Answer, this parameter has no effect.

Parameter	Effect
Atalk-Peer-Mode	In the Appletalk-Options subprofile, specifies whether the remote site is a dial-in client or another AppleTalk router. When set to Dialin-Peer, the MAX TNT negotiates a routing session with the dial-in client by assigning the client a node address on the virtual AppleTalk network defined in the Atalk-Global profile. The client must accept the network number assigned.

In Figure 7, the dial-in client is running ARA 3.0, and has selected PPP encapsulation, or is using another PPP dialer that supports AppleTalk. The client will be assigned a network address on the virtual 1000-1002 network.

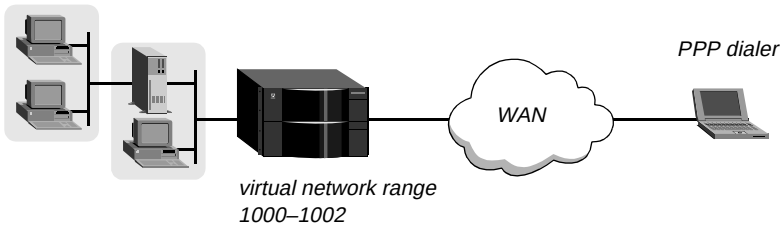

Figure 7. AppleTalk connection using a PPP dialer

The administrator might configure a Connection profile for the client as follows:

```
admin> new connection ppp-atalk
CONNECTION/ppp-atalk read
admin> set active = yes
admin> set encapsulation-protocol = ppp
admin> set ppp recv-password = localpw
admin> set appletalk atalk-routing-enabled = yes
admin> set appletalk atalk-peer-mode = dialin
admin> write
CONNECTION/ppp-atalk written
```

For details about other PPP settings, see the *MAX TNT Network Configuration Guide*.

Configuring an AppleTalk routing connection

An AppleTalk routing connection uses the PPP encapsulation protocol or one of its multilink variants (MP or MP+). You configure an AppleTalk routing connection by using the following parameters:

Parameter	Effect
Encapsulation-Protocol	Specifies PPP, MP, or MPP for an AppleTalk routing connection.
Recv-Password	In the PPP-Options subprofile, specifies the password sent to the MAX TNT by the PPP client.
Atalk-Routing-Enabled	In the AppleTalk-Options subprofile, enables AppleTalk routing for the connection. If AppleTalk routing has not been enabled in the Atalk-Interface profile, or if the Answer-Defaults profile does not enable ARA-Answer, this parameter has no effect.
Atalk-Peer-Mode	In the Appletalk-Options subprofile, specifies whether the remote site is a dial-in client or another AppleTalk router. When set to Router-Peer (the default), the MAX TNT acquires the remote site's network information during session negotiation.

Note: In this release, the Atalk-Static parameters have no useful purpose, because they configure a dialout static route to a remote AppleTalk router, and only dial-in is currently supported. In a future release, Atalk-Static-ZoneName will specify a zone name to be used with the static route to a remote site, and the Atalk-Static-NetStart and Atalk-Static-NetEnd parameters will define the network range for packets that are to be routed to the remote site. These settings will follow the rules described in "Atalk-Interface profile settings" on page 53.

In Figure 8, the remote Pipeline unit is configured as an AppleTalk router on the extended AppleTalk network 2000-2001, in the Branch zone.

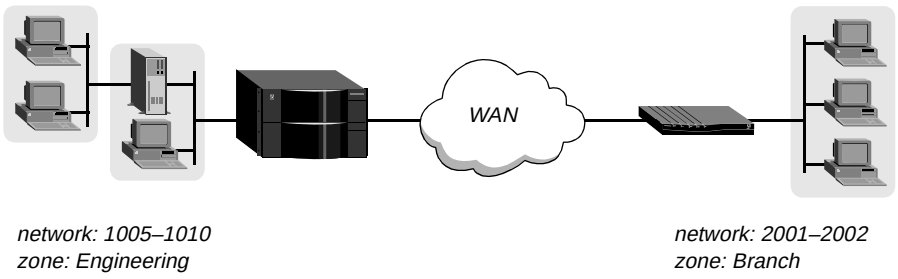

Figure 8. AppleTalk routing connection

Following is an example Connection profile for the remote Pipeline:

```
admin> read connection atalk-router
CONNECTION/atalk-router read

admin> set active = yes

admin> set encaps = ppp

admin> set ppp recv-password = rtr-password

admin> set appletalk atalk-routing enabled = yes

admin> set appletalk atalk-peer-mode = router-peer
```

```
admin> write
CONNECTION/atalk-router written
```

Supporting IP over AppleTalk

To route IP and AppleTalk, the MAX TNT must be configured both as an IP router as well as an AppleTalk router. For details about configuring the IP router and individual IP connections, see the *MAX TNT Network Configuration Guide*.

To support IP, the Connection profile for a dial-in client must specify an IP configuration, and the client must configure Macintosh TCP/IP software (such as Open Transport). Table 2 describes Macintosh TCP/IP configurations for a PPP connection:

Table 2. Macintosh TCP/IP settings for PPP connections

Macintosh software	IP settings for a PPP AppleTalk connection
Open Transport	The TCP/IP Control Panel must specify a PPP connection and the client IP address. If the Connection profile has a hard-coded IP address, type that address manually in the Control Panel. If the Connection profile specifies dynamic address assignment, set the Control Panel to obtain an address from the PPP server.
MacTCP	The MacTCP Control Panel should select the PPP icon, and the client IP address. If the Connection profile has a hard-coded IP address, type that address manually in the Control Panel. If the Connection profile specifies dynamic assignment of an address, set the Control Panel to obtain an address from a Server. (The Dynamic option in MacTCP is not supported.)

The same requirements apply to apply to an ARA connection. When ARA encapsulation is in use, the MAX TNT handles IP packets by encapsulating and decapsulating the packets in DDP. Table 3 describes Macintosh TCP/IP configurations for a PPP connection:

Table 3. Macintosh TCP/IP settings for ARA connections

Macintosh software	IP settings for an ARA connection
Open Transport	The TCP/IP Control Panel must specify a connection via Mac-IP, and the client IP address. If the Connection profile has a hard-coded IP address, type that address manually in the Control Panel. If the Connection profile specifies dynamic assignment of an address, set the Control Panel to obtain an address from the Mac-IP server.
MacTCP	The MacTCP Control Panel should select the ARA icon, and the client IP address. If the Connection profile has a hard-coded IP address, type that address manually in the Control Panel. If the Connection profile specifies dynamic assignment of an address, set the Control Panel to obtain an address from a Server. (The Dynamic option in MacTCP is not supported.)

In Figure 9, the dial-in client is running ARA 3.0 (which includes DDP-IP tunneling capabilities) and an IP application such as Telnet to communicate with an IP host on the MAX TNT local interface. The client has a hard-coded IP address.

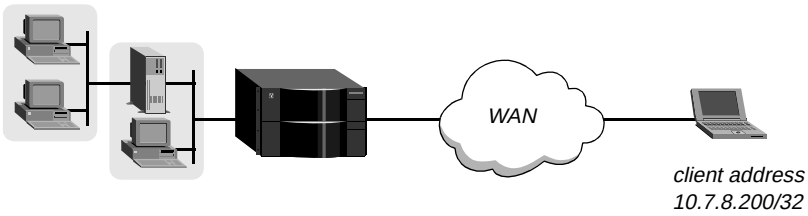

Figure 9. ARA connection that encapsulates IP packets in DDP

The following sample configuration enables the client to dial in using ARA client 3.0 and then initiate a Telnet connection to a host on the MAX TNT unit's IP network:

```
admin> read connection ddpip-client
CONNECTION/ddpip-client read

admin> set active = yes

admin> set encaps = ara

admin> set ara ara-enabled = yes

admin> set ara recv-password = ara-password

admin> set ip-options remote = 10.7.8.200/32

admin> write
CONNECTION/ddpip-client written
```

Frame Relay Switching

In this release, the MAX TNT supports a switched mode of operation between Frame-Relay interfaces (as opposed to routed or bridged operation). When a Frame Relay circuit has been specified in two Connection profiles, the MAX TNT switches data received on one of the paired interfaces to the other paired interface. Circuit switching occurs at layer 2 based on the assigned DLCIs. The layer 3 IP router never receives the packets.

A circuit is a Permanent Virtual Circuit (PVC) segment that consists of two DLCI endpoints, and typically uses two different Frame-Relay profiles. In this release, the Frame-Relay profiles can specify Network-to-Network Interface (NNI) operations. NNI is a mode of operation in which the system acts as a Frame Relay switch communicating with another Frame Relay switch.

Note: NNI operations are not required for the MAX TNT to operate in switched mode using Frame Relay circuits. However, to make an NNI interface useful for passing data, it must be paired with another interface to complete a switch circuit.

New parameters for Frame Relay circuits

Following are the parameters related to Frame Relay circuits, which are shown with sample settings:

```
CONNECTION station-1
   encapsulation-protocol = frame-relay-circuit
```

```
      fr-options
         frame-relay-profile = max
         dlci = 55
         circuit-name = frcir1

   CONNECTION station-2
      encapsulation-protocol = frame-relay-circuit
      fr-options
         frame-relay-profile = p130-east
         dlci = 77
         circuit-name = frcir1
```

Parameter	Effect
Encapsulation-Protocol	Both endpoints of the circuit must specify Frame-Relay-Circuit encapsulation.
Frame-Relay-Profile	Specifies the name of a Frame-Relay profile.
DLCI	Specifies a DLCI number for this endpoint of the circuit.
Circuit-Name	Specifies a name for the circuit (up to 16 characters). The other endpoint must specify the same circuit name.

Note: IP routing is not enabled in the circuit profiles. Circuit switching occurs at layer 2 based on the assigned DLCIs. The layer 3 IP router never receives the packets.

New parameter setting for NNI

The following parameter specifies NNI operations on a data link to another FR switch. The parameter is shown with a sample setting:

```
FRAME-RELAY fr-name
   link-type = nni
```

Parameter	Effect
Link-Type	Specifies the kind of logical interface between the MAX TNT and the Frame Relay network on the data link. Setting the Link-Type to NNI puts the link into NNI operating mode. The MAX TNT performs bidirectional LMI on the link.

Following are important differences in NNI as opposed to UNI operation:

- Received frames are switched to another DLCI circuit segment, rather than being routed (layer 3) or bridged.
- Link management is bidirectional on an NNI interface.
- Asynchronous update reports are sent when DLCI state changes occur.

Configuring the physical link for a Frame Relay interface

Frame Relay switching requires two Frame Relay interfaces (Frame-Relay profiles) and two Connection profiles to define the circuit.

Each Frame-Relay profile requires its own nailed physical link. For example, you could set all 24 channels on a T1 to the same nailed group number to dedicate a T1 line to Frame-Relay, or you could configure the link on a FrameLine card, SWAN card, or any other card that supports sufficient nailed bandwidth.

Following is an example that shows how to define one physical link by configuring all channels of a T1 line for nailed usage and assigning them to the Nailed-Group number 11:

```
admin> read t1 { 1 13 6 }
T1/{ shelf-1 slot-13 6 } read
admin> set line channel 1 channel-usage = nailed-64-channel
admin> set line channel 1 nailed-group = 11
admin> set line channel 2 channel-usage = nailed-64-channel
admin> set line channel 2 nailed-group = 11
admin> set line channel 3 channel-usage = nailed-64-channel
admin> set line channel 3 nailed-group = 11
admin> set line channel 4 channel-usage = nailed-64-channel
admin> set line channel 4 nailed-group = 11
admin> set line channel 5 channel-usage = nailed-64-channel
admin> set line channel 5 nailed-group = 11
admin> set line channel 6 channel-usage = nailed-64-channel
admin> set line channel 6 nailed-group = 11
admin> set line channel 7 channel-usage = nailed-64-channel
admin> set line channel 7 nailed-group = 11
admin> set line channel 8 channel-usage = nailed-64-channel
admin> set line channel 8 nailed-group = 11
admin> set line channel 9 channel-usage = nailed-64-channel
admin> set line channel 9 nailed-group = 11
admin> set line channel 10 channel-usage = nailed-64-channel
admin> set line channel 10 nailed-group = 11
admin> set line channel 11 channel-usage = nailed-64-channel
admin> set line channel 11 nailed-group = 11
admin> set line channel 12 channel-usage = nailed-64-channel
admin> set line channel 12 nailed-group = 11
admin> set line channel 13 channel-usage = nailed-64-channel
admin> set line channel 13 nailed-group = 11
admin> set line channel 14 channel-usage = nailed-64-channel
admin> set line channel 14 nailed-group = 11
admin> set line channel 15 channel-usage = nailed-64-channel
admin> set line channel 15 nailed-group = 11
admin> set line channel 16 channel-usage = nailed-64-channel
admin> set line channel 16 nailed-group = 11
admin> set line channel 17 channel-usage = nailed-64-channel
```

```
admin> set line channel 17 nailed-group = 11
admin> set line channel 18 channel-usage = nailed-64-channel
admin> set line channel 18 nailed-group = 11
admin> set line channel 19 channel-usage = nailed-64-channel
admin> set line channel 19 nailed-group = 11
admin> set line channel 20 channel-usage = nailed-64-channel
admin> set line channel 20 nailed-group = 11
admin> set line channel 21 channel-usage = nailed-64-channel
admin> set line channel 21 nailed-group = 11
admin> set line channel 22 channel-usage = nailed-64-channel
admin> set line channel 22 nailed-group = 11
admin> set line channel 23 channel-usage = nailed-64-channel
admin> set line channel 23 nailed-group = 11
admin> set line channel 24 channel-usage = nailed-64-channel
admin> set line channel 24 nailed-group = 11
admin> write
T1/{ shelf-1 slot-13 6 } written
```

For details about configuring lines and channels, see the *MAX TNT Hardware Installation Guide*.

Configuring a circuit between UNI interfaces

Figure 10 shows a circuit configuration in the MAX TNT. In this example, assume that the two T1 lines in the MAX TNT are configured for nailed usage. The channels of the T1 line connecting to the Pipeline 130 labeled *P130-East* are in Nailed-Group 33, and the channels of the T1 line connecting to the MAX are in Nailed-Group 11 (see “Configuring the physical link for a Frame Relay interface” on page 62).

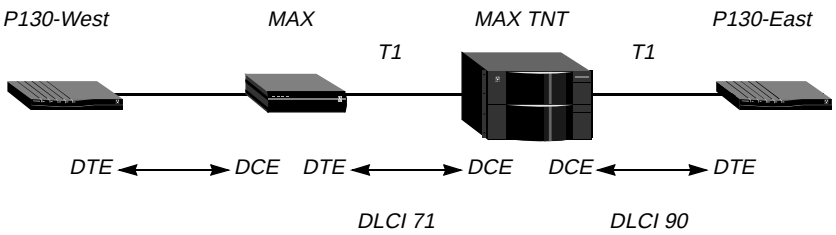

Figure 10. Frame Relay circuit with UNI interfaces

The following commands define a Frame-Relay profile to the MAX in Figure 10:

```
admin> new frame max
FRAME-RELAY/max read
admin> set active = yes
admin> set nailed-up-group = 11
admin> set link-type = dce
```

```
admin> write
FRAME-RELAY/max written
```

The following commands define a Frame-Relay profile to *P130-East* in Figure 10:

```
admin> new frame p130east
FRAME-RELAY/p130east read
admin> set active = yes
admin> set nailed-up-group = 33
admin> set link-type = dce
admin> write
FRAME-RELAY/p130east written
```

The following commands configure a Connection profile to send or receive data to or from the MAX on DLCI 71:

```
admin> read conn max-frcir
CONNECTION/max-frcir read
admin> set active = yes
admin> set encaps = frame-relay-circuit
admin> set ip-options ip-routing-enabled = no
admin> set fr-options frame-relay-profile = max
admin> set fr-options dlci = 71
admin> set fr-options circuit-name = frcir1
```

The following commands configure a Connection profile to send or receive data to or from *P130-East* on DLCI 90:

```
admin> read conn p130-frcir
CONNECTION/p130-frcir read
admin> set active = yes
admin> set encaps = frame-relay-circuit
admin> set ip-options ip-routing-enabled = no
admin> set fr-options frame-relay-profile = p130east
admin> set fr-options dlci = 90
admin> set fr-options circuit-name = frcir1
```

Configuring a circuit between NNI interfaces

Figure 11 shows a circuit configuration that uses NNI interfaces. In this example, assume that the lines connecting the MAX TNT to each of the switches have been configured for nailed usage. For this example, the line connecting to the switch labeled *FR-Asnd-A* is in Nailed-Group 52, and the line connecting to the switch labeled *FR-Asnd-B* is in Nailed-Group 128.

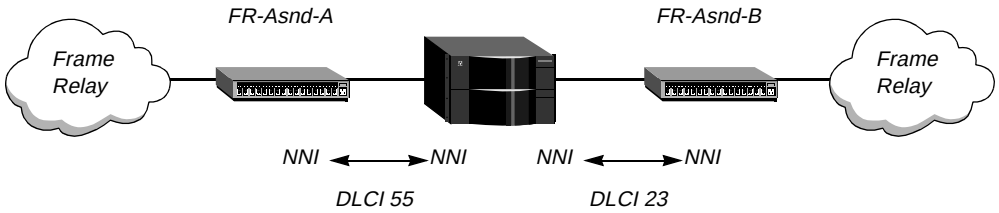

Figure 11. Frame-Relay profile defines NNI operation

The following commands define a Frame-Relay profile to *FR-Asnd-A* in Figure 11:

```
admin> new frame fr-asnd-a
FRAME-RELAY/fr-asnd-a read
admin> set active = yes
admin> set nailed-up-group = 52
admin> set link-type = nni
admin> set link-mgmt = ansi-t1.617d
admin> set n391-val = 6
admin> set t391-val = 10
admin> set t392-val = 15
admin> write
FRAME-RELAY/fr-asnd-a written
```

The following commands define a Frame-Relay profile to *FR-Asnd-B* in Figure 11:

```
admin> new frame fr-asnd-b
FRAME-RELAY/fr-asnd-b read
admin> set active = yes
admin> set nailed-up-group = 128
admin> set link-type = nni
admin> set link-mgmt = ansi-t1.617d
admin> set n391-val = 6
admin> set t391-val = 10
admin> set t392-val = 15
admin> write
FRAME-RELAY/fr-asnd-b written
```

The following commands configure a Connection profile to send or receive data to or from the switch named *FR-Asnd-A* on DLCI 55:

```
admin> new conn asnd-a
CONNECTION/asnd-a read
admin> set active = yes
admin> set encaps = frame-relay-circuit
admin> set ip-options ip-routing-enabled = no
admin> set fr-options frame-relay-profile = fr-asnd-a
admin> set fr-options dlci = 55
```

```
admin> set fr-options circuit-name = pvc-pipe
admin> write
CONNECTION/asnd-a written
```

The following commands configure a Connection profile to send or receive data to or from the switch named *FR-Asnd-B* on DLCI 23:

```
admin> new conn asnd-b
CONNECTION/asnd-b read
admin> set active = yes
admin> set encaps = frame-relay-circuit
admin> set ip-options ip-routing-enabled = no
admin> set fr-options frame-relay-profile = fr-asnd-b
admin> set fr-options dlci = 23
admin> set fr-options circuit-name = pvc-pipe
admin> write
CONNECTION/asnd-b written
```

Ascend Tunnel Management Protocol (ATMP)

ATMP is a UDP/IP based protocol that provides a tunneling mechanism between two Ascend units across an IP network. The data is transported in Generic Routing Encapsulation (GRE) as described in RFC 1701. For a complete description of ATMP, see RFC 2107, K. Hamzeh, "Ascend Tunnel Management Protocol - ATMP."

ATMP tunnels

Figure 12 shows an ATMP tunnel between two MAX TNT units. The unit that authenticates the mobile client is the ATMP *foreign agent*. The unit that accesses the home network is the ATMP *home agent.* The *home network* is the destination network for mobile clients. For example, in Figure 12, the mobile client might be a sales person who logs into an ISP (the foreign agent) to access his or her home network.

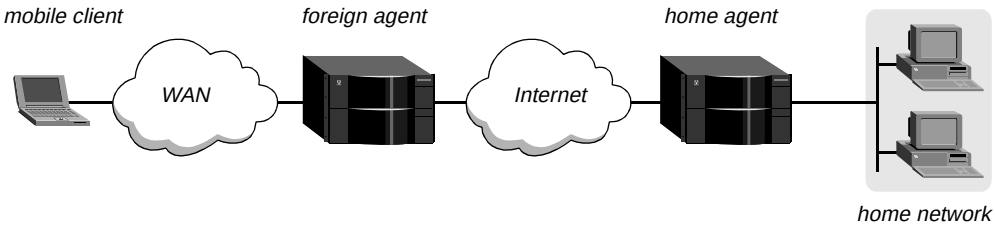

Figure 12. ATMP tunnel across the Internet

A mobile client dials into the foreign agent, which authenticates the client by means of a Connection profile or RADIUS. The foreign agent then establishes an IP connection to the home agent, and requests an ATMP tunnel on top of the established IP connection.

The home agent is the terminating part of the tunnel, where most of the ATMP intelligence resides. It must be able to communicate with the home network through a direct connection, another router, or across a nailed connection.

Setting the system address

If the home agent or foreign agent has multiple interfaces into the IP cloud that separates the two units, it is very important that you set a system IP address. Otherwise, you might encounter communication problems if a route changes within the IP cloud. Following is the relevant parameter, shown with a sample setting:

```
IP-GLOBAL
   system-ip-addr = 10.2.3.4
```

When configuring mobile clients, the IP address of the home agent must be the IP address of the unit (the system address), not the IP address of the interface on which the home agent receives tunneled data.

For more information about the System-IP-Addr parameter, see the *MAX TNT Network Configuration Guide* or the *MAX TNT Reference Guide*.

ATMP profile settings

In the ATMP profile, you specify whether the MAX TNT operates as a home or foreign agent. Create the ATMP profile as follows:

```
admin> new atmp
ATMP read

admin> write
ATMP written
```

The ATMP profile contains the following parameters, which are shown with their default values:

```
ATMP
   agent-mode = tunnel-disabled
   agent-type = gateway-home-agent
   udp-port = 5150
   home-agent-password = ""
   retry-timeout = 3
   retry-limit = 10
```

Table 4. ATMP profile parameters

ATMP parameter	Home agent	Foreign agent
Agent-Mode	Required	Required
Agent-Type	Required	N/A
UDP-Port	Optional	Optional
Home-Agent-Password	Recommended	N/A
Retry-Timeout	Optional	Optional
Retry-Limit	Optional	Optional

Specifying the agent mode

The Agent-Mode parameter specifies whether the MAX TNT operates as a foreign agent, a home agent, or as both on a tunnel-by-tunnel basis. The default tunnel-disabled mode disables ATMP.

Setting the agent type

When Agent-Mode is set to home-agent, the Agent-Type parameter specifies whether the MAX TNT reaches the home network as a gateway or a router. The default is gateway.

When it is set to gateway-home-agent, the home agent delivers tunneled data to the home network without routing. The tunneled data does not bring up a connection to the home network, so the connection between the home agent and home network must already be up, as for example in a nailed or direct connection.

When the Agent-Type parameter is set to router-home-agent, the home agent routes tunneled data to the home network.

Specifying a UDP port for the tunnel

The UDP-Port parameter sets the UDP port the unit uses locally to manage the tunnel. The default value is 5150. Both ends of a tunnel must agree on its value.

Setting the home agent password

When the MAX TNT operates as a home agent, the Home-Agent-Password parameter sets the password a foreign agent must supply to establish a tunnel with this box. You can specify up to 21 characters.

Specifying retry limits

The Retry-Timeout parameter controls the time to wait between retries when attempting to establish a tunnel. The default value is 3 seconds, which is appropriate for most sites.

The Retry-Timeout parameter controls the maximum number of attempts to establish a tunnel before switching to an alternate home agent. You can specify a number between 1 and 100. The default is 10.

Example of setting up a foreign agent

Following is an example of an ATMP profile for a foreign agent:

```
admin> new atmp
ATMP read

admin> set agent-mode = foreign-agent

admin> write
ATMP written
```

For a more detailed example, see “Example of a foreign agent configuration” on page 73.

Example of setting up a home agent

Following is an example of an ATMP profile for a home agent in gateway mode:

```
admin> new atmp
ATMP read
admin> set agent-mode = home-agent
admin> set agent-type = gateway-home-agent
admin> set home-agent-password = my-password
admin> write
ATMP written
```

For a more detailed example, see “Example of a home agent configuration in gateway mode” on page 74.

Connection profile parameters

When the MAX TNT is acting as a home agent in gateway mode, a Connection profile defines the connection to the home network. When it is operating as a foreign agent, Connection profiles authenticate mobile client connections that will be tunneled to a home agent.

Following are the parameters for configuring either of these ATMP-related connections, shown with their default values:

```
CONNECTION station
   tunnel-options
      profile-type = disabled
      max-tunnels = 0
      primary-home-agent = ""
      secondary-home-agent = ""
      udp-port = 5150
      home-agent-password = ""
      home-network-name = ""
```

Table 5. Connection profile tunnel-option parameters

Tunnel parameter	Mobile client	Gateway to home network
Profile-Type	Required	Required
Max-Tunnels	N/A	Optional
Primary-Home-Agent	Required	N/A
Secondary-Home-Agent	Optional	N/A
UDP-Port	Optional	N/A
Home-Agent-Password	Required if specified in home agent	N/A
Home-Network-Name	Required if home agent is in gateway mode	N/A

Specifying a profile type

The Profile-Type parameter specifies the type of connection. Three options are available:

- Disabled

  The connection is not used for ATMP.

- Mobile-Client

  Specifying this setting if the Connection profile is used to authenticate a mobile client.

- Gateway-Profile.

  Specifying this setting if the Connection profile defines a gateway-mode connection to a home network.

Although you cannot set the profile type in RADIUS, it is effectively set to Mobile-Client if the RADIUS profile specifies a home agent address attribute (Ascend-Home-Agent-IP-Addr, Ascend-Primary-Home-Agent or Ascend-Secondary-Home-Agent).

Specifying a tunnel maximum

When the Connection profile defines a gateway-mode connection to a home network, the Max-Tunnels parameter controls the maximum number of mobile clients that can use the connection, all at the same time, to tunnel into that home network. A value of 0 sets no limit. The default value is 0.

Specifying a primary and secondary home agent

For a Mobile-Client connection, the Primary-Home-Agent parameter specifies the IP address or host name of the primary home agent, and the Secondary-Home-Agent parameter specifies the IP address or host name of a secondary home agent.

If you specify a host name (up to 31 characters), the MAX TNT attempts to look up the host IP address in DNS. If the home agent requires a UDP port number different than the value specified in the UDP-Port parameter, you can specify a port value by appending a colon character (:) and the port number to the host name. For example:

```
admin> set primary-home-agent=10.11.22.33:8877
admin> set primary-home-agent=home-agent.company.com:6969
admin> set secondary-home-agent=11.56.12.128:4000
```

The home agent IP address should be the system address, not the IP address of the interface on which it receives tunneled data. (For more detail, see “Setting the system address” on page 68.)

You can specify this information in RADIUS by means of the Ascend-Primary-Home-Agent, Ascend-Secondary-Home-Agent, or Ascend-Home-Agent-IP-Addr attribute.

Specifying a UDP port for the tunnel

For a Mobile-Client connection, the UDP-Port parameter sets the default UDP port to use when communicating with a home agent. You can override this value by specifying the port in the home agent address string, as described in the preceding section.The default value is 5150. Both ends of a tunnel must agree on the value.

You can specify this information in RADIUS by means of the Ascend-Home-Agent-UDP-Port attribute.

Specifying the home agent password

For a Mobile-Client connection, the Home-Agent-Password parameter specifies the password required by the home agent (up to 20 characters). For related information, see “Setting the home agent password” on page 69.

You can specify this information in RADIUS by means of the Ascend-Home-Agent-Password attribute.

Specifying the home network name

For a Mobile-Client connection when the home agent is running in gateway mode, the Home-Network-Name parameter refers to the name of the home network connection. (If the home agent is operating in router mode, leave this parameter blank.) The name of the home network connection is specified in the station parameter of that Connection profile on the home agent.

You can specify this information in RADIUS by means of the Ascend-Home-Network-Name attribute.

Example of a mobile client Connection profile

Following is an example of a procedure that configures the ATMP aspects of a Connection profile for a mobile client:

```
admin> read connection rachel
CONNECTION/rachel read

admin> list tunnel-options
profile-type = disabled
max-tunnels = 0
primary-home-agent = ""
secondary-home-agent = ""
udp-port = 5150
home-agent-password = ""
home-network-name = ""

admin> set profile-type = mobile-client

admin> set primary-home-agent = jupiter.xyz.com

admin> set home-agent-password = jupiter-password

admin> write
CONNECTION/rachel written
```

Example of a home agent gateway mode Connection profile

Following is an example of a procedure that configures the ATMP aspects of a Connection profile for a home agent in gateway mode, supporting a maximum of 120 tunnels to the home network at xyz.com:

```
admin> read connection xyz
CONNECTION/xyz read
```

```
admin> list tunnel-options
profile-type = disabled
max-tunnels = 0
primary-home-agent = ""
secondary-home-agent = ""
udp-port = 5150
home-agent-password = ""
home-network-name = ""

admin> set profile-type = gateway-profile

admin> set max-tunnels = 120

admin> write
CONNECTION/xyz written
```

Example ATMP configurations

This section contains several examples that show how to set up foreign agent, mobile client, and home agent profiles.

Example of a foreign agent configuration

The foreign agent is responsible for authenticating mobile clients and requesting a tunnel across an IP connection to a home agent. Following is an example of a procedure that configures the MAX TNT to operate as a foreign agent, and configures two Connection profiles for mobile clients—one to a home agent in gateway mode, and one to a home agent in router mode.

1 Open the ATMP profile (create it if necessary):

```
admin> new atmp
ATMP read
```

2 Specify foreign-agent mode, and then write the profile:

```
admin> set agent-mode = foreign-agent

admin> write
ATMP written
```

3 Create an IP-routing Connection profile to the home agent.

For information about configuring IP routing connections, see the *MAX TNT Network Configuration Guide*. For information about configuring connections in RADIUS, see the *MAX TNT RADIUS Configuration Guide.*

4 Create a Connection profile for each mobile client that will tunnel to the home agent. The following example shows the ATMP configuration of a Connection profile to a home agent in router mode:

```
admin> read connection ca1
CONNECTION/ca1 read

admin> list tunnel-options
profile-type = disabled
max-tunnels = 0
primary-home-agent = ""
secondary-home-agent = ""
udp-port = 5150
home-agent-password = ""
home-network-name = ""
```

```
admin> set profile-type = mobile-client
admin> set primary-home-agent = home-agent.abc.com
admin> set secondary-home-agent = 11.56.12.128
admin> set home-agent-password = abc-password
admin> write
CONNECTION/ca1 written
```

The following example shows the ATMP configuration of a Connection profile for connecting to a home agent in gateway mode:

```
admin> read connection pac
CONNECTION/pac read
admin> list tunnel-options
profile-type = disabled
max-tunnels = 0
primary-home-agent = ""
secondary-home-agent = ""
udp-port = 5150
home-agent-password = ""
home-network-name = ""
admin> set profile-type = mobile-client
admin> set primary-home-agent = home-agent.xyz.com
admin> set secondary-home-agent = 12.66.56.120
admin> set home-agent-password = abc-password
admin> set home-network-name = homenet
admin> write
CONNECTION/pac written
```

In the last set of commands, *homenet* is the name of the home agent's Connection profile to the home network.

Example of a home agent configuration in gateway mode

A home agent in gateway mode receives GRE-encapsulated IP packets from the foreign agent, strips off the encapsulation, and passes the packets across a WAN connection to the home network. The WAN connection must already be up, because tunneled data does not bring up a connection.

To enable hosts and routers on the home network to reach the mobile client, you must configure a static route in the Customer Premise Equipment (CPE) router on the home network (not in the home agent). The static route must specify the home agent as the route to the mobile client. That is, the route's destination address specifies the address of the mobile client, and its gateway address specifies the IP address of the home agent.

Following is an example of a procedure that configures the MAX TNT to operate as a home agent in gateway mode and configures a Connection profile for connecting to the home network:

1 Open the ATMP profile (create it if necessary):

```
admin> new atmp
ATMP read
```

2 Configure the home agent parameters, and then write the profile:

```
admin> set agent-mode = home-agent

admin> set agent-type = gateway-home-agent

admin> set home-agent-password = my-password

admin> write
ATMP written
```

3 Create an IP-routing Connection profile to the foreign agent.

For information about configuring IP routing connections, see the *MAX TNT Network Configuration Guide*. For information about configuring connections in RADIUS, see the *MAX TNT RADIUS Configuration Guide.*

4 Create a Connection profile for connecting to the home network.

The following example shows the ATMP configuration of a Connection profile for connecting to the home network. For more information about configuring nailed connections, see the *MAX TNT Hardware Installation Guide* and *MAX TNT Network Configuration Guide*.

```
admin> read connection homenet
CONNECTION/homenet read

admin> list tunnel-options
profile-type = disabled
max-tunnels = 0
primary-home-agent = ""
secondary-home-agent = ""
udp-port = 5150
home-agent-password = ""
home-network-name = ""

admin> set profile-type = gateway-profile

admin> set max-tunnels = 50

admin> write
CONNECTION/ca1 written
```

The profile for the connection to the home network must be a local profile (it cannot be specified in RADIUS), and the name of this Connection profile must be specified in the Home-Network-Name parameter or Ascend-Home-Network-Name attribute in the mobile client's configured profile.

Example of a home agent configuration in router mode

A home agent in router mode receives GRE-encapsulated IP packets from the foreign agent, strips off the encapsulation, and then routes the packets to the home network in the usual way. It also adds to its routing table a host route to the mobile client.

If you enable RIP on the home agent's local interfaces, other hosts and networks can route to the mobile client. Enabling RIP is particularly useful if the home network is one or more hops away from the home agent's Ethernet. If RIP is turned off, other routers require static routes that specify the home agent as the route to the mobile client.

Note: If the home agent's local interface is the home network (a direct connection), you should turn on proxy ARP in the home agent to enable local hosts to ARP for the mobile client.

Following is an example of a procedure that configures the MAX TNT to operate as a home agent in router mode:

1 Open the ATMP profile (create it if necessary):

```
admin> new atmp
ATMP read
```

2 Configure the home agent parameters, and then write the profile:

```
admin> set agent-mode = home-agent
admin> set agent-type = router-home-agent
admin> set home-agent-password = my-password
admin> write
ATMP written
```

3 Create an IP-routing Connection profile to the foreign agent.

For information about configuring IP routing connections, see the *MAX TNT Network Configuration Guide*. For information about configuring connections in RADIUS, see the *MAX TNT RADIUS Configuration Guide.*

Example of a home-and-foreign agent configuration

In Figure 13, the MAX TNT operates as a home agent for network B and as a foreign agent for network A. For each tunnel, it meets all of the same requirements described in previous sections for a home agent or foreign agent.

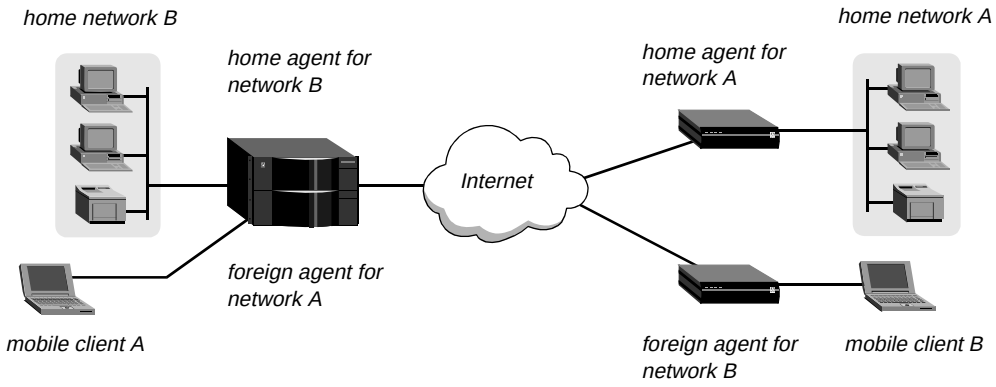

Figure 13. MAX TNT acting as both home agent and foreign agent

Following is an example of a procedure that configures the MAX TNT in Figure 13 to operate as a home agent in router mode for network B, and as a foreign agent for network A:

1 Open the ATMP profile (create it if necessary):

```
admin> new atmp
ATMP read
```

2 Specify home-and-foreign-agent mode:

```
admin> set agent-mode = home-and-foreign-agent
```

3 Configure the home agent parameters, and then write the profile:

```
admin> set agent-type = router-home-agent
admin> set home-agent-password = my-password
admin> write
ATMP written
```

4 Configure a Connection profile for mobile client A:

```
admin> read connection mclientA
CONNECTION/mclientA read

admin> list tunnel-options
profile-type = disabled
max-tunnels = 0
primary-home-agent = ""
secondary-home-agent = ""
udp-port = 5150
home-agent-password = ""
home-network-name = ""

admin> set profile-type = mobile-client

admin> set primary-home-agent = max.home-network-A.com

admin> set home-agent-password = homenetA-password

admin> write
CONNECTION/mclientA written
```

ATMP support for connecting to a GRF switch

Two parameters, MTU-Limit and Force-Fragmentation, have been added to the ATMP profile to enable the MAX TNT to operate as an ATMP foreign agent tunneling to a GRF switch configured as home agent. Figure 14 shows the MAX TNT tunneling to a GRF across a 100-BaseT Ethernet segment:

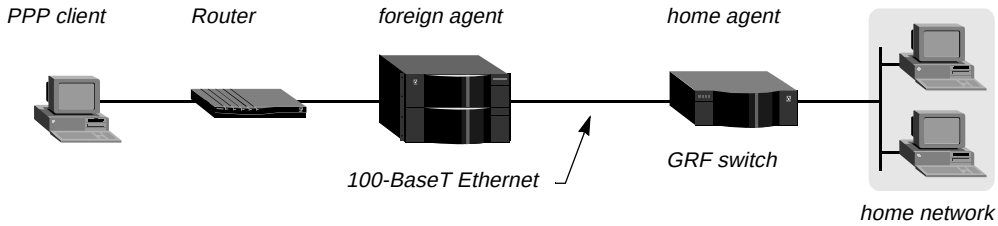

Figure 14. ATMP tunnel to GRF switch

The MAX TNT can receive packets that are larger than the Ethernet Maximum Transmission Unit (MTU) from a PPP client that logs in through a remote access router, such as the Pipeline router in Figure 14. In addition, some clients might send frames larger than the negotiated Maximum Receive Unit (MRU) with the Don't Fragment (DF) bit set, a behavior intended to discover the path's MTU.

In either case, the unit dialing in negotiates an MRU that is large enough to make fragmentation unnecessary. But when the MAX TNT encapsulates the packet in GRE, it adds an 8-byte GRE header and a 20-byte IP header, which can make the packet size larger than the MTU of the tunneled link. So, the packet must either be fragmented or rejected. However, having a very high aggregate forwarding rate at the ATMP gateway requires that the foreign agent fragment the IP packets before encapsulation rather than after.

The following parameters (shown with their default values) enable the MAX TNT to fragment the IP packets before encapsulating them in GRE:

```
ATMP
   mtu-limit = 0
   force-fragmentation = no
```

Prefragmentation is required when the cost of reassembling IP packets should be shifted to the end clients instead of loading the gateway, as is the case with the GRF.

Setting the MTU limit

To determine the maximum size packet it can send to the home agent without prefragmentation, the MAX TNT checks the value of the MTU-Limit parameter. If the parameter is set to zero, the MAX TNT does not perform prefragmentation. If the parameter is set to a non-zero value, it fragments packets at the specified size. For example, to enable the MAX TNT to send full 1500-byte frames on Ethernet, set the MTU-Limit parameter as shown in the following example:

```
admin> read atmp
ATMP read

admin> set mtu-limit = 1472

admin> write
ATMP written
```

Forcing fragmentation

Typically, if the incoming frame has the DF bit set and is too large to tunnel to the ATMP home agent, the MAX TNT returns an ICMP message that informs the client that the host is unreachable without fragmentation. This standard, expected behavior improves end-to-end performance by enabling the client to determine the path's MTU and thereby avoid unnecessary fragmentation and reassembly.

The Force-Fragmentation parameter changes this standard behavior. When prefragmentation is enabled (MTU-Limit parameter set to a non-zero value), and the Force-Fragmentation parameter is set to Yes, the MAX TNT ignores the DF bit and fragments the frame anyway. Following is an example that enables the MAX TNT to fragment and tunnel 1500-byte frames across 100-BaseT, even if the frames' DF bits are set:

```
admin> read atmp
ATMP read

admin> set mtu-limit = 1472

admin> set force-fragmentation = yes

admin> write
ATMP written
```

Note: The Force-Fragmentation parameter enables behavior that is *not standard* and might cause problems. In particular, setting this parameter to Yes disables MTU discovery mechanisms.

Home agent inactivity timers for ATMP tunnels

When an ATMP foreign agent restarts, tunnels that were established to a home agent are not normally cleared, because the home agent is not informed that the mobile clients are no longer connected. So, the home agent does not release the resources held by the unused tunnel. To enable the home agent to reclaim the resources held by unused tunnels, ATMP home agents can now set an inactivity timer using the following parameter, which is shown with its default value:

```
ATMP
   idle-timer = 0
```

The inactivity timer runs only on the home agent side.Its value specifies the number of minutes—from 0 to 65535— that the home agent maintains an idle tunnel before disconnecting it. A value of 0 disables the timer, which means that established tunnels remain connected forever. The setting affects only tunnels created after the timer was set. Existing tunnels are not affected.

In the following example, the timer is set to 30 minutes:

```
admin> new atmp
ATMP read

admin> list
agent-mode = tunnel-disabled
agent-type = gateway-home-agent
udp-port = 5150
home-agent-password = ""
retry-timeout = 3
retry-limit = 10
idle-timer = 0
mtu-limit = 0
force-fragmentation = no

admin> set idle-timer = 30

admin> write
ATMP written
```

New administrative features

A progress indication for Load or Save commands

Load and Save commands now provide an indication that they were progressing during the time required to complete the command. The progress indication is a spinning cursor that is rotated once for every few data packets processed.

Changes to Syslog output

The MAX TNT now reports additional session information about various errors logged via Syslog. The information should assist in identifying all messages associated with a session. It also supports a new Syslog message to provide the disconnect and progress codes for a given session. Error messages associated with a session can contain the following information:

Syntax	Description
`[shelf/slot/line/channel]`	a physical channel identifier
`[MBID xxx]`	a session identifier
`[name]`	the authenticated name
`[ calling -> called ]`	the calling number or the called number, or both

For a given session identifier, multiple physical channel identifiers are possible (for example, one identifier might be for the T1 line, and another for the HDLC channel or modem number). This is shown in the sample log below, in which messages include the MBID, DNIS, and CLID in brackets. Note that slot 1/2 is an 8T1 card, and slot 1/3 is a 48-modem card.

```
...: [1/2/1/2] [MBID 1; 9995551212 -> 7898] Incoming Call
...: [1/3/1/0] [MBID 1; 9995551212 -> 7898] Assigned to port
...: [1/2/1/2] [MBID 1; 9995551212 -> 7898] Call Connected
...: [1/3/1/0] [MBID 1] [balsup-pc] LAN session up: <balsup-pc>
...: [1/3/1/0] [MBID 1] [balsup-pc] LAN session down: <balsup-pc>
...: [1/3/1/0] [MBID 1; 9995551212 -> 7898] Call Terminated
...: [1/3/1/0] [MBID 1] [balsup-pc] : STOP: 'balsup-pc'; cause 45.; progress
60.; host 10.1.26.2
```

Fatal crash information on console

If the MAX TNT crashes, it now prints a stack trace to the console serial port at the bitrate defined in the Serial profile. The following information is included:

```
FE: N, Load: loadname, Version: version
Stack trace: 0xaddr-0 0xaddr-1 0xaddr-2 0xaddr-3 0xaddr-4 0xaddr-5
```

In the first line of output, *N* is a fatal error number, *loadname* is the name of the load (for example, tntsr or tntmdm56k), and *version* is the software version (for example, 2.0.0).

The second line of output displays the top six program counter addresses from the execution stack active at the time of the crash.

Finger (RFC 1288) support and Userstat enhancements

The MAX TNT now supports additional options for displaying user session information. If Finger is enabled in the IP-Global profile, the MAX TNT can return user information to a remote Finger query, such as UNIX client. In addition, the native Userstat command now includes additional options for displaying information in a 140-character-wide table format and includes additional fields.

Finger user information protocol

Finger is described in RFC 1288. To enable it in the MAX TNT, set the following parameter to Yes:

```
IP-GLOBAL
   finger = yes
```

The default value for this parameter is No, which causes the MAX TNT to reject queries from Finger clients with the following message:

```
Finger online user list denied.
```

Setting the Finger parameter to Yes enables the MAX TNT to accept Finger queries and return the requested active session details to a remote client. The client can ask for short or wide format; for example, a UNIX client can request the wide format by using the –l option. The following command:

```
# finger @tnt1
```

displays the narrow (80-character wide) format, and the following command

```
# finger -l @tnt1
```

displays a wide (140-character-wide format of session information for the system named "tnt1." The client can also request the details of all sessions, or of a single session. For example, to request information about a single user named Gavin:

```
# finger gavin@tnt1
```

The Finger forwarding service, which uses the hostname format "@host1@host2", is not supported. If the remote client uses the forwarding request format, the client sees the following message:

```
Finger forwarding service denied.
```

Userstat options

The MAX TNT Userstat command displays session status information. In previous releases, the information was always 80 characters wide, for example:

```
admin> userstat -s
SessionID  Line/Chan  Slot:Item     Rate     Svc Address       Username
228687860  1.01.02/01 1:03:01/01    56K      PPP 10.100.0.1    barney
228687861  1.02.03/02 1:04:02/00    28800    PPP 10.168.6.24   jake
<end user list>  2 active user(s)
```

For information on the fields shown in the output immediately above, see the *MAX TNT Reference Guide*. In this release, the Userstat command supports two new options. The –l option produced a 140-character-wide format with additional fields, and the –d option to cause the output to be "dumped" to the display rather than being shown one page at a time. When the Userstat command line includes the –l option, the following additional fields are reported:

Table 6. Additional fields displayed for Userstat –l

Field Name	Description
Dialed#	The number dialed to initiate this session.
ConnTime	The amount of time in hours:minutes:seconds format since the session was established.
IdleTime	The amount of time in hours:minutes:seconds format since data was last transmitted across the connection.

Userstat –k to terminate user sessions

The Userstat command can now terminate a user session that uses one of the following service types: PPP, SLIP, MPP, Telnet, Telnet binary, Raw TCP, or terminal server. The command cannot terminate Frame Relay or DTPT service types. To terminate a user session, include the –k option on the command line; for example:

```
admin> userstat
SessionID  Line/Chan  Slot:Item   Rate  Svc Address         Username
246986325  1.01.02/01 1:13:01/000 33600 PPP 100.100.8.2     100.100.8.2
<end user list>  1 active user(s)
```

```
admin> userstat -k 246986325
Session 246986325 cleared
```

Slot card and system uptime information

A new command, Uptime, reports how long the system and its individual cards have been up. To enable network management stations to obtain uptime information, a new MIB object named slotLastChange has been added to the Ascend Enterprise MIB.

The Uptime command uses the following syntax:

```
admin> help uptime
uptime usage: uptime [ [ -a ] | [ [ shelf ] slot ] ]
uptime                  display the TNT system uptime.
uptime slot             display the TNT slot card uptime.
uptime shelf slot       display the TNT slot card uptime.
uptime -a               display the uptime for all TNT slot cards.
```

Without an argument, the command displays system uptime. But in the following example, the command displays the uptime for all slot cards in the UP state (cards that are not in the UP state are not reported):

```
admin> uptime -a
13:26:54
{ shelf-1 slot-1 }                 8t1-card    0 days 00:07:04
{ shelf-1 slot-2 }              48modem-card    0 days 00:06:00
{ shelf-1 slot-4 }              128hdlc-card    0 days 00:05:20
{ shelf-1 slot-5 }               4ether-card    0 days 00:06:38
```

Uptime displays the current time (13:26:54 in the preceding example), identifies the slot card, and then displays the length of time the system has been up in days followed by hours:minutes:seconds. The following example shows how long a modem card in slot 2 has been up:

```
admin> uptime 1 2
13:26:39 { shelf-1 slot-2 }         48modem-card 0000 days 00:05:53
```

The slotLastChange object has the following definition in the Ascend Enterprise MIB:

```
slotLastChange   OBJECT-TYPE
     SYNTAX      TimeTicks
     ACCESS      read-only
     STATUS      mandatory
     DESCRIPTION "The value of sysUpTime at the time the TNT slot card
                  entered its current state. For non-TNT systems 0 is
                  always reported."
     ::= { slotEntry 9 }
```

The slotLastChange variable reports the value of sysUpTime at the time the slot card entered its current state.

Frame Relay information reported on SWAN cards

The Ascend Enterprise MIB eventGroup, callStatusGroup, and sessionStatusGroup now contain information for Frame Relay profiles on SWAN slot cards. As a result, the status window in the command-line interface now shows the name of Frame Relay profiles on SWAN lines. For example:

```
5 Connections, 5 Sessions       |tnt-ma Status
0001 frswan1 FRY 13/01/1 64000  |Serial number: 7021016        Version: 2.0.0
0002 frswan2 FRY 13/02/1 64000|
0005 fr2      FRY 15/07/2 1984K| Rx Pkt:       2002
0006 fr1      FRY 15/01/2 1984K| Tx Pkt:       1844
0007 fr4      FRY 15/05/2 1984K|     Col:          4
                                |
                                |-------------------------------------------
                                | 1/13/01 LA n
                                | 1/13/02 LA n
                                | 1/15/01 TE .nnnnnnn nnnnnnnn nnnnnnnn
                                | 1/15/02 TE .------- -------- s------- ------
                                | 1/15/04 RA ........ ........ ........
                                | 1/15/05 TE .nnnnnnn nnnnnnnn nnnnnnnn
                                | 1/15/07 TE .nnnnnnn nnnnnnnn nnnnnnnn
--------------------------------------------------------------------------------
```

New disconnect code (210—slot card down)

Previously, when a slot card went down, the disconnect reason was reported as either 185 (remote end hung up) or 11 (modem loss carrier). For the MAX TNT, the 210 (slot card down) disconnect code has been added to indicate calls terminated due to a slot card that is down or entering the DOWN state. The 210 disconnect code is retrievable from the Ascend accounting events MIB, but it is not reported in RADIUS accounting Stop packets at this release.

It may take up to a minute to detect a card that is down or entering the DOWN state. If the caller terminates the call due to lack of response during that time, the disconnect code 185 or 11 will be reported.

The 210 disconnect code is likely to be reported under the following conditions:

- The card is administratively downed. This may take 2 to 6 seconds to detect.
- The card goes from the UP to the DOWN state. This may take 2 to 6 seconds to detect. If the card fails POST, no calls should be active, so the 210 code will not appear.
- The card reports diagnostics while dumping core. A transition from UP to DIAG indicates a card that is dumping core. This may take 2 to 6 seconds to detect.
- The card dies but stays in UP state. If the remote end doesn't disconnect first, the watchdog timer expires and the shelf resets the card automatically. The 210 is reported a few seconds after card is reset. The total time before 210 is reported is under 70 seconds.

Separate transmit and receive data rates reported

In previous releases, reported data rates reflected the receive rate only, and the transmit rate was not reported. Now, separate transmit (xmit) and receive (recv) data rates are supported in the Ascend enterprise call.mib and event.mib, in Userstat command output, and in RADIUS accounting. In other cases, where a single data-rate field is reported, the data rate still represents the receive rate.

For ISDN calls, the transmit rate shows the transmit data rate. For analog calls, it shows the modem baud rate at the time of the initial connection.

Ascend Call and Event MIBs

The Call MIB now includes callStatusXmitRate (callStatusEntry 14) and callActiveXmitRate (callActiveEntry 14). Following are the new object definitions:

```
callStatusXmitRate   OBJECT-TYPE
     SYNTAX          INTEGER
     ACCESS          read-only
     STATUS          mandatory
     DESCRIPTION     "The transmit rate for ISDN calls or the baud rate
                     for modem calls. A value of 0 is returned if entry
                     is invalid."
     ::= { callStatusEntry 14 }

callActiveXmitRate   OBJECT-TYPE
     SYNTAX          INTEGER
     ACCESS          read-only
     STATUS          mandatory
     DESCRIPTION     "The transmit rate for ISDN calls or the baud rate
                     for modem calls."
     ::= { callActiveEntry 14 }
```

The Event MIB now includes eventXmitRate (eventEntry 23). Following is the object definition:

```
eventXmitRate        OBJECT-TYPE
     SYNTAX          INTEGER
     ACCESS          read-only
     STATUS          mandatory
     DESCRIPTION     "The transmit data rate for ISDN calls or the baud
                rate for modem calls. Rate is given as bits-per-second.
                Applicable for all 'eventType's except callCleared(3).
                For callCleared(3), 0 will be returned. For modem
                calls, value will be 0 for callAnswered(2) events
                since rate is unknown at the time incoming call is
                detected."
     ::= { eventEntry 23 }
```

Userstat output

The Userstat command now reports both the transmit and receive rates. The following example output shows an active session from a 56K modem user:

```
admin> userstat
SessionID  Line/Chan  Slot:Item   Tx/Rx  Rate Svc Address         Username
247351303  3.01.08/12 3:04:05/000 48000/31200 PPP 11.168.6.124    pc-3
<end user list>  1 active user(s)
```

RADIUS accounting

RADIUS accounting now reports the receive date rate in the Ascend-Data-Rate attribute and the transmit data rate in the Ascend-Xmit-Rate attribute. For example, the following RADIUS record was generated by a 56K modem user session:

```
Sat Nov 8 08:11:37 1997
        User-Name = "pc-3"
        NAS-Identifier = 11.168.6.124
        NAS-Port = 33003
```

```
NAS-Port-Type = Async
Acct-Status-Type = Stop
Acct-Delay-Time = 0
Acct-Session-Id = "247351302"
Acct-Authentic = RADIUS
Acct-Session-Time = 36
Acct-Input-Octets = 659
Acct-Output-Octets = 457
Acct-Input-Packets = 18
Acct-Output-Packets = 14
Ascend-Disconnect-Cause = 100
Ascend-Connect-Progress = 60
Ascend-Xmit_rate = 48000
Ascend-Data-Rate = 31200
Ascend-PreSession-Time = 24
Ascend-Pre-Input-Octets = 464
Ascend-Pre-Output-Octets = 423
Ascend-Pre-Input-Packets = 12
Ascend-Pre-Output-Packets = 12
Ascend-Modem-PortNo = 4
Ascend-Modem-SlotNo = 4
Ascend-Modem-ShelfNo = 3
Caller-Id = "5108641846"
Client-Port-DNIS = "7835"
Framed-Protocol = PPP
Framed-Address = 11.168.6.124
```

IP-Pools command

To view the status of the IP address pools configured in the IP-Global profile, use the IP-Pools command, as shown in the following example:

```
admin> ip-pools
Pool#               Base         Count      InUse
1              10.154.3.50        50        0
3              10.154.3.150       50        1

Number of remaining allocated addresses:     99
```

The sample output shows two configured pools, with the base address, address count, and number of addresses in use for each pool.

Netstat command displays active sockets on slot cards

The Netstat command now displays the shelf/slot and socket numbers for active UDP and TCP sockets on slot cards. For example:

```
admin> netstat
udp:
-Socket-  Local Port  InQLen  InQMax    InQDrops    Total Rx
1/c    0        1023       0       1           0           0
1/c    1       route       0       0           0          25
1/c    2        echo       0      32           0           0
1/c    3         ntp       0      32           0           1
1/c    4        1022       0     128           0           0
1/c    5        snmp       0     128           0           0
1/1    0           1       0     256           0           0
```

```
1/1    1        1018       0     128           0           0
1/3    0           3       0     256           0           0
1/3    1        1021       0     128           0           0
1/5    0           5       0     256           0           0
1/5    1        1020       0     128           0           0
1/8    0           8       0     256           0           0
1/8    1        1019       0     128           0           0

tcp:
-Socket- Local                    Remote                State
1/c    0  lab-60.eng.ascen.telnet *.*                  LISTEN
```

In the preceding output, the Socket field shows the shelf/slot followed by the socket number.

Netstat reports TCP statistics collected from slot cards

The Netstat command now displays TCP statistics collected from slot cards, as well as the shelf-controller statistics reported in previous releases. In addition, several new statistics are included. The following command now displays the total of TCP statistics from the shelf-controller and all slot cards:

```
admin> netstat -s tcp
tcp:
0 active opens
3 passive opens
0 connect attempts failed
0 connections were reset
1 connections currently established
3565 segments received
0 segments received out of order
3620 segments transmitted
0 segments retransmitted
1 active closes
1 passive closes
0 disconnects while awaiting retransmission
```

Netstat command reports Finger service (port 79)

Netstat command output now reports the Finger service (port 79). For example, the following output reports both Finger and Telnet services:

```
admin> netstat tcp
tcp:
-Socket- Local                 Remote                       State
1/c    0 host1.eng.abc.finger *.*                         LISTEN
1/c    1 host1.eng.abc.telnet *.*                         LISTEN
1/c    2 host1.eng.abc.finger ridgeback.eng.abc..38040   TIME-WAIT
1/c    3 host1.eng.abc.finger ridgeback.eng.abc..38041 ESTABLISHED
1/c    4 host1.eng.abc.finger ridgeback.eng.abc..38042 ESTABLISHED
1/c    5 host1.eng.abc.finger ridgeback.eng.abc..38043 ESTABLISHED
1/c    6 host1.eng.abc.finger ridgeback.eng.abc..38044 ESTABLISHED
```

In the example output, the *name.service* field is fixed width, and the *name* (a hostname or full domain name) has been truncated to accommodate that width.

For TCP, Netstat now reports the following services:

Service	TCP port number
Telnet	23
TACACS+	49
Finger	79

For UDP, Netstat now reports the following services:

Service	UDP port number
Route	520
Echo	7
NTP	123
SNMP	161
SNMPTrap	162

As always, if the port being used is not found among these named services, it is printed as a number. Also, if the –n option is used on the Netstat command line, numeric addresses and port numbers are always reported instead of names.

Add DNIS and CLID to Syslog messages

DNIS and CLID information are displayed in Syslog messages that relate to a call, provided that the information is known. Following is an example that shows the DNIS 7895 in Syslog messages:

```
LOG info, Shelf 1, Controller, Time: 17:48:56--
† shelf 1, slot 1, line 1, channel 6, dnis 7895, Incoming Call, MBID 001

LOG info, Shelf 1, Controller, Time: 17:48:56--
† shelf 1, slot 2, dnis 7895, Assigned to port, MBID 001

LOG info, Shelf 1, Controller, Time: 17:48:57--
† shelf 1, slot 1, line 1, channel 6, dnis 7895, Call Connected, MBID 001

LOG warning, Shelf 1, Controller, Time: 17:49:20--
† shelf 1, slot 1, line 1, channel 6, dnis 7895, Call Disconnected

LOG info, Shelf 1, Controller, Time: 17:49:20--
† shelf 1, slot 2, Call Terminated
```

Call logging using the RADIUS accounting protocol

Call logging is a RADIUS-accounting based feature for logging call information from the MAX TNT. Its main purpose is to duplicate accounting information for sites that wish to keep accounting records separate from other groups that might need call-logging details to manage resources or troubleshoot call problems.

Once you have configured call logging, the MAX TNT sends Start Session, Stop Session, and Failure-to-Start Session packets to a call-log host. A call-log host is a local host that supports the RADIUS accounting protocol and is configured properly to communicate with the MAX

TNT (for example, a RADIUS accounting server or a host running NavisAccess). The call-log information is sent independently of RADIUS accounting records. If both call logging and RADIUS accounting are in use, the information is sent in parallel.

You set the following parameters, shown with their default values, to configure the MAX TNT to communicate with one or more call-log hosts:

```
CALL-LOGGING
   call-log-enable = no
   call-log-host-1 = 0.0.0.0
   call-log-host-2 = 0.0.0.0
   call-log-host-3 = 0.0.0.0
   call-log-port = 0
   call-log-key = ""
   call-log-timeout = 0
   call-log-id-base = acct-base-10
   call-log-reset-time = 0
   call-log-stop-only = yes
   call-log-limit-retry = 0
```

Following is an example of a procedure that enables call logging and specifies one call-log host on the local network:

```
admin> read call-logging
CALL-LOGGING read

admin> list
call-log-enable = no
call-log-host-1 = 0.0.0.0
call-log-host-2 = 0.0.0.0
call-log-host-3 = 0.0.0.0
call-log-port = 0
call-log-key = ""
call-log-timeout = 0
call-log-id-base = acct-base-10
call-log-reset-time = 0
call-log-stop-only = yes
call-log-limit-retry = 0

admin> set call-log-enable = yes

admin> set call-log-host-1 = 10.2.3.4

admin> write
CALL-LOGGING written
```

The parameters shown have the following functions:

Parameter	Function
Call-Log-Enable	Enables call logging. If set to No, none of the other call-logging parameters apply. If set to Yes, you must specify the IP address of at least one call-log host in the Call-Log-Host-*N* parameters

Parameter	Function
Call-Log-Host-*N*	Each specifies the IP address of one call-log host. The MAX TNT first tries to connect to server #1 for call-logging. If it receives no response, it tries to connect to server #2. If it receives no response from server #2, it tries server #3. If the MAX TNT connects to a server other than server #1, it continues to use that server until it fails to service requests, even if the first server has come online again.
Call-Log-Port	Specifies the UDP destination port to use for call-logging requests. The default value of 0 (zero) indicates any UDP port. If you specify a different number, the call-log host must specify the same port number (the numbers must match).
Call-Log-Key	A shared secret that enables the server to receive data from the MAX TNT. The value must match the configured shared secret on the call-log host.
Call-Log-Timeout	Specifies the number of seconds the MAX TNT waits for a response to a call-logging request. It can be set to a value of from 1 to 10. The default value is 0 (zero), which disables the timer.
Call-Log-ID-Base	Specifies whether the MAX TNT presents a session ID to the call-log host in base 10 or base 16. The default is base 10.
Call-Log-Reset-Time	Indicates the number of seconds that must elapse before the MAX TNT returns to using the primary call-log host (Call-Log-Host-1). The default value of 0 (zero) disables the reset to the primary call-log host.
Call-Log-Stop-Only	Specifies whether the MAX TNT should send an Stop packet with no user name. The MAX TNT typically sends Start and Stop packets to record connections. Authentication is required to send a Start packet. There are situations that the MAX TNT will send an Stop packet without having sent an Start packet in which case the Stop packets have no user name. The default value for Call-Log-Stop-Only is Yes. You can set it to No to prevent the unit from sending Stop packets with no user name.

Call-Log-Limit-Retry	If the server does not acknowledge a Start or Stop packet within the number of seconds specified in Call-Log-Timeout, the MAX TNT tries again, resending the packet until the server responds or the packet is dropped because the queue is full. The Call-Log-Limit-Retry parameter sets the maximum number of retries for these packets. The default value of 0 (zero) indicates an unlimited number of retries. There is minimum of 1 retry. The following example limits the number of retries to 10. There will be a total of 11 attempts: the original attempt plus 10 retries.
	`admin> **read call-logging**`
	`CALL-LOGGING read`
	`admin> **set call-log-limit-retry = 10**`
	`admin> **write**`
	`CALL-LOGGING written`

Configurable session listing output

Administrators can display a list of active sessions by using the Userstat command or sending a Finger query to a MAX TNT that has enabled Finger in the IP-Global profile. Both methods of displaying the list of sessions use the same output format, for example:

```
SessionID  Line/Chan  Slot:Item   Tx/Rx Rate  Svc Address       Username

250808749  1.01.01/02 1:04:03/018 56000/56000 PPP 100.100.1.8     max8
250808750  1.01.01/01 1:04:03/019 56000/56000 PPP 100.100.1.9     max9
<end user list>  2 active user(s)
```

This output format can now be customized. Following is the relevant parameter, which is shown with its default value:

```
SYSTEM
   userstat-format = %i %l %s %r %d %a %u %c %t %n
```

The Userstat-Format parameter specifies a series of conversion strings (a character preceded by a percent-sign), which are described in Table 1. You can enter up to 72 characters. The maximum width of the output string depends on the width of the fields present in the session listing output (see Table 1). If you enter a character without a percent-sign, it is printed as a literal character in the session listing output.

Table 1. Userstat-Format conversion strings

String	Field Width	Output Text	Meaning
%i	10	SessionID	Unique ID assigned to the session
%l	10	Line/Chan	Physical address (shelf.slot.line/chan)
%s	11	Slot:Item	shelf:slot:item/logical-item of the host port
%r	11	Tx/Rx Rate	Transmit and receive rates
%d	3	Svc	A three-letter code showing the type of service
%a	15	Address	IP address
%u	14	Username	Connection profile name
%c	10	ConnTime	Amount of time connected in hours:minutes:seconds
%t	10	IdleTime	Amount of time idle in hours:minutes:seconds
%n	24	Dialed#	Number dialed if known

The default value of the Userstat-Format parameter causes the standard session listing output format for both the Userstat command and responses to Finger queries. For example:

```
admin> get system userstat-format
userstat-format = %i %l %s %r %d %a %u %c %t %n
```

```
admin> userstat

SessionID  Line/Chan  Slot:Item   Tx/Rx Rate  Svc Address       Username

250808749  1.01.01/02 1:04:03/018 56000/56000 PPP 100.100.1.8     max8
250808750  1.01.01/01 1:04:03/019 56000/56000 PPP 100.100.1.9     max9
<end user list>  2 active user(s)
```

The following example customizes the session listing output to include only the Username, Svc, and ConnTime information, with an at-sign appearing between the service and connection-time for each session:

```
admin> read system
SYSTEM read

admin> set userstat-format = %u (%d) @ %c

admin> write
SYSTEM written

admin> userstat

Username       Svc    ConnTime

joeb          (PPP) @ 1:22:34
jimmyq        (PPP) @ 3:44:19
sallyg        (PPP) @ 5:12:56

<end user list>  3 active user(s)
```

External authentication and accounting features

Conflicts between RADIUS and local configurations resolved

Because external RADIUS and local profiles are integrated into the same database, the MAX TNT platform does not allow external RADIUS profiles and local profiles with the same name. If the administrator executes a Write command to save a local profile, and there is already an external RADIUS profile with the same name, the following error message appears:

```
error: Cannot overwrite duplicate external CONNECTION/testdlci profile
```

When external RADIUS profiles are retrieved, if the profile index is a duplicate of a local profile of the same type, the following message is logged at the Error level:

```
"Could not add duplicate RADIUS CONNECTION profile 'testdlci'"
```

In addition, external RADIUS profiles now correctly report the time they were last refreshed. Note that configurations saved via TFTP or the serial console do not include profiles that were retrieved from an external database or that are read-only.

Setting the number of RADIUS accounting retries

When the MAX TNT is configured for RADIUS accounting, it sends accounting Start and Stop packets to the RADIUS server to record connections. If the server does not acknowledge a packet within the number of seconds in Acct-Timeout, the MAX TNT tries again, resending the packet until the server responds or the packet is dropped because the queue is full. The following new parameter specifies the maximum number of retries for accounting packets:

```
EXTERNAL-AUTH
   rad-acct-client
      acct-limit-retry = 0
```

The default value of 0 (zero) indicates an unlimited number of retries, which means that the MAX TNT resends the packet until the server responds or the packet is dropped because the queue is full. There is minimum of 1 retry. The following example limits the number of retries to 10. There will be a total of 11 attempts: the original attempt plus 10 retries.

```
admin> read external-auth
EXTERNAL-AUTH read

admin> list rad-acct-client
acct-server-1 = 10.2.3.56
acct-server-2 = 10.7.8.62
acct-server-3 = 10.5.6.11
acct-port = 1646
acct-src-port = 0
acct-key = *******
acct-timeout = 5
acct-sess-interval = 0
acct-id-base = acct-base-10
acct-reset-time = 0
acct-checkpoint = 0
acct-limit-retry = 0
acct-stop-only = yes

admin> set acct-limit-retry = 10

admin> write
EXTERNAL-AUTH written
```

NAS-port type added to RADIUS accounting

The MAX TNT bit-encodes the 16-bit NAS-Port number sent to RADIUS accounting daemons to indicate the shelf, slot, line, and channel on which a call was received. It now also includes a NAS-Port-Type value to indicate whether the established session uses asynchronous or synchronous transmission.

The NAS-Port-Type may be one of the following values:

- NAS_Port_Type_Sync (1) for synchronous sessions.
- NAS_Port_Type_Async (0) for asynchronous sessions.

RADIUS accounting for failed authentication

RADIUS accounting keeps records of sessions, which are typically used for billing and security tracking. When RADIUS accounting is in use, the MAX TNT sends a Stop packet to the RADIUS server when a session terminates. RADIUS accounting Stop packets are normally sent for authenticated connections, connections that are dropped before authentication, and connections that fail authentication.

You can configure the MAX TNT not to send Stop packets for connections that fail authentication by changing the setting of the following parameter:

```
EXTERNAL-AUTH
   radius-acct-client
      acct-drop-stop-on-auth-fail = no
```

The default value is No. If the parameter is set to Yes, RADIUS accounting Stop packets are not sent for connections that fail authentication. The commands in the following example configure the MAX TNT not to send Stop packets for connections that fail authentication:

```
admin> read external-auth
EXTERNAL-AUTH read

admin> set radius-acct-client acct-drop-stop-on-auth-fail = yes

admin> write
EXTERNAL-AUTH written
```

For more information about RADIUS accounting, see the *MAX TNT RADIUS Configuration Guide*.

Preventing accounting Stop packets with no user name

When the MAX TNT is configured for RADIUS accounting, it sends accounting Start and Stop packets to the RADIUS server to record connections. Authentication is required to send a Start packet. There are situations in which the MAX TNT will send an accounting Stop packet without having sent an accounting Start packet, in which case the Stop packets have no user name.

The following parameter has been added to specify whether the MAX TNT should send an accounting Stop packet with no user name:

```
EXTERNAL-AUTH
   rad-acct-client
      acct-stop-only = yes
```

The default value is Yes. You can set this parameter to No to prevent the unit from sending Stop packets with no user name to the RADIUS server. For example:

```
admin> read external-auth
EXTERNAL-AUTH read

admin> list rad-acct-client
acct-server-1 = 10.2.3.56
acct-server-2 = 10.7.8.62
acct-server-3 = 10.5.6.11
acct-port = 1646
acct-src-port = 0
acct-key = *******
acct-timeout = 5
acct-sess-interval = 0
acct-id-base = acct-base-10
acct-reset-time = 0
acct-checkpoint = 0
acct-limit-retry = 0
acct-stop-only = yes

admin> set acct-stop-only = no

admin> write
EXTERNAL-AUTH written
```

Distinct ID sequences for RADIUS authentication and accounting

RADIUS uses an ID value to aid in request-response matching. By default, the MAX TNT uses a single sequence space for the RADIUS ID number in all RADIUS messages. A single space limits the number of IDs available for assignment to 256. In this release, you can configure distinct ID sequence spaces for RADIUS accounting and authentication packets.

When you configure the MAX TNT to use distinct ID sequence spaces, the RADIUS server must perform additional checks to detect duplicates. The server should check the RADIUS ID value as well as the service type and destination UDP port in each packet. The service type can be determined by sorting all values of the code field into two classes—auth and acct—and then comparing the received code value to determine to which class it belongs. The destination UDP port can be the same for both services when a single RADIUS server performs both services.

To configure the MAX TNT to use distinct ID sequence spaces, use the following parameter (shown with its default setting):

```
EXTERNAL-AUTH
   rad-id-space = unified
```

When rad-id-space is set to Unified, RADIUS authentication and accounting packets share the same ID sequence space, so a combined total of 256 authentication and accounting packets are sent before the ID sequence rolls over. The commands in the following example configure the MAX TNT to use distinct sequence spaces:

```
admin> read external
EXTERNAL-AUTH read

admin> set rad-id-space = distinct

admin> write
EXTERNAL-AUTH written
```

When RAD-ID-Space is set to Distinct, RADIUS authentication and accounting packets do not share the same ID sequence space. The MAX TNT can send a total of 256 authentication packets before the authentication ID sequence rolls over, and 256 accounting packets before the accounting ID sequence rolls over. Three sequence spaces are allocated: one for the Unified sequence space, and one each for the authentication and accounting ID sequences.

Unique RADIUS accounting IDs based on source port number

RADIUS ID values are used in request-response matching. For each unique accounting request (including retries, if a response is not received within the configured timeout period), an 8-bit ID value is assigned. The assigned value is freed when the request is no longer pending (when the request has been matched with a response or timed out).

On a high-capacity NAS, such as the MAX TNT, the RADIUS ID space can be used up, so that no unique values are available for the next accounting request. Recognizing this problem, the IETF RADIUS Working Group will propose a change to RFC 2138 (RADIUS), requiring use of the source UDP port in request-response matching. Each request will be identified by the UDP source port as well as the RADIUS ID value.

The proposed change will allow large NASes to have more than 256 outstanding requests. The Ascend RADIUS server supports the use of the source UDP port in request-response matching, as do several other RADIUS server implementations. To configure the MAX TNT to send the

source UDP port number in RADIUS request-to-response matching, use the following parameter (shown with its default setting):

```
EXTERNAL-AUTH
   rad-id-source-unique = system-unique
```

The RAD-ID-Source-Unique parameter controls whether ID values must be unique system-wide or per requesting port. Unique IDs per-system is the default behavior. To include the source UDP port number in response matching, set the parameter to port-unique, as shown in the following example:

```
admin> read external
EXTERNAL-AUTH read

admin> set rad-id-source-unique = port-unique

admin> write
EXTERNAL-AUTH written
```

Note that if the RADIUS server does not distinguish requests on the basis of port and ID values, this configuration has no effect.

RADIUS accounting Start packet includes user's IP address

RADIUS accounting sends a Start packet to a server when a user is authenticated. The RADIUS accounting Start packet has been extended to include the following fields:

Field	Value
Framed-Address	The IP address assigned to the user. This field is included when RADIUS authentication for a scripted user returned a Framed-User attribute.
Framed-Protocol	The encapsulation protocol (PPP/SLIP) in use. This field is included when RADIUS authentication for a scripted user returned a Framed-Protocol attribute.

ATMP attributes in RADIUS accounting Stop records

Previously, RADIUS Stop packets for ATMP-tunneled connections contained the Tunneling-Protocol attribute (127). RADIUS accounting now reports more information about ATMP connections with the addition of three attributes in accounting Stop packets generated by ATMP foreign agents for terminated mobile-client connections. The new attributes are shown in Table 7:

Table 7. ATMP-related attributes in RADIUS Stop packets

Attribute Name	Number	Type
Ascend-Home-Agent-IP-Addr	183	integer
Ascend-Home-Agent-UDP-Port	186	integer
Ascend-Home-Network-Name	185	string

The Ascend-Home-Agent-IP-Addr attribute indicates the IP address of the home agent used for the mobile client whose tunneled connection has terminated.

The Ascend-Home-Agent-UDP-Port attribute indicates the UDP port used for communicating with the home agent for the terminated connection.

The Ascend-Home-Network-Name attribute indicates the name of the mobile client's home network for this mobile client. This attribute is not present in the Stop record if the home agent is configured in ATMP router mode.

For details about these aspects of ATMP configuration, see "Ascend Tunnel Management Protocol (ATMP)" on page 67 of this Release Note.

Following is an example of a Stop record that contains these new attributes:

```
Mon Oct 27 02:41:38 1997
User-Name = "Jacob"
NAS-Identifier = 1.1.1.1
NAS-Port = 10105
Acct-Status-Type = Stop
Acct-Delay-Time = 0
Acct-Session-Id = "111111111"
Acct-Authentic = RADIUS
Acct-Session-Time = 0
Acct-Input-Octets = 215
Acct-Output-Octets = 208
Acct-Input-Packets = 10
Acct-Output-Packets = 10
Ascend-Disconnect-Cause = 1
Ascend-Connect-Progress = 60
Ascend-Data-Rate = 56000
Ascend-PreSession-Time = 1
Ascend-Pre-Input-Octets = 215
Ascend-Pre-Output-Octets = 208
Ascend-Pre-Input-Packets = 10
Ascend-Pre-Output-Packets = 10
Framed-Protocol = PPP
Framed-Address = 2.2.2.2
Tunneling-Protocol = ATMP
Ascend-Home-Agent-IP-Addr = 3.3.3.3
Ascend-Home-Agent-UDP-Port = 5150
Ascend-Home-Network-Name = homenet
```

New Local-Profiles-First setting

The Local-Profiles-First parameter in the External-Auth profile specifies what action to take if the MAX TNT fails its first attempt to authenticate a connection, whether locally or by using an external server. Prior to this release, the parameter could be set to either Yes or No to set the authentication sequence for the TNT. In this release, a new RNAK setting has been added. With this setting, if a NAK is returned from the remote server, the MAX TNT terminates the connection without proceeding to a local profile. Following is a synopsis of the system behavior associated with each setting of the Local-Profiles-First parameter:

When Local-Profiles-First = Yes, the MAX TNT checks the local (NVRAM) profile first.

- If the profile exists and the password matches, it allows the connection.

- If the profile exists and the password does not match, it checks external authentication.
- If the profile does not exist, it proceeds with external authentication.

If Local-Profiles-First = No, the MAX TNT checks external authentication first.

- If the server ACKs the request, it allows the connection.
- If the server doesn't respond, it checks for a matching local profile.
- If the server NAKs the request, it checks for a matching local profile.

If Local-Profiles-First = RNAK, the MAX TNT checks external authentication first.

- If the server ACKs the request, it allows the connection.
- If the server doesn't respond, it checks for a matching local profile.
- If the server NAKs the request, it terminates the connection.

The SNMP sysAuthPreference variable has also been updated to include the new RNAK setting, as shown below:

```
sysAuthPreference   OBJECT-TYPE
                    INTEGER {
                    no-op(1),
                    local-first(2),
                    remote-first(3),
                    remote-first-no-local-if-naked(4)
                    }
ACCESS              read-write
STATUS              mandatory
DESCRIPTION
"An incoming call can be authenticated using a local profile or one from
an authentication server such as RADIUS or TACACS. Local-first means
authenticate from a local profile first, and if that fails, try the
authentication server. Remote-first means get a profile from the authen-
tication server and authenticate from that and if that fails, try to
authenticate from a local profile. Remote-first-no-local-if-naked is
similar to remote-first, except if the external authentication server
NAK the request, than the external authentication server NAK the
request, than the connection will be denied, i.e. no search of the local
profiles will be made."
::= { systemStatusGroup 10 }
```

Configurable cause element in ISDN Disconnect packets

When CLID or DNIS authentication fails, the MAX TNT can now return either User Busy (decimal 17) or Normal Call Clearing (decimal 16) as the Cause Element in ISDN Disconnect packets. The default behavior is to send Normal Call Clearing (16).

The administrator can configure the Cause Element value by setting the following parameters, shown with their default values:

```
EXTERNAL_AUTHENTICATION
   rad-auth-client
      auth-id-fail-return-busy = no
      auth-id-timeout-return-busy = no
```

If these parameters are set to No (the default), Normal Call Clearing (16) is sent when CLID or DNIS authentication fails or times out.

If you set the Auth-ID-Fail-Return-Busy parameter to Yes, User Busy (17) is sent when CLID or DNIS authentication fails.

If you set the Auth-ID-Timeout-Return-Busy parameter to Yes, User Busy (17) is sent when CLID or DNIS authentication times out.

Nailed connections retrieved from RADIUS

The MAX TNT now brings up nailed connections retrieved from a RADIUS server. Previously, when the MAX TNT retrieved a nailed Connection or Frame Relay profile from a RADIUS server, it did not take the action required to bring the connection up. Now, the retrieved profiles are handled as if they were local profiles, and are visible when a user executes a Dir command. For example, in the following Dir command output, the sample profile named "rad" is a nailed connection retrieved from a RADIUS server:

```
admin> dir conn
    44  07/30/1997 14:59:25  don
    67  09/05/1997 13:32:23  Miami
    52  10/30/1997 12:16:22  rad
```

Nailed RADIUS profiles are displayed along with the local profiles, and you can open them. However, the profiles retrieved from RADIUS are read-only in the command-line interface.

Shelf-controller proxy RADIUS accounting

In earlier releases, RADIUS accounting records were handled solely by the host card that performed the packet-handling functions, such as a modem or HDLC card. If the card went down with open sessions, no Stop records were sent to the accounting server for those sessions. (A RADIUS accounting checkpoint feature provides periodic session information to enable session billing even if the RADIUS accounting server didn't receive a Stop record, but this is not considered an ideal solution.)

In this release, the master shelf-controller keeps track of all accounting Start records sent by host cards. If the shelf-controller determines that a host card has gone down for any reason, it acts as proxy for the card and sends the accounting server a fail-safe Stop record for each of the card's open sessions. The host card may be brought down administratively (Slot –d), removed from the system, or it may go down due to an error condition.

How proxy accounting works

When RADIUS accounting is in use, the usual situation occurs as shown in Figure 15:

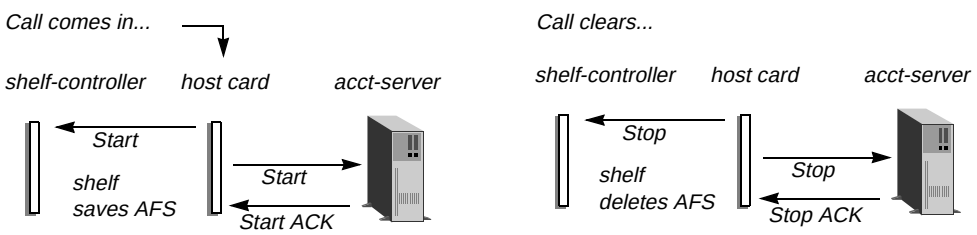

Figure 15. Normal RADIUS accounting (no proxy necessary)

When a call comes in, the host card first sends a Start record to the shelf-controller, which stores it as an Accounting Fail-Safe (AFS) record. The host card then sends one or more Start records to the RADIUS accounting server, repeating until it receives an ACK from the server. Similarly, when the call clears, the host card sends a Stop record to the shelf-controller, which causes it to delete the AFS record for that session. The host card then sends the accounting server Stop records until it receives an ACK from the server.

When RADIUS accounting is in use and the host card goes down for any reason, proxy accounting occurs as shown in Figure 16:

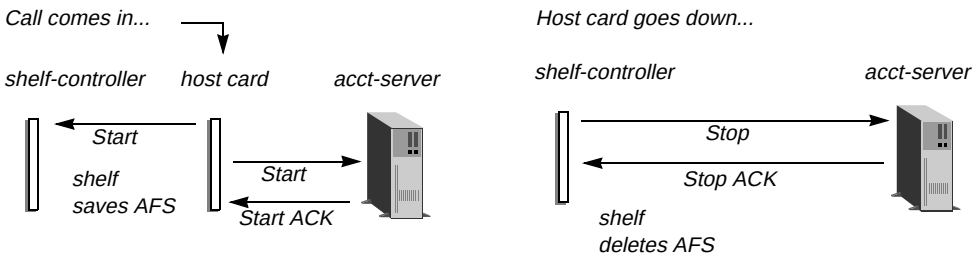

Figure 16. Proxy accounting (host card goes down)

In this case, when the shelf-controller notes that the host card is down, it uses its own information about the host card and the stored AFS record to send a Stop record directly to the RADIUS accounting server, repeating until it receives a Stop ACK from the server. The shelf-controller then deletes the AFS record for that session.

Note that if the accounting server is accessible only via the host card that goes down, Stop records cannot be delivered successfully in any case.

Contents of the Stop record sent by proxy

The AFS Stop record does not contain all of the information it would have if the host card had sent it; in particular it does not contain the input/output octet count fields or any other dynamic information related to the session. In Table 2, Yes means that the attribute is included in the Stop record if applicable. For example, a modem port number is not applicable to a non-modem call. No means that the attribute is either not included in the record or is set to null, as appropriate.

Table 2. Accounting attributes included in proxy Stop records

Attribute in regular Stop record	In proxy Stop record:
Acct-Authentic	Yes
Acct-Delay-Time	Yes
Acct-Input-Octets	No
Acct-Input-Packets	No
Acct-Output-Octets	No
Acct-Output-Packets	No
Acct-Session-Id	Yes

Table 2. Accounting attributes included in proxy Stop records

Attribute in regular Stop record	In proxy Stop record:
Acct-Status-Type	Yes
Acct-Session-Time	Yes. (The session time is accurate to within a few seconds.)
Ascend_Called_Number	No
Ascend-Connect-Progress	Yes
Ascend-Data-Rate	Yes
Ascend-Disconnect-Cause	Yes. (The Disconnect reason is always 210, slot card down.)
Ascend-First-Dest	No
Ascend-Home-Agent-IP-Addr	Yes
Ascend-Home-Agent-UDP-Port	Yes
Ascend_Modem_PortNo	Yes
Ascend_Modem_ShelfNo	Yes
Ascend_Modem_SlotNo	Yes
Ascend-Multilink-ID	Yes
Ascend-Num-In-Multilink	Yes
Ascend-Pre-Input-Octets	No
Ascend-Pre-Input-Packets	No
Ascend-Pre-Output-Octets	No
Ascend-Pre-Output-Packets	No
Ascend-PreSession-Time	Yes
Caller-Id	No
Class	No
Framed-Address	Yes
Framed_IPX_Network	Yes
Framed-Protocol	Yes
Login-Host	Yes
Login-Service	Yes

Table 2. Accounting attributes included in proxy Stop records

Attribute in regular Stop record	In proxy Stop record:
Login-TCP-Port	Yes
NAS-Identifier	Yes
NAS-Port	Yes
NAS-Port-Type	Yes
Tunneling-Protocol	Yes
User-Name	Yes
per-user-rad-acct	Yes. (This information is used to choose an accounting server when a connection is authenticated, and is not part of accounting logging.)

Example of a Stop record sent by proxy

Following is an example of a shelf-controller accounting proxy for an HDLC call:

```
Wed Nov 5 14:50:21 1997
        User-Name = "joel-mhp"
        NAS-Identifier = 206.65.212.199
        NAS-Port = 2272
        NAS-Port-Type = Sync
        Acct-Status-Type = Stop
        Acct-Delay-Time = 0
        Acct-Session-Id = "246212864"
        Acct-Authentic = RADIUS
        Acct-Session-Time = 4
        Acct-Input-Octets = 0
        Acct-Output-Octets = 0
        Acct-Input-Packets = 0
        Acct-Output-Packets = 0
        Ascend-Disconnect-Cause = 210
        Ascend-Connect-Progress = 67
        Ascend-Data-Rate = 0
        Ascend-PreSession-Time = 0
        Ascend-Pre-Input-Octets = 0
        Ascend-Pre-Output-Octets = 0
        Ascend-Pre-Input-Packets = 0
        Ascend-Pre-Output-Packets = 0
        Framed-Protocol = PPP
        Framed-Address = 192.168.6.66
```

Debugging proxy accounting

The Acct-Failsafe debug command is available on the master shelf or the slot host cards for verifying correct accounting proxying. (Slot host cards do not include the "-d" option.) Slave shelf controllers and slot line cards do not support this command. With debug permissions enabled, you can display the usage statement by typing:

```
admin> acct-failsafe
usage: acct-failsafe -option [ params ]
       -d <shelf> <slot>
           (d)isplay AFS info for <shelf> <slot>
       -d (d)isplay AFS info for all relevant slots
       -t (t)oggle module debug level
       -? display this summary
```

To display information about the calls on any slot which are candidates for proxy accounting.:

```
admin> acct-failsafe -d
Slot 1/8:
HashTable @ 10542160, bucketCount: 192, callCount: 23, hashName <afs-1:8>
Slot 2/5:
HashTable @ 10585730, bucketCount: 48, callCount: 7, hashName <afs-2:5>
```

To display the same information for a single slot card in shelf 1, slot 8:

```
admin> acct-failsafe -d 1 8
Slot 1/8:
HashTable @ 10542160, bucketCount: 192, callCount: 23, hashName <afs-1:8>
```

To specify which level of debug to use for the command, use the –t option. A debug level of zero indicates none (no messages). A level of 7 is fairly verbose. For example:

```
admin> acct-failsafe -t 7
acct-failsafe debug output at level 7
```

NAS port identifier optionally reported in new format

By default, the MAX TNT reports the NAS port value in a format that is appropriate to the multishelf and multislot architecture of the system. For details on this default format, see the *MAX TNT RADIUS Configuration Guide*. Now, administrators can specify that the NAS port value must be represented in five digits, as follows:

tllcc

In this format, the following definitions apply:

- *t* = 1 = digital call or 2 = analog call
- *ll* = line number
- *cc* = channel number

For this definition to make sense with the MAX TNT architecture, the following requirements must be met:

- Multishelf is not supported.
- All line cards must be placed contiguously in the first slots of the standalone system. Lines must be numbered 1 through 8 for slot 1, 9 through 16 for slot 2, and so on.

Following is the new parameter that configures the MAX TNT to use the five-digit NAS port value:

```
SYSTEM
   new-nas-port-id-format = yes
```

If set to Yes (the default), the MAX TNT uses the NAS port identifier format that matches its architecture. For details on this format, see the *MAX TNT RADIUS Configuration Guide*. If

New-NAS-Port-ID-Format is set to No, the MAX TNT uses the five-digit NAS port values. Note that the system must be standalone.

The following example configures the MAX TNT to use five-digit NAS port values:

```
admin> read system
SYSTEM read

admin> set new-nas-port-id-format = no

admin> write
SYSTEM written
```

Note: Do not set the New-NAS-Port-ID-Format parameter while the system has active sessions if external accounting is in use.

Enhanced SNMP support

SNMP agent on multishelf system now reports on slave cards

In a multishelf system, the master shelf-controller keeps status information about all slots in the system. The SNMP agent on the master shelf-controller reports status information on the slots in the Ascend Enterprise MIB Slots group. The slotIndex for the cards in each shelf in a multishelf system is shown below:

Multishelf slots	slotIndex value
Shelf 1 Slots 1 – 18	1 – 18
Shelf 2 Slots 1 – 18	19 – 36
Shelf 3 Slots 1 – 18	37 – 54
Shelf 4 Slots 1 – 18	55 – 72
Shelf 5 Slots 1 – 18	73 – 90
Shelf 6 Slots 1 – 18	91 – 108
Shelf 7 Slots 1 – 18	109 – 126
Shelf 8 Slots 1 – 18	127 – 144
Shelf 9 Slots 1 – 18	145 – 162

Slots 1-16 represent the actual removable slot cards. Slots 17 represents the shelf controller. Slot 18 is reserved for future use.

For example, for a multishelf MAX TNT with master shelf 4 and slave shelves 3 and 7, the slotIndex range would be 37-54 for slave shelf 3, 55-72 for the master shelf, and 109-126 for slave shelf 7.

Ability to disconnect user via SNMP request

An SNMP Set request can now terminate a user session by setting one of the following SNMP objects in the session MIB to invalid(1):

`ssnStatusValidFlag` as part of `sessionStatusTable`

`ssnActiveValidFlag` as part of `sessionActiveTable`

Additional information about SNMP-initiated transfers

The MAX TNT supports the following MIB variable, which is used to ascertain the status of a TFTP download or upload initiated via SNMP:

```
ascend.systemStatusGroup.sysConfigTftp.sysConfigTftpStatus
```

In the past, the reported status was limited to Passed or Failed. It can now be used to determine the following states:

```
ok( 1 ),                    -- tftp operation succeeded
notFound( 2 ),              -- file not found
access( 3 ),                -- access violation
noSpace( 4 ),               -- no disk space to write file
badTid( 6 ),                -- unknown transfer ID
badOp( 5 ),                 -- bad tftp operation
exists( 7 ),                -- file already exists
noSuchUser( 8 ),            -- no such user
parameter( 9 ),             -- parameter error
busy( 10 ),                 -- tftp server cannot handle request
noResources( 11 ),          -- no memory for request
timeout( 12 ),              -- timed out
unrecoverable( 13 ),        -- unrecoverable error
tooManyRetries( 14 ),       -- too many retries
createFile( 15 ),           -- create file
openFile( 16 ),             -- open file
inProgress( 17 )            -- get/put request in progress
```

Ability to enable and disable modems via SNMP

An SNMP Set request can now enable or disable a modem by setting the following SNMP object in the slots group to invalid(1):

```
ascend.slots.slotMdmTable.slotMdmEntry.slotMdmItemConfig
```

SNMP support for TNT IDSL slot card

Line status, line traps, and user connection statistics for the IDSL slot card are now retrievable via SNMP.

SNMP advanced.mib now supported

This release supports the Ascend Advanced MIB, previously called the WAN MIB.The Advanced MIB defines objects related to WAN lines, channels, and ports.

DTPT sessions to the ZGR identified in Session MIB

In this release, outgoing DTPT sessions to the ZGR are identified in the sessionStatusTable and sessionActiveTable as part of session.mib. To use this feature, administrators must compile the new session.mib file into their management stations.

The following variables have been added to the two tables in the session.mib to identify DTPT sessions to the ZGR:

- ssnStatusCurrentService

- ssnActiveCurrentService

These variables support the following new values:

- virtualConnect(16),-- Virtual Connect to a modem
- dchannelX25(17), -- D Channel X.25
- dtpt(18) -- DTPT session to ZGR.

Multishelf traps enabled by default

In previous releases, the value of the SNMP MIB object multiShelfStateTrapState (multiShelf 6) was set to Disabled by default. Now, it is set to Enabled by default, as shown in the following object definition:

```
multiShelfStateTrapState OBJECT-TYPE
SYNTAX  INTEGER { enabled(1), disabled(2) }
ACCESS   read-write
STATUS   mandatory
DESCRIPTION
"This variable indicates whether the master system
produces the multiShelfStateChange trap."
DEFVAL { enabled }
::= { multiShelf 6 }
```

This object determines whether a trap is generated when a multishelf link is down (if one of the shelves is down.) If it is set to Disabled (2), the trap is not sent, regardless of Trap profile configurations. If it is set to Enabled (the default), the Slot-Enabled parameter in a Trap profile determines whether the specified host receives the multishelf trap. Only if Slot-Enabled is set to Yes in the Trap profile does the specified host receives multishelf traps.

In the following example, Host-A (10.2.3.4) receives multishelf traps and Host-B (10.5.6.7) does not:

```
admin> new trap host-a
TRAP/host-a read

admin> list
host-name* = test
community-name = ""
host-address = 0.0.0.0
alarm-enabled = yes
security-enabled = no
port-enabled = no
slot-enabled = no

admin> set host-address = 10.2.3.4

admin> set slot-enabled = yes

admin> write
TRAP/host-a written

admin> new trap host-b
TRAP/host-b read

admin> set host-address = 10.5.6.7

admin> write
TRAP/host-b written
```

If the administrator sets the multiShelf.multiShelfStatTrapState object to 2 (disabled), neither host receives multishelf traps.

Slave shelves generate trap when multishelf link is down

In this release, both the master shelf and slave shelf can forward SNMP traps if the multishelf link between them is down. If the link is down because the master shelf is powered down or reset, the slave shelf forwards a trap. If the link is down because the multishelf cables are disconnected, both the master and slave shelves can forward a trap.

If traps are enabled on both the master and slave shelf controllers, a trap with the following OID may be generated to indicate multishelf link conditions:

`.1.3.6.1.4.1.529.19.5.1.2.`*X*

A trap is reported by both the master and slave shelf-controllers when a multishelf cable between the master and slave is disconnected. In this case, *X* in the OID is the number of the shelf that lost communication, and the trap value is 1 (idle).

A trap is reported by either the master or slave shelf-controller if one of them is reset.In this case, *X* in the OID is the number of the shelf that lost communication, and the trap value is 1 (idle).

A trap is reported by the master shelf-controller when the link is back up again. In this case, *X* in the OID is the destination shelf number, and the trap value is 4 (up). This trap is reported only by the master shelf to indicate that the entire multishelf system is up.

Customized features: T-Online

This section describes features that apply only to the Deutsche Telekom T-Online service. T-Online support was introduced in release 1.3A. Throughout this section, DIRDO stands for Dial-In Redirect Dial-Out.

T-Online: PRI-PRI switching for E1

PRI-PRI switching for T-Online provides a network side implementation of NET-5 to support switching calls from the Deutsche Telekom public network to a T-Online server. If T-Online is enabled in the System profile, the MAX TNT compares the phone number and subaddress number it obtains from the call Setup and Info messages to the DIRDO info stored in RADIUS. It switches the inbound call to the T-Online server if it finds any of the following matches in RADIUS:

- The phone number and subaddress of the incoming call match a phone number and subaddress entry in RADIUS.
- The phone number matches a phone number entry in RADIUS and there is no subaddress.
- The subaddress matches a subaddress entry in RADIUS and there is no phone number.
- There is no incoming-call phone number or subaddress.

The MAX TNT begins collecting the subaddress information, and for each call Setup message from the switch that does *not* include “Sending Complete Information Element,” it starts the T302 timer (the Setup Ack timer).

The MAX TNT stops the timer when it receives a message that includes “Sending Complete Information Element.” The MAX TNT assumes there are no more subaddress digits to collect when the T302 timer stops or expires.

Setting up PRI-PRI switching in the System profile

To support PRI-PRI switching for T-Online, the following parameters have been added to the System profile (shown with their default values):

```
SYSTEM
   t-online = no
   t-online-most-avail-chan = no
```

To set up PRI-PRI switching for T-Online, you enable T-Online and specify how the system chooses which NT-configured line to choose for redirecting the call.

The T-Online parameter enables the MAX TNT to route calls to a T-Online server.

Note: Trunk group 8 is reserved for DTPT calls when the T-Online parameter is set to Yes.

The T-Online-Most-Avail-Chan parameter specifies the channel allocation algorithm. Set it to Yes to choose the link with the most available channels. Set it to No to choose a link by the round-robin method.

The commands in the following example enable T-Online and specify that the system should use the link with the most available channels:

```
admin> read system
SYSTEM read

admin> set t-online = yes

admin> set t-online-most-avail-chan = yes

admin> write
SYSTEM written
```

Configuring the E1 lines

To enable PRI-PRI switching on the E1 lines, the following parameters have been added to the E1 line profile (shown with their default values):

```
E1 {shelf-N slot-N N}
   line-interface
      t-online-type = none
      t302-timer = 1500
```

The T-Online-Type parameter is set to None by default. For PRI-PRI switching, you can set it to TE or NT. If you set it to TE, the E1 line should connect to the switch. If you set it to NT, the line should connect to the ZGR server. One TE-configured line can switch calls to one or more NT-configured lines.

The T302-Timer parameter sets the number of milliseconds the MAX TNT waits before assuming that there are no more subaddress digits to collect. You can specify a value range from 100 to 30000 (.1 sec to 30 sec).

Following is an example of a procedure that configures both the TE and NT lines:

```
admin> read e1 {1 16 7}
E1/{ shelf-1 slot-16 7 } read
```

```
admin> set t-online-type = te
admin> set t302-timer = 3132
admin> write
E1/{ shelf-1 slot-16 7 } written
admin> read e1 {1 16 8 }
E1/{ shelf-1 slot-16 8 } read
admin> set t-online-type = nt
admin> set t302-timer = 3132
admin> write
E1/{ shelf-1 slot-16 8 } written
```

With this configuration, when both lines are up, the Line status window will display TE for line 7 and NT for line 8.The window displays information about the channels for these lines, based on the call type, as follows:

Normal calls display the standard characters that indicate call status:

Character	Meaning
d	dialing
* (asterisk)	connected call
r	ringing

PRI-PRI calls display the following characters to indicate call status:

Character	Meaning
N	dialing
= (equal-sign)	connected call
R	ringing

DTPT outgoing calls display the following characters to indicate call status:

Character	Meaning
D	dialing
% (percent-sign)	connected call

Pseudo-tunneling PPP for T-Online

The Deutsche Telekom T-Online service enables individuals with single-channel PPP connections to contact a ZGR. The ZGR is itself highly restrictive, requiring a one-to-one correspondence of packets' IP source address to the B channel on which the packets are received. A call to the ZGR via the MAX TNT has the same interface-to-address restriction as a call made directly to the ZGR.

So that the MAX TNT can handle PPP connections in the usual way, and yet comply with the ZGR restrictions, it makes use of a new DTPT encapsulation protocol for connections to a ZGR. WAN traffic that is not routed to a ZGR undergoes no special handling. PPP

authentication, bringing up and tearing down sessions, security, and the IP routing engine all operate normally.

WAN -to-WAN routing to the ZGR

In the usual case, a PPP connection comes in and the MAX TNT authenticates it as usual, builds a session, and then routes the data out over the WAN to the ZGR via a single-channel dedicated call, as shown in Figure 17. Each B-channel connection to the ZGR is associated with one source IP address.

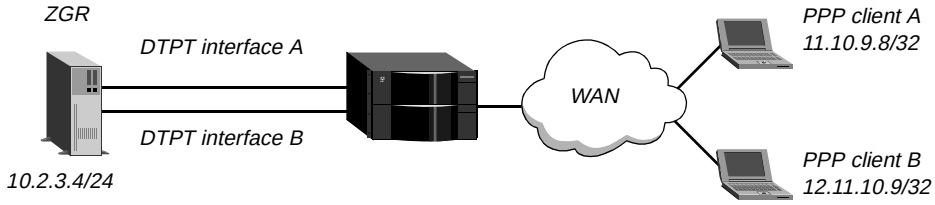

Figure 17. Dial-in PPP connections connecting to a ZGR

IP packets received from a PPP client and destined for the ZGR must all have the same source address. If the dial-in equipment is a router, the first IP address that generates a packet destined for the ZGR owns the resulting ZGR connection, and no other IP addresses behind the same router can connect to the ZGR.

The dial-in client must use PPP encapsulation or one of its multi-channel variants for the initial connection. The actual connection to the T-Online ZGR server is limited to a single B channel, but the user can access the ZGR server and other destinations in the same session.

A call from the MAX TNT to the ZGR can be disconnected for any of the reasons applicable to an ordinary PPP connection, including termination by the ZGR. If the connection to the end-user terminates for any reason, the MAX TNT immediately terminates the associated call to the ZGR.

Local-to-WAN routing to the ZGR

If the packet bringing up the call is originated by the MAX TNT itself, its associated source interface is the loopback interface. Call handling is similar to the WAN-to-WAN routing case, except that, because there is no inbound call, the outbound call is not actively torn down when the source traffic stops. Instead, it behaves like an ordinary outbound PPP call in that it will eventually time out, assuming it has not been terminated by the remote end.

LAN-to-WAN routing to the ZGR

IP packets from the LAN can also be routed to the ZGR via the MAX TNT, as shown in Figure 18:

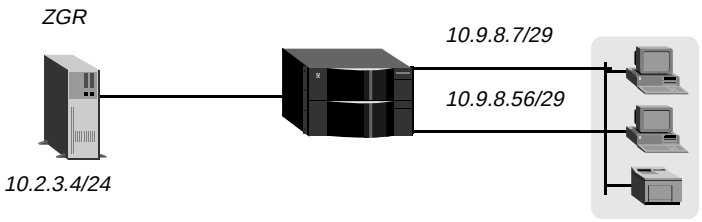

Figure 18. Local clients connecting to a ZGR

Packets from many source addresses might arrive over a single LAN interface. However, there is still a one-to-one correspondence between source IP addresses and outgoing channels to the ZGR. Because the number of source addresses on the LAN can vary, up to a number larger than the number of interfaces on a MAX TNT, it is impossible to guarantee that there will be a B channel available for every attempted access to the ZGR. If LAN-initiated calls to the ZGR are allowed, access might fail for LAN users, WAN users, or both.

Calls placed in response to packets received over the LAN do not automatically terminate when the source of the packets stops sending them, unless you set a value for the Idle-Timer parameter in the Connection Session-Options subprofile. If you set the Idle-Timer parameter, some users might obtain a connection to the ZGR on an initial attempt, but not on a reconnect attempt that occurs after the timer has expired.

Because the LAN clients do not require authentication for sending packets to the MAX TNT, you should ensure security on the LAN by some other means.

Configuring a connection to a ZGR

The following parameters in a Connection profile enable pseudo-tunneled PPP to the ZGR server. The parameters are shown with sample values.

```
CONNECTION station
   encapsulation-protocol = dtpt

   ppp-options
      send-auth-mode = pap-ppp-auth
      send-password = zgr-password
      link-compression = stac
      MRU = 1524

   ip-options
      vj-header-prediction = yes
      remote-address = 10.2.3.4/24
```

These parameters are not new for the DTPT encapsulation type. With the exception of the Encapsulation-Protocol parameter, they do not support new settings for DTPT. However, they are required for optional settings that apply to the DTPT connection to the ZGR.

DTPT encapsulation sets up the call management required for managing multiple discrete B-channel connections to the same destination (the ZGR server). Although in some respects the set of calls to the ZGR resembles an MP or MP+ bundle, they are separate PPP calls, and no bundling or dynamic bandwidth allocation applies.

DTPT encapsulation differs from PPP in that when the MAX TNT uses DTPT to call the remote end, it reports its IP address during NCP negotiations as the IP source address of the packet that caused it to place the call, instead of reporting its own (router) address.

The parameters in Table 8 are not applicable when the DTPT encapsulation type is specified.

Table 8. Parameters that are not applicable for DTPT-encapsulated calls

Parameter that does not apply	Explanation
Telco-Options Answer-Originate	Calls to the ZGR are always outbound.
IP-Options Private-Route	Because the ZGR server has a single IP address, the routing table contains a single route to the ZGR, regardless of how many calls are active. That route is never advertised.
IP-Options RIP	The route to the ZGR is never advertised.

Following is a sample Connection profile for connecting to a ZGR server:

```
admin> new connection zgr-1
CONNECTION/zgr-1 read

admin> set encaps = dtpt

admin> set ppp send-auth-mode = pap-ppp-auth

admin> set ppp send-password = zgr-password

admin> set ip remote-address = 10.2.3.4/24

admin> write
CONNECTION/zgr-1 read
```

Note that PAP authentication is recommended for connecting to the ZGR. If the MAX TNT places calls to the ZGR without proper authentication, the ZGR connection is subject to denial-of-service attacks.

Interface and route handing for DTPT interfaces

For DTPT, the MAX TNT creates a pseudo-interface and aims the routes to any destination with the encapsulation type DTPT at that interface. The pseudo-device name for the DTPT pseudo-interface is dtpt*nnn* where *nnn* is the number of the interface in the interface table. WAN traffic that is not routed to a DTPT interface is not handled any differently than usual.

When the MAX TNT receives a packet whose destination has a Connection profile specifying DTPT encapsulation, it routes the packet (by conventional operation of the routing engine) to the DTPT pseudo-interface, where the source address is examined. The MAX TNT then finds the corresponding output session (with a real outbound interface) and forwards the packet on that interface. No DTPT session is ever used to carry outbound packets from more than one source address.

If the MAX TNT receives a packet for which there is not already a connection to the ZGR, it creates a new dialout interface, associates it with the inbound call, and places the call. The MAX TNT deactivates the interface if the attempt to open the new session fails or if the interface fails a validation test. Because the interface is not permanent, deactivating it removes it from the table.