MAX TNT Network Configuration Guide

Ascend Communications, Inc.
Part Number: 7820-0547-001
For Software Version 1.3A
September 26, 1997

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

Figures

Tables

Introduction

1

This introduction covers the following topics:

What is in this guide

This guide describes how to configure the MAX TNT for network connectivity. It assumes that you have already set up the MAX TNT system (standalone or multishelf), installed the slot cards, and provisioned and tested the lines. If you have not already finished those tasks, please see the *MAX TNT Hardware Installation Guide*.

Each chapter in the guide focuses on a particular aspect of network configuration. To get the full network connectivity you need, you might need information from only a few chapters, or from many chapters.

For example, many dial-in connections require packet routing, either onto a local network or to a next-hop router. In that case, you have to configure both the routing parameters and the encapsulation protocol settings (such as PPP or Frame Relay) that enable the MAX TNT to negotiate a WAN link. So you have to refer to more than one chapter in this guide.

You can also configure the MAX TNT as a multicast forwarder, which maintains IGMP membership lists and forwards multicasts across the WAN (or LAN) to members. In this role, the MAX TNT is not a multicast router, so the multicast configuration requirement is minimal.

The appendixes of this guide focus on network security issues. Security issues are also discussed in Chapter 7, “Packet and Route Filters.”

What you should know

While this guide attempts to provide enough conceptual framework to enable an administrator who is not an expert in a particular network technology to configure the unit accurately, it does not start from the beginning with any network management topic. Following are the general areas in which it is helpful have some existing knowledge when configuring the related network capabilities:

- Dial-in connections such as PPP and multi-link PPP
- Connection cost management and accounting
- Modems
- Frame relay
- IP routing
- DNS
- OSPF routing (if applicable)
- Multicast (if applicable)
- Packet structure and formats (for defining filters)
- Network security

Related publications

Additional information is available in the other guides in the MAX TNT documentation set. If you need more background information than these guides provide, many external references

are readily available on the Web or in technical bookstores. You'll find a partial list of such references below.

MAX TNT documentation set

The MAX TNT documentation set 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* (this manual). Describes how to use the command-line interface to configure WAN connections and other related features.
- *MAX TNT RADIUS 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.

Related RFCs

RFCs are available on the Web at http://ds.internic.net.

Information about PPP connections

For information about PPP connections and authentication, you might want to download one or more of the following:

- RFC 2153: *PPP Vendor Extensions*
- RFC 1994: *PPP Challenge Handshake Authentication Protocol (CHAP)*
- RFC 1990: *The PPP Multilink Protocol (MP)*
- RFC 1989: *PPP Link Quality Monitoring*
- RFC 1974: *PPP Stac LZS Compression Protocol*
- RFC 1962: *The PPP Compression Control Protocol (CCP)*
- RFC 1934: *Ascend's Multilink Protocol Plus (MP+)*
- RFC 1877: *PPP Internet Protocol Control Protocol Extensions for Name Server Addresses*
- RFC 1662: *PPP in HDLC-like Framing*
- RFC 1661: *The Point-to-Point Protocol (PPP)*
- RFC 1638: *PPP Bridging Control Protocol (BCP)*
- RFC 1618: *PPP over ISDN*
- RFC 1332: *The PPP Internet Protocol Control Protocol (IPCP)*

Information about IP routers

RFCs that describe the operation of IP routers include:

- RFC 2030: *Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI*
- RFC 2002: *IP Mobility Support*
- RFC 1812: *Requirements for IP Version 4 Routers*
- RFC 1787: *Routing in a Multi-provider Internet*
- RFC 1582: *Extensions to RIP to Support Demand Circuits*
- RFC 1519: *Classless Inter-Domain Routing (CIDR): An Address Assignment and Aggregation Strategy*
- RFC 1433: *Directed ARP*
- RFC 1393: *Traceroute Using an IP Option*
- RFC 1256: *ICMP Router Discovery Messages*

Information about OSPF routing

For information about OSPF routing, see:

- RFC 1850: *OSPF Version 2 Management Information Base*
- RFC 1587: *The OSPF NSSA Option*
- RFC 1586: *Guidelines for Running OSPF Over Frame Relay Networks*
- RFC 1583: *OSPF Version 2*
- RFC 1246: *Experience with the OSPF protocol*
- RFC 1245: *OSPF protocol analysis*

Information about multicast

For information about multicast, see:

- RFC 1949: *Scalable Multicast Key Distribution*
- RFC 1584: *Multicast Extensions to OSPF*
- RFC 1458: *Requirements for Multicast Protocols*

Information about packet filtering

RFCs that describe firewalls and packet filters include:

- RFC 1858: *Security Considerations for IP Fragment Filtering*
- RFC 1579: *Firewall-Friendly FTP*

Information about general network security

RFCs pertinent to network security include:

- RFC 1704: *On Internet Authentication*
- RFC 1636: *Report of IAB Workshop on Security in the Internet Architecture*
- RFC 1281: *Guidelines for the Secure Operation of the Internet*
- RFC 1244: *Site Security Handbook*

Information about external authentication

For information about RADIUS and TACACS authentication, see:

- RFC 2138: *Remote Authentication Dial In User Service (RADIUS)*
- RFC 1492: *An Access Control Protocol, Sometimes Called TACACS*

ITU-T recommendations

ITU-T recommendations (formerly CCITT) are available commercially. You can order them at http://www.itu.ch/publications/.

Related books

The following books are available in technical bookstores.

- *Routing in the Internet*, by Christian Huitema. Prentice Hall PTR, 1995. Recommended for information about IP, OSPF, CIDR, IP multicast, and mobile IP.
- *SNMP, SNMPV2 and RMON: Practical Network Management*, by William Stallings. Addison-Wesley, 1996. Recommended for network management information.
- *Enterprise Networking: Fractional T1 to Sonet Frame Relay to Bisdn*, by Daniel Minoli. Artech House, 1993. Recommended as a WAN reference.
- *TCP/IP Illustrated*, volumes 1&2, by W. Richard Stevens. Addison-Wesley, 1994.

Documentation conventions

This section shows the documentation conventions used in this guide.

Table 1-1. Documentation conventions

Convention	Meaning
`Monospace text`	Represents text that appears on your computer's screen, or that could appear on your computer's screen.
`Boldface monospace text`	Represents characters that you enter exactly as shown (unless the characters are also in ***`italics`***—see *Italics*, below). If you could enter the characters, but are not specifically instructed to, they do not appear in boldface.
Italics	Represent variable information. Do not enter the words themselves in the command. Enter the information they represent. In ordinary text, italics are used for titles of publications, for some terms that would otherwise be in quotation marks, and to show emphasis.
[ ]	Square brackets indicate an optional argument you might add to a command. To include such an argument, type only the information inside the brackets. Do not type the brackets unless they appear in bold type.

Table 1-1. Documentation conventions (continued)

Convention	Meaning
|	Separates command choices that are mutually exclusive.
>	Points to the next level in the path to a parameter. The parameter that follows the angle bracket is one of the options that appears when you select the parameter that precedes the angle bracket.
Key1-Key2	Represents a combination keystroke. To enter a combination keystroke, press the first key and hold it down while you press one or more other keys. Release all the keys at the same time. (For example, Ctrl-H means hold down the Control key and press the H key.)
Press Enter	Means press the Enter, or Return, key or its equivalent on your computer.
Note:	Introduces important additional information.
 **Caution:**	Warns that a failure to follow the recommended procedure could result in loss of data or damage to equipment.
 **Warning:**	Warns that a failure to take appropriate safety precautions could result in physical injury.

WAN Connections

2

This chapter covers the following topics:

Introduction to WAN connections

WAN connections can be synchronous or asynchronous. A synchronous data link uses HDLC encoding and connects to an access router for a network-to-network link. It is initially routed as a digital call to an HDLC channel in the MAX TNT, and then to the router software. Synchronous connections use Point-to-Point Protocol (PPP), Multilink Protocol (MP), or Multilink Protocol Plus (MP+) encapsulation. Synchronous connections can also use Frame Relay encapsulation, which is described in Chapter 3, "Frame Relay Configuration."

An asynchronous data link uses the kind of serial communications provided by a PC COM port, and is typically initiated by a dial-up modem or V.120 terminal adapter (TA) for a host-to-network or host-to-host connection. It is initially routed as a voice call to a digital modem in the MAX TNT, and then to the terminal-server software.

For asynchronous connections, you must enable the terminal-server software. You might also need to configure modem parameters that determine how the MAX TNT unit's digital modems negotiate with the incoming calls. Asynchronous connections use PPP or V.120 encapsulation, or raw (unencapsulated) TCP.

You can also configure general settings in a Connection profile, which apply to both synchronous and asynchronous connections. After describing how to configure the WAN encapsulation protocols and telco options that enable the MAX TNT to negotiate, authenticate, and build a session, this chapter discusses some telco and session management parameters for controlling costs and providing network security.

Where to find additional configuration information

This chapter does not describe routing configurations. For example, it does not explain how to assign IP addresses or routing metrics. For information about routing, see Chapter 4, "IP Router Configuration."

The Terminal-Server profile contains a wide range of options related to authentication and defining what a user is allowed to do once he or she has logged into the terminal-server software. For information about these topics, see Appendix A, "Authentication Methods," and Appendix B, "Authorization Options."

Frame Relay requires both a Frame-Relay profile and user connection configuration. For a discussion of all aspects of Frame Relay configuration, see Chapter 3, "Frame Relay Configuration."

For information about network security, see Chapter 7, "Packet and Route Filters," and the appendixes of this guide.

Other options for configuring connections

An external authentication server such as RADIUS or TACACS enables administrators to centralize management and authentication of thousands of connections, and many sites use external authentication rather than local Connection profiles. Many of the same options described here are provided in another format in RADIUS or TACACS profiles.

If you are using RADIUS authentication, Ascend has added features to the standard RADIUS daemon to support Ascend-specific connection features. For information about configuring WAN connections in a RADIUS profile, see the *MAX TNT RADIUS Guide*.

If you are using TACACS or TACACS+, the documentation that accompanied the server software explains how to set up the server. Following are the parameters used to configure the MAX TNT to authenticate connections by means of TACACS or TACACS+:

```
EXTERNAL-AUTH
   tac-auth-client
      auth-server-1 = 0.0.0.0
      auth-server-2 = 0.0.0.0
      auth-server-3 = 0.0.0.0
      auth-port = 0
      auth-src-port = 0
      auth-key = ""
      auth-timeout = 0
   tacplus-auth-client
      auth-server-1 = 0.0.0.0
      auth-server-2 = 0.0.0.0
      auth-server-3 = 0.0.0.0
      auth-port = 0
      auth-src-port = 0
      auth-key = ""
      auth-reset-time = 0
      auth-timeout-time = 0
      auth-retries = 0
```

You can specify up to three server addresses, the TCP port to use, a password (key) required by the server, and a timeout value in seconds. In the case of TACACS+, you can also specify when to reset the primary server after a server failure, the amount of time that should elapse before attempting to connect to a backup server, and the number of connection attempts to make. For detailed information about these parameters, see the *MAX TNT Reference Guide*.

About the Answer-Defaults profile

The Answer-Defaults profile sets baseline values that determine how the MAX TNT evaluates incoming calls before accepting (answering) them. If the call does not comply with the Answer-Defaults settings, the unit rejects the call without answering it, so you must check the Answer-Defaults values to make sure they are set properly for your site.

The Answer-Defaults values are applied *before* the MAX TNT routes the call to a modem or HDLC channel for processing, and before it locates the caller's profile locally or in RADIUS. If the caller's profile contains a similar parameter with a different value, the MAX TNT uses the connection-specific value to build the session.

To display the contents of the Answer-Defaults profile, use the Get command:

```
admin> get answer
use-answer-for-all-defaults = yes
force-56kbps = no
profiles-required = yes
clid-auth-mode = ignore
```

```
ppp-answer = { yes no-ppp-auth yes 0 none 1524 no 600 600 }
mp-answer = { yes 1 2 }
mpp-answer = { yes quadratic transmit 0 0 15 5 10 70 }
fr-answer = { yes }
tcp-clear-answer = { yes }
ara-answer = { no }
v120-answer = { yes 256 }
ip-answer = { yes yes no 1 }
session-info = { "" "" no 120 no-idle 120 0 }
```

Default settings

By default, the Answer-Defaults profile enables all types of encapsulation and routing, and the basic call-setup parameters use the lowest common denominator settings. This is appropriate for many sites, but you might want to change the settings to fine-tune the criteria for which calls are accepted, or to constrain how much bandwidth is accessible to multilink PPP calls.

By default, no PPP or Calling Line ID (CLID) authentication is required for incoming calls. Most sites change this default to ensure authentication of the call before a session is established. For instructions, see Appendix A, “Authentication Methods.”

Where to find more information

For information about CLID, Dial Number Information Service (DNIS), and PPP authentication, see Appendix A, “Authentication Methods.” For information about each parameter, see the *MAX TNT Reference Guide*. See the *MAX TNT RADIUS Guide* for information about settings that are useful for RADIUS-authenticated calls.

Configuring PPP connections

This section shows how to configure a PPP connection. Following are the Connection profile parameters related to PPP configuration, shown with the default settings:

```
CONNECTION station
   encapsulation-protocol = ppp
   ppp-options
      send-password = ""
      recv-password = ""
      link-compression = stac
      mru = 1524
      lqm = no
      lqm-minimum-period = 600
      lqm-maximum-period = 600
```

For the MAX TNT to use link compression, both sides of the connection must be configured to use the same compression method. You can set STAC compression to use an Ascend-modified version of draft 0 of the CCP protocol, STAC-9 to use draft 9 of the Stac LZS compression protocol, or MS-STAC for Microsoft/STAC (the method used by Windows95).

You can use the Maximum Receive Units (MRU) parameter to reduce the size of the largest acceptable frame if necessary. Typically, you should use the default of 1524 unless the far end requires a reduction.

You can enable the PPP Link Quality Monitoring (LQM) protocol and specify a maximum and minimum period for generating link quality reports.

For the details of PPP authentication, see Appendix A, "Authentication Methods."

Example of a synchronous PPP connection

The connection shown in Figure 2-1 uses PPP encapsulation and CHAP (Challenge Handshake Authentication Protocol) authentication. The far-end device is a Pipeline unit with the IP address 10.2.3.31/24. The Connection profile sets parameters that enable the MAX TNT to dial out to and receive calls from the Pipeline. This is a single-channel synchronous PPP call.

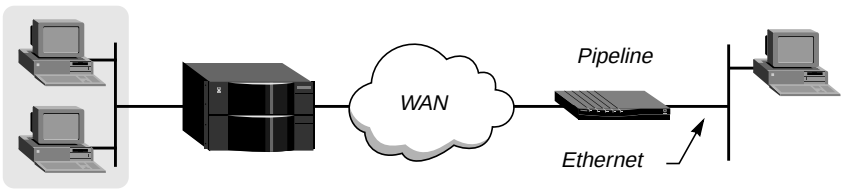


Figure 2-1. Synchronous PPP connection

Following are the commands entered to configure this call, and the system's responses:

```
admin> read answer
ANSWER-DEFAULTS read
admin> set ppp receive-auth = any-ppp-auth
admin> write
ANSWER-DEFAULTS written
admin> new connection phani
CONNECTION/phani read
admin> set active = yes
admin> set encapsulation-protocol = ppp
admin> set dial-number = 1212
admin> set ip remote-address = 10.2.3.31/24
admin> set ppp send-auth-mode = chap-ppp-auth
admin> set ppp send-password = remotepw
admin> set ppp recv-password = localpw
admin> write
CONNECTION/phani written
```

where *remotepw* is the password sent to the remote device for a dial-out connection, and *localpw* is the password expected from the remote device for an inbound connection. The dial-number, send-auth-mode, and send-password parameters are used only for dialing out.

Example of an asynchronous PPP connection

The connection shown in Figure 2-2 uses PPP encapsulation and PAP or CHAP authentication. The calling device is a modem. This is a single-channel asynchronous PPP call.

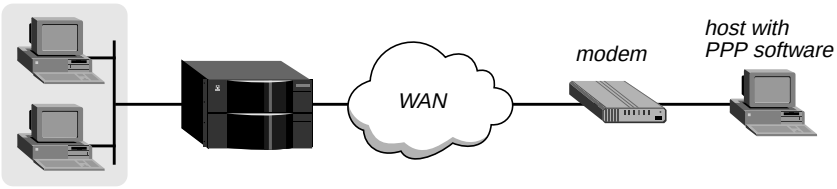


Figure 2-2. Asynchronous PPP connection

Following are the commands entered to configure this call, and the system's responses:

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

admin> set enabled = yes

admin> write
TERMINAL-SERVER written

admin> read answer
ANSWER-DEFAULTS read

admin> set ppp receive-auth = any-ppp-auth

admin> write
ANSWER-DEFAULTS written

admin> new connection carlos
CONNECTION/carlos read

admin> set active = yes

admin> set encapsulation-protocol = ppp

admin> set ppp recv-password = localpw

admin> write
CONNECTION/carlos written
```

where *localpw* is the password expected from the remote device for an inbound connection. For related information, see Appendix A, "Authentication Methods."

Configuring MP connections

Multilink Protocol (MP) uses the encapsulation defined in RFC 1990. MP enables the MAX TNT to interact with MP-compliant equipment from other vendors to use multiple channels for a call. Both sides of the connection must support MP encapsulation.

PPP Answer-Defaults and Connection profile settings also apply to MP connections. If you configure an MP connection and the MAX TNT cannot successfully negotiate the connection, it falls back to single-channel PPP (see "Configuring PPP connections" on page 2-4).

Following are the Connection profile parameters related to MP connections. The MP options are shown with their default settings.

```
CONNECTION station
   encapsulation-protocol = mp
   mp-options
      base-channel-count = 1
```

```
minimum-channels = 1
maximum-channels = 2
```

The minimum and maximum number of channels available to any multi-channel PPP call depend on these parameters. Settings in a Connection profile override the Answer-Defaults settings. When a call is received, the MAX TNT authenticates the first channel of the call and then uses the parameters in the caller's Connection profile to determine the maximum and minimum settings.

The base channel count is the number of channels to use for an MP connection. MP does not support Dynamic Bandwidth Allocation, so the number of channels is fixed for the duration of the session. For example, if you specify a base channel count of 3, the MAX TNT establishes the connection with three channels and maintains those channels until the call is terminated.

The base channel count for an MP call must be greater than or equal to the minimum count and less than or equal to the maximum count. For optimum performance, both sides of a connection should set the base-channel-count, minimum-channel-count, and maximum-channel-count parameters to the same values. For example, the MP connection shown in Figure 2-3 is allocated two channels. It uses MP encapsulation and CHAP authentication.

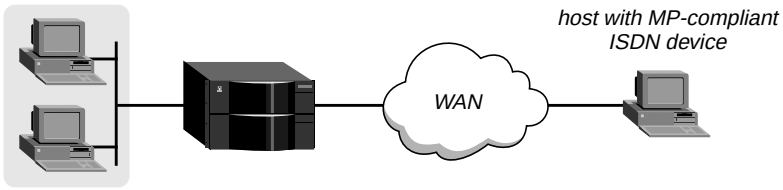


Figure 2-3. Multilink Protocol (MP) connection

Following are the commands entered to configure this call, and the system's responses:

```
admin> read answer
ANSWER-DEFAULTS read
admin> set ppp receive-auth = any-ppp-auth
admin> write
ANSWER-DEFAULTS written
admin> new connection kory
CONNECTION/kory read
admin> set active = yes
admin> set encapsulation-protocol = mp
admin> set dial-number = 1212
admin> set ip remote-address = 10.10.1.2/29
admin> set ppp send-auth-mode = chap-ppp-auth
admin> set ppp send-password = remotepw
admin> set ppp recv-password = localpw
admin> set mp base-channel-count = 2
admin> write
CONNECTION/kory written
```

where *remotepw* is the password sent to the remote device for a dial-out connection, and *localpw* is the password expected from the remote device for an inbound connection.

Configuring MP+ connections

Multilink Protocol Plus (MP+) uses PPP encapsulation with Ascend extensions, as described in RFC 1934. MP+ enables the MAX TNT to connect to another Ascend unit through multiple channels. The criteria for adding or dropping a link are part of the Ascend extensions, and are supported only by Ascend equipment.

PPP and MP Answer-Defaults and Connection profile settings also apply to MP+ connections. If you configure an MP+ connection and the MAX TNT cannot successfully negotiate the connection, it falls back to MP. If the MAX TNT also fails to negotiate an MP connection, it falls back to single-channel PPP (see "Configuring PPP connections" on page 2-4).

Following are the related Connection profile parameters, shown with default values for the MP+ options:

```
CONNECTION station
   encapsulation-protocol = mpp
   mpp-options
      aux-send-password = ""
      dynamic-algorithm = quadratic
      bandwidth-monitor-direction = transmit
      increment-channel-count = 1
      decrement-channel-count = 1
      seconds-history = 15
      add-persistence = 5
      sub-persistence = 10
      target-utilization = 70
```

To specify the base channels of an MP+ connection, you must configure the mp-options subprofile (see "Configuring MP connections" on page 2-6).

How the MAX TNT adds bandwidth

To add bandwidth on demand, the MAX TNT dials additional connections and inverse multiplexes those channels into the call. For information about configuring per-channel add-on numbers that enable the MAX TNT to add bandwidth on demand, see the *MAX TNT Hardware Installation Guide*.

The MAX TNT can reject the request to add bandwidth if there are no more channels available at one or both ends, or if the network is congested. Under either of those conditions, the two ends enter bandwidth-addition-lockout mode, in which neither side can request bandwidth. The restriction prevents both ends from continually trying to add new channels unsuccessfully. The MAX TNT and the Ascend unit at the other end automatically remove the lockout restriction when the condition that caused the lockout changes. Changes typically result from plugging in a new switched-service line, reconfiguration of the line profile, or a switched-service congestion timeout. Once the lockout is removed, either end is free to add bandwidth.

Monitoring bandwidth usage

The bandwidth-monitor-direction parameter specifies whether criteria for adding or dropping links apply to traffic received across the link, transmitted across the link, or both. If both sides of the link have bandwidth-monitor-direction set to None, bandwidth-on-demand is disabled.

Specifying bandwidth increments

You can add channels one at a time or, if the MAX TNT is configured for parallel dialing, in multiples. To configure the unit for parallel dialing, set the parallel-dialing parameter in the System profile. For example, the following command shows that the parallel-dialing parameter in the System profile is set to 2 (the default), which enables two concurrent dial-out calls:

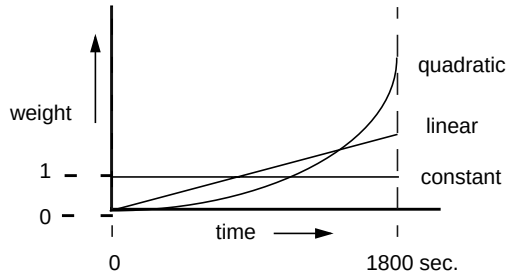
```
admin> get system parallel
parallel-dialing = 2
```

In a Connection profile, the increment-channel-count and decrement-channel-count parameters specify the number of channels the MAX TNT can add and subtract, respectively, at one time. When adding bandwidth, the MAX TNT adds the number of channels specified in the increment-channel-count parameter. When subtracting bandwidth, it subtracts the number of channels specified in decrement-channel-count, dropping the newest channels first.

Specifying the utilization rate that forces a request for bandwidth

To determine when to change the bandwidth allocated to a connection, the MAX TNT uses the parameters shown in Table 2-1, specified in a Connection profile.

Table 2-1. Parameters that force a request for bandwidth

Parameter	Effect
dynamic-algorithm	Specifies an algorithm for calculating average line utilization (ALU) over a certain number of seconds (seconds-history).  Quadratic (the default) gives more weight to recent utilization samples than to older samples within seconds-history. The weighting grows at a quadratic rate. Linear gives more weight to recent utilization samples than to older samples within seconds-history. The weighting grows at a linear rate. Constant gives equal weight to all utilization samples.
seconds-history	Specifies a number of seconds to use as the basis for calculating average line utilization (ALU).
target-utilization	Specifies a percentage of line utilization (default 70%) to use as a threshold when determining when to add or subtract bandwidth.

Specifying how long the utilization rate should persist

The add-persistence parameter specifies a number of seconds for which ALU must persist beyond the target-utilization threshold before the MAX TNT adds bandwidth. Conversely, sub-persistence specifies a number of seconds for which the ALU must persist below the target-utilization threshold before the unit subtracts bandwidth.

ALU spikes

The values for seconds-history, add-persistence, and sub-persistence should smooth out spikes in bandwidth utilization that last for a shorter time than it takes to add capacity. Over T1 lines, the MAX TNT can add bandwidth in less than ten seconds. Over ISDN lines, the unit can add bandwidth in less than five seconds.

Telco charges

Once the MAX TNT adds bandwidth, there is typically a minimum usage charge, after which billing is time-sensitive. The sub-persistence value should be at least equal to the minimum duration charge plus one or two billing time increments. Typically, billing is done to the next multiple of six seconds, with a minimum charge for the first thirty seconds.

Adding or subtracting channels too quickly (less than 10-20 seconds apart) leads to many short duration calls, each of which incur the carrier's minimum charge. In addition, adding or subtracting channels too quickly can affect link efficiency, because the devices on either end have to retransmit data when the link speed changes.

Example of an MP+ configuration

The connection in Figure 2-4 uses MP+ encapsulation with CHAP authentication and configures Dynamic Bandwidth Allocation between the MAX TNT and a MAX unit. (The far-end device must be an Ascend unit.)

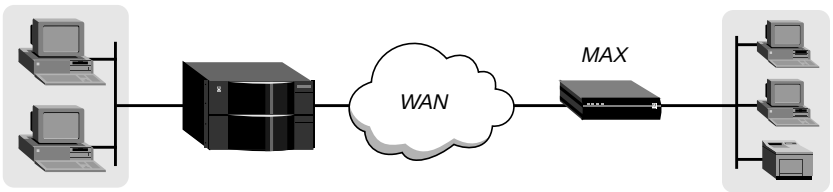


Figure 2-4. Multilink Protocol Plus (MP+) connection

Following are the commands entered to configure this connection, and the system's responses:

```
admin> read answer
ANSWER-DEFAULTS read

admin> set ppp receive-auth = any-ppp-auth

admin> write
ANSWER-DEFAULTS written

admin> new connection moshoula
CONNECTION/moshoula read

admin> set active = yes
```

```
admin> set encapsulation-protocol = mpp
admin> set dial-number = 9-1-333-555-1212
admin> set ip remote-address = 10.10.10.64/24
admin> set ppp send-auth = chap-ppp-auth
admin> set ppp send-password = remotepw
admin> set ppp recv-password = localpw
admin> set mp base-channel-count = 2
admin> list mpp
aux-send-password = ""
dynamic-algorithm = quadratic
bandwidth-monitor-direction = transmit
increment-channel-count = 1
decrement-channel-count = 1
seconds-history = 15
add-persistence = 5
sub-persistence = 10
target-utilization = 70
admin> set bandwidth-monitor-direction = transmit-recv
admin> set increment-channel-count = 2
admin> set seconds-history = 30
admin> set add-persistence = 10
admin> write
CONNECTION/moshoula written
```

where *remotepw* is the password sent to the remote device for a dial-out connection, and *localpw* is the password expected from the remote device for an inbound connection.

Example of a fractional T1 plus switched MP+ connection

An ft1-mpp connection starts as a nailed connection but can use switched channels either to increase the bandwidth as needed or to provide a backup if the nailed channels go down. The maximum number of channels for the ft1-mpp connection is either the maximum channel count for the connection or the number of nailed channels in the specified group, whichever is greater.

The base channels of an ft1-mpp connection are nailed. When a nailed channel is temporarily down, the MAX TNT polls continuously while trying to reestablish that connection. If an outbound packet arrives while the nailed connection is still down, the unit replaces the nailed channel with a switched channel, even if the call is online with more than the minimum number of channels.

If you modify the Connection profile for an ft1-mpp connection, most changes become active only after the call is brought down and then back up, because the connection is primarily a nailed one. However, if you add a group number to the nailed-groups parameter and write the modified profile, the additional channels are available immediately.

After the MP+ (switched) part of the connection was configured as usual, the telco-options subprofile in this example was configured as follows:

```
admin> read connection moshoula
CONNECTION/moshoula read

admin> list telco
answer-originate = ans-and-orig
callback = no
call-type = off
nailed-groups = 1
ft1-caller = no
force-56kbps = no
data-service = 56k-restricted
call-by-call = 0
billing-number = ""
transit-number = ""
expect-callback = no
dialout-allowed = no

admin> set answer-originate = originate-only

admin> set ft1-caller = yes

admin> set call-type = ft1-mpp

admin> set nailed-groups = 1,2

admin> write
CONNECTION/moshoula written
```

The answer-originate and ft1-caller parameters specify that the MAX TNT is the designated caller for the switched part of the connection. Because bandwidth is added on the basis of calculations made at both ends of the connection, make sure that only one of the devices can originate calls for this connection. For more information about these parameters, see “Answering and originating calls” on page 2-17.

The group numbers represent groups of nailed channels configured in a line profile. For information about channel grouping, see the*MAX TNT Hardware Installation Guide*.

Configuring terminal-server connections

The MAX TNT terminal-server software receives asynchronous calls after they have been processed by a digital modem. These calls are typically dialed in by a modem or V.120 TA. If the caller does not send PPP packets immediately, the terminal server starts a login sequence.

For an async PPP call, the terminal-server forwards the call to the router software as soon as it detects a PPP packet. For information about configuring async PPP calls, see “Configuring PPP connections” on page 2-4.

For a login session, each user must have a Connection profile (or external profile) that specifies a name and password to be used in the terminal-server login sequence. In addition, a global Terminal-Server profile defines how these calls are authenticated, and where the call is directed following authentication. For information about both of these issues, see Appendix A, “Authentication Methods,” and Appendix B, “Authorization Options.”

Enabling terminal-server connections

You must enable the terminal-server software to allow the MAX TNT to handle any incoming login-user calls. Following is the related parameter with its default setting:

```
TERMINAL-SERVER
   enabled = no
```

Following is an example that enables the terminal-server software:

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

admin> set enabled = yes

admin> write
TERMINAL-SERVER written
```

Setting defaults for answered calls

Following are the Answer-Defaults parameters related to encapsulation processing and session management of terminal-server connections. The parameters are shown with their default settings:

```
ANSWER-DEFAULTS
   tcp-clear-answer
      enabled = yes
   v120-answer
      enabled= yes
      frame-length = 256
   session-info
      ts-idle-mode = no-idle
      ts-idle-timer = 120
```

V.120 and TCP calls are enabled by default. For V.120 encapsulation, the frame-length parameter specifies the V.120 maximum transmit and receive frame sizes. The value should correspond to these settings in the TA software.

The ts-idle-mode parameter specifies whether the MAX TNT uses the terminal-server idle timer and, if so, whether it monitors traffic in one or both directions to determine when the session is idle. The ts-idle-timer parameter specifies how long the terminal server session can remain idle before the MAX TNT logs out the user and terminates the connection. The next set of commands sets the parameters for bringing down any terminal-server connection after 30 seconds of idle time:

```
admin> read answer-defaults
ANSWER-DEFAULTS read

admin> set session ts-idle-mode = input-output

admin> set session ts-idle-timer = 30

admin> write
ANSWER-DEFAULTS written
```

Configuring modem connections

The following example shows the commands entered to configure a Connection profile for a modem connection that does not use PPP software, and the system's responses:

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

admin> set active = yes

admin> set ppp recv-password = localpw

admin> set session ts-idle-mode = no-idle

admin> set session ts-idle-timer = 60

admin> write
CONNECTION/tom written
```

where Tom is the name and *localpw* is the password expected from the user during the login session. For information about modem expect-send scripts and recommended authentication settings, see Appendix A, "Authentication Methods."

In addition to authentication and session management, certain kinds of modem connections might require changes in the modem-configuration subprofile of the Terminal-Server profile. Following are the parameters that affect how the digital modems negotiate with the calling modem:

```
TERMINAL-SERVER
   modem-configuration
      v42/mnp = will-v42
      max-baud-rate = 33600-max-baud
      modem-transmit-level = -10-db-mdm-trn-level
      cell-mode-first = no
      cell-level = -18-db-cell-level
      7-even = no
```

V42/MNP

The v42/mnp parameter determines how the digital modems negotiate LAPM/MNP error control with the analog modem at the other end of the connection. The MAX TNT can request LAPM/MNP and accept the call anyway if it is not provided (will-v42), request it and drop the call if it is not provided (must-v42), or not use LAPM/MNP error control at all (wont-v42).

Baud-rate

Typically, the digital modems start with the highest possible baud rate (3360) and negotiate down to the rate accepted by the far-end modem. You can adjust the maximum rate to bypass some of the negotiation cycles, provided that no inbound calls will use a baud rate higher than what you specify in the max-baud-rate parameter.

Modem-transmit-level

When a modem calls the MAX TNT, the unit attempts to connect at the transmit attenuation level you specify. This is the amount of attenuation in decibels the MAX TNT should apply to the line, causing the line to lose power when the received signal is too strong. Generally, you

do not need to change the transmit level. But you might need to lower it (increase the negative dB setting of the modem-transmit-level parameter) due to line problems or irregularities.

Cellular modem calls

The cell-mode-first parameter determines whether the MAX TNT first attempts cellular modem or conventional modem negotiation when answering incoming calls. If the first negotiation fails, the MAX TNT attempts the other negotiation. The cell-level parameter determines the gain level of the cellular modem.

Seven-bit even parity

The MAX TNT does not use 7-bit even parity on outbound data unless you set the 7-even parameter to Yes. Most applications do not use this parameter.

In 7-bit communication, each device sends only the first 128 characters in the ASCII character set, because each of these characters can be represented by seven bits or fewer. Parity is a way for a device to determine whether it has received data exactly as the sending device transmitted it. Each device must determine whether it will use even parity, odd parity, or no parity.

The sending device adds the 1s in each string it sends and determines whether the sum is even or odd. Then, it adds an extra bit, called a parity bit, to the string. If even parity is in use, the parity bit makes the sum of the bits even. If odd parity is in use, the parity bit makes the sum of the bits odd. For example, if a device sends the binary number 1010101 under even parity, it adds a 0 (zero) to the end of the byte, because the sum of the 1s is already even. However, if it sends the same number under odd parity, it adds a 1 to the end of the byte in order to make the sum of the 1s an odd number.

The receiving device checks whether the sum of the 1s in a character is even or odd. If the device is using even parity, the sum of the 1s in a character should be even. If the device is using odd parity, the sum of the 1s in a character should be odd. If the sum of the 1s does not equal the parity setting, the receiving device knows that an error has occurred during the transmission of the data.

For the special ASCII characters (128–256), eight bits are necessary to represent the data. In 8-bit communication, no parity bit is used.

Example of a modem setting

Following is an example that shows how to set the maximum negotiable baud rate to 26400:

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

admin> set modem max-baud = 26400

admin> write
TERMINAL-SERVER written
```

Configuring V.120 terminal adapter connections

V.120 terminal adapters (also known as ISDN modems) are asynchronous devices that use CCITT V.120 encapsulation. Following are the values that seem to work best for V.120 operation:

- Maximum information field size for send and receive packets = 260 bytes
- Maximum number of retransmissions (N200) = 3
- Logical link ID (LLI) = 256
- Idle timer (T203) = 30 seconds
- Maximum number of outstanding frames = 7
- Modulo = 128
- Retransmission timer (T200) = 1.5 seconds
- Types of frames accepted = UI, I. (I-type frames are recommended.)
- Call placement: The MAX TNT can receive V.120 calls, but cannot place them.

Note: If the user's dial-in software supports async-to-sync conversion, the Connection profile can be set for PAP or CHAP authentication and the user can access the terminal server by PPP automatic login. In this case, V.120 encapsulation is not required in the Connection profile. For recommended authentication settings for connections using terminal adapters, see Appendix A, "Authentication Methods."

The V.120 device must be correctly configured to place calls to the MAX TNT. The settings required for compatible operation of a V.120 device and the MAX TNT are listed below. Refer to the manual for the V.120 device for information about how to enter these settings.

- V.120 maximum transmit frame size = 260 bytes
- V.120 maximum receive frame size = 260 bytes
- Logical link ID = 256
- Modulo = 128
- Line channel speed = Select 56K if the MAX TNT accepts calls from the V.120 device on a T1 line, or if you are not sure that you have 64-Kbps channel speed end-to-end.

Following is an example configuration for a dial-in connection from a V.120 TA:

```
admin> new conn tom
CONNECTION/tom read
admin> set active = yes
admin> set encap = v120
admin> set ppp recv-password = localpw
admin> set session ts-idle-mode = no-idle
admin> set session ts-idle-timer = 60
admin> write
CONNECTION/tom written
```

where Tom is the name and *localpw* is the password expected from the user during the login session. For information about recommended authentication settings for terminal adapters, see Appendix A, "Authentication Methods."

Configuring TCP-clear connections

TCP-clear is used to support encapsulation performed by the application that runs on top of TCP, which must be understood by both the login host and the caller. As soon as the connection is authenticated, the MAX TNT establishes a TCP connection to the host specified

in the Connection profile. For information about immediate mode, which enables you to set an immediate TCP connection globally rather than on a per-connection basis, see Appendix B, "Authorization Options."

Following are commands entered to configure a TCP-clear connection to a host named techpubs, and the system's responses:

```
admin> new conn richard
CONNECTION/richard read
admin> set active = yes
admin> set ppp recv-password = localpw
admin> set tcp host = techpubs
admin> set tcp port = 23
admin> write
CONNECTION/richard written
```

where Richard is the name and *localpw* is the password expected from the user during the login session. If DNS were not configured, the host's IP address would be required. The port number is the TCP port on the host to use for the connection. A port number of zero means "any port."

Setting telco and session options

Connection profiles contain several general settings that do not fall into the categories already described in this chapter. They apply equally to synchronous or asynchronous connections.

Connection-specific telco options

Following are the telco options in a Connection profile, shown with default settings:

```
CONNECTION station
   telco-options
      answer-originate = ans-and-orig
      callback = no
      call-type = off
      nailed-groups = 1
      ft1-caller = no
      force-56kbps = no
      data-service = 56k-restricted
      call-by-call = 0
      billing-number = ""
      transit-number = ""
      expect-callback = no
      dialout-allowed = no
```

Answering and originating calls

The answer-originate parameter specifies whether the MAX TNT can use this profile to answer incoming calls, dial out, or both. The ft1-caller parameter specifies whether this unit (the MAX TNT) can initiate calls on fractional T1 to add switched channels to a nailed MPP connection (only one side of the connection should have this parameter set to Yes).

The dialout-allowed parameter (at the end of the subprofile) allows or prevents the connection from using the MAX TNT unit's digital modems to dial out.

Using callback

When callback is set to Yes, the MAX TNT hangs up on the caller and immediately dials back to the dial number in this profile.

When expect-callback is set to Yes, the MAX TNT expects the far end to hang up and dial back when it receives a call from the MAX TNT. This is recommended when CLID is required on the far-end unit and Ping or Telnet are in use. For more information about callback, see Appendix A, "Authentication Methods."

Using nailed channels

A nailed connection is a permanent link that is always up as long as the physical connection persists. For a nailed connection, you must set the call-type parameter and specify the number assigned to a group of nailed channels the connection will use.

The call-type parameter is set to Off by default, which means that no nailed channels are used. If the connection uses nailed channels, you can specify ft1 (all nailed channels), ft1-mpp (nailed channels that can be augmented with switched channels if bandwidth is needed), or ft1-bo (a nailed connection that can use switched channels both for additional bandwidth and for a backup method of reaching the site if the nailed connection is down).

The nailed-groups parameter is set to 0 (zero) by default. For a nailed connection, you set it to the group number assigned to channels. For example:

```
admin> read connection karl
CONNECTION/karl read

admin> set telco call-type = ft1

admin> set nailed-groups = 6

admin> write
CONNECTION/karl written
```

You can specify multiple groups by separating the numbers with a comma. This combines the groups of nailed channels to create For example:

```
admin> set nailed-groups = 3,4
```

For an example ft1-mpp configuration, see "Example of a fractional T1 plus switched MP+ connection" on page 2-11. For information about configuring groups of nailed channels, see the *MAX TNT Hardware Installation Guide*.

Specifying the data service

The data-service parameter specifies the bandwidth and service of outgoing calls. The data service depends on what is supported in the far-end device and the telco end-to-end path. For users, the associated bandwidth is the most important aspect of this setting. Note that data services with bandwidth greater than 64K do not apply to connections that do not use switched channels.

When the MAX TNT initiates a dialout, it requests a data service and bandwidth rate, and the answering end rate-adapts. If the answering side cannot match the requested bandwidth or if the switch cannot provide the requested service, the call attempt fails.

The `modem` data service setting is related only to dialouts, it is not required for inbound modem calls.

Using billing numbers

Billing-number can specify a billing number for charges incurred on the line. Your carrier can provide a billing number for use in sorting your bill. For example, each department might require its own billing number. The billing number can contain up to 24 characters.

Session management

Once the call has been answered and the session established, the MAX TNT manages the session. Session management can include packet filtering, bringing down an inactive connection, and sending accounting information about the session to a RADIUS or TACACS+ accounting server. Following are the session-options Connection profile parameters, shown with their default values:

```
CONNECTION station
   session-options
      call-filter = ""
      data-filter = ""
      filter-persistence = no
      idle-timer = 120
      ts-idle-mode = no-idle
      ts-idle-timer = 120
      backup = ""
      max-call-duration = 0
```

Applying call or data filters to a session

For information about defining and applying filters and firewalls to a WAN connection, see Chapter 7, “Packet and Route Filters.”

Timing out inactive sessions

The idle-timer and ts-idle-timer parameters specify how long a network or terminal server session may remain idle before the MAX TNT drops the connection.

The idle-timer applies to sessions where the data on the WAN is packetized and passes through the MAX TNT router. The ts-idle-timer applies to sessions where the data on the WAN is not packetized and is processed the terminal server. See “Setting defaults for answered calls” on page 2-13 for more information about the ts-idle-timer.

For related information about dropping idle connections, see Chapter 7, “Packet and Route Filters.”

Specifying a backup connection when a nailed connection fails

Backup specifies the name of a backup Connection profile for a nailed connection. Its intended use is for providing a backup in the event that the far-end device goes out of service, in which case the backup call is made. The backup parameter is not intended to provide alternative lines for getting to a single destination.

Specifying a maximum call duration

The value of the max-call-duration parameter is the number of minutes of connect time before a call will be dropped. A value of zero disables the connection timer. The connection is checked once per minute, so the actual time of the call will be slightly longer (usually less than a minute longer) than the actual time you set.

Session accounting

The MAX TNT supports RADIUS and TACACS+ accounting. Only RADIUS accounting can be specified on a per-connection basis. Following are the relevant parameters for both RADIUS and TACACS+, shown with their default values:

```
EXTERNAL-AUTH
   acct-type = RADIUS

   rad-acct-client
      acct-server-1 = 0.0.0.0
      acct-server-2 = 0.0.0.0
      acct-server-3 = 0.0.0.0
      acct-port = 0
      acct-src-port = 0
      acct-key = ""
      acct-timeout = 0
      acct-sess-interval = 0
      acct-id-base = acct-base-10

   tacplus-acct-client
      acct-server-1 = 0.0.0.0
      acct-server-2 = 0.0.0.0
      acct-server-3 = 0.0.0.0
      acct-port = 0
      acct-src-port = 0
      acct-key = ""

CONNECTIONS
   usrRad-options
      acct-type = global
      acct-host = 0.0.0.0
      acct-port = 1646
      acct-key = ""
      acct-timeout = 1
      acct-id-base = acct-base-10
```

Using RADIUS

You can send accounting statistics for a specific connection to the server specified in the External-Auth profile (global), the server specified in the usrRad-options subprofile (local), or both. When the accounting policy includes the "local" RADIUS accounting server (the one specified in a Connection profile), you can specify the server's address (acct-host), a UDP port on that server (acct-port), a password (acct-key), timeout, and numeric base (10 or 16).

For information about using RADIUS, see the *MAX TNT RADIUS Guide*.

Using TACACS+

For information about configuring the TACACS+ server for session accounting, see the TACACS+ documentation. This section shows how to configure the MAX TNT to send accounting statistics about all WAN sessions to the specified TACACS+ server. Following is an example, with explanations of the parameters:

```
admin> read external-auth
EXTERNAL-AUTH read

admin> list
auth-type = None
acct-type = none
rad-serv-enable = no
rad-auth-client = { 0.0.0.0 0.0.0.0 0.0.0.0 0 0 "" no 0 no 0 yes yes
no+
rad-acct-client = { 0.0.0.0 0.0.0.0 0.0.0.0 0 0 "" 0 0 acct-base-10
0 }
rad-auth-server = { 0 no rad-serv-attr-any [ 0.0.0.0 0.0.0.0
0.0.0.0 0+
tac-auth-client = { 0.0.0.0 0.0.0.0 0.0.0.0 0 0 "" 0 }
tacplus-auth-client = { 0.0.0.0 0.0.0.0 0.0.0.0 0 0 "" 0 0 0 }
tacplus-acct-client = { 0.0.0.0 0.0.0.0 0.0.0.0 0 0 "" }
local-profiles-first = yes

admin> set acct-type = tacacsplus

admin> list tacplus-acc
acct-server-1 = 0.0.0.0
acct-server-2 = 0.0.0.0
acct-server-3 = 0.0.0.0
acct-port = 0
acct-src-port = 0
acct-key = ""

admin> set acct-server-1 = 10.1.2.3

admin> set acct-server-2 = 10.2.3.4

admin> set acct-port = 5000

admin> set acct-key = Ascend

admin> write
EXTERNAL-AUTH written
```

In the External-Auth profile, the acct-type parameter specifies the type of accounting to be performed: RADIUS or TACACS+.

Each acct-server-*n* parameter can specify the IP address of one TACACS+ server. The MAX TNT first tries to connect to server #1. 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.

The acct-port parameter specifies the destination port to use to access the server. The port specified must match the port used by the TACACS+ daemon.

The acct-src-port parameter specifies the source port to use to access the server. If zero, the source port is selected from the non-privileged port range (1024-2000).

The acct-key is an accounting access key shared with the server.

Frame Relay Configuration

3

This chapter covers the following topics:

Using the MAX TNT as a Frame Relay concentrator

In a Frame Relay backbone, every access line connects directly to a Frame Relay switch. In the past, most connections to the Frame Relay network were relatively high speed, such as full T1 or E1 lines. With recent changes in Frame Relay pricing, providers now want to concentrate many low-speed dial-in connections into one high-speed nailed connection to a Frame Relay switch. You can configure the MAX TNT as a Frame Relay concentrator, as shown in Figure 3-1. It then accepts incoming dial-in connections (shown at the right of Figure 3-1) as usual and forwards them out to the Frame Relay network.

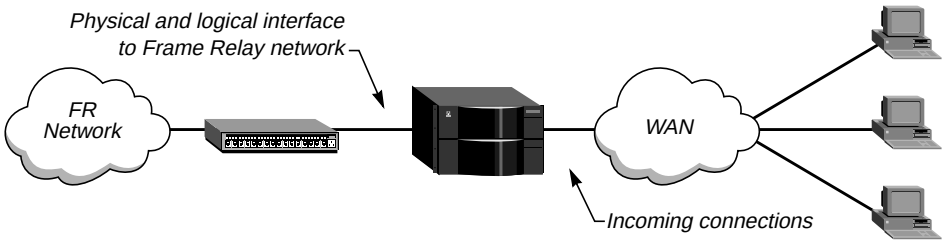


Figure 3-1. The MAX TNT operating as a Frame Relay concentrator

The main elements of this configuration are:

- A physical interface to the Frame Relay network (nailed-up WAN channels).
- A logical interface to the Frame Relay network (defined in a Frame-Relay profile).
- Connections that are forwarded to the Frame Relay network (defined in Connection profiles or RADIUS).

Kinds of physical network interfaces

The MAX TNT uses a nailed-up channels to connect to a Frame Relay switch. For information about configuring nailed channels in a line profile, see the *MAX TNT Hardware Installation Guide*.

Kinds of logical interfaces (data links) to a Frame Relay network

You can configure the MAX TNT as either Data Communications Equipment (DCE) or Data Terminal Equipment (DTE).

DCE refers to a device that connects a computer to a data communications service, such as Frame Relay. (A Frame Relay switch performs DCE operations.)

DTE refers to a computer that is the source or destination of the data traversing the service (for example, a user's PC accessing a router or switch).

User-to-Network Interface (UNI) refers to the interface between user equipment (a router, for example, or perhaps a computer) and the network. When you configure a MAX TNT interface for DCE operations, the interface is called UNI-DCE. When you configure it for DTE operations, the interface is called UNI-DTE.

UNI-DCE interfaces

When you configure a UNI-DCE interface in the MAX TNT, it operates on that interface as a Frame Relay router communicating with Customer Premise Equipment (CPE) that performs DTE operations, as shown in Figure 3-2. To the DTE devices, it appears as a Frame Relay network end point. For an example of how to configure the Frame-Relay profile for this kind of logical interface, see "Example of a UNI-DCE interface" on page 3-6.

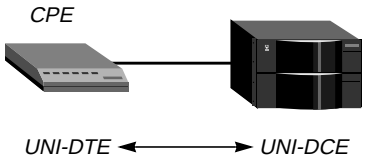


Figure 3-2. DCE interface connecting to DTE

UNI-DTE interfaces

When you configure a UNI-DTE interface in the MAX TNT, it operates on that interface as a computer accessing a Frame Relay switch, as shown in Figure 3-3. In this kind of configuration, the MAX TNT sends data out to the Frame Relay switch and performs the DTE functions specified for link management.

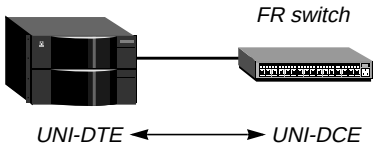


Figure 3-3. DTE interface connecting to DCE (a switch)

For an example of how to configure the Frame-Relay profile for this kind of interface, see "Example of a UNI-DTE interface" on page 3-7.

Note: When it has a UNI-DTE interface, the MAX TNT is able to query the device at the other end of the link about the status of the DLCIs on that interface. If a DLCI becomes unusable, and its Connection profile specifies a Backup connection, the MAX TNT dials the Connection profile specified by the Backup parameter in the session-options subprofile. For details, see the description of the Backup parameter in the *MAX TNT Reference Guide*.

Kinds of dial-in connections that use Frame Relay

When the MAX TNT is configured with a data link to the Frame Relay network, you can configure many dial-in connections that are destined to be forwarded out to the Frame Relay network. Dial-in connections that can make use of the Frame Relay capabilities of the MAX TNT include gateway connections, Frame Relay circuits, and redirect connections (the last type is rarely used).

Gateway connections

In a configuration known as gateway mode, the MAX TNT receives an incoming PPP call, examines the destination IP address, and brings up the appropriate Connection profile to reach the destination, as usual. If the Connection profile specifies Frame Relay encapsulation, the

Frame-Relay profile, and a DLCI number, the MAX TNT encapsulates the packets in Frame Relay (RFC 1490) and forwards the data stream out to the Frame Relay switch, assigning the specified DLCI. The Frame Relay switch uses the DLCI to route the frames on the Frame Relay network.

Frame Relay circuits

A Frame Relay circuit is a Permanent Virtual Circuit (PVC) segment that consists of two DLCI end points and possibly two Frame Relay profiles. A circuit requires two and only two DLCI numbers: data is dropped if the circuit has only one DLCI, and if more than two are defined, only two are used. Circuits are defined in two Connection profiles. Data coming in on the DLCI configured in the first Connection profile is switched to the DLCI configured in the second one.

Redirect connections

When the MAX TNT receives an incoming PPP call for which the session options specify FR Direct, it ignores the destination IP address in the packets from the dial-in client. Instead, it uses the FR DLCI specified in the session options to forward the packet. For redirect connections, the MAX TNT doesn't route packets from the client in the usual sense. It simply forwards them along to the Frame Relay network and assumes that another device will route the packets on the basis of the destination IP address. This is known as redirect mode, and is not commonly used.

Configuring the link to the Frame Relay network

To define the interface between the MAX TNT and the Frame Relay network, you configure a Frame-Relay profile. Following are the parameters contained in a Frame-Relay profile, shown with their default settings:

```
FRAME-RELAY fr-name
   fr-name* = ""
   active = no
   nailed-up-group = 32769
   nailed-mode = ft1
   called-number-type = 2
   switched-call-type = 56k-restricted
   phone-number = ""
   billing-number = ""
   transit-number = ""
   Link-Mgmt = none
   call-by-call-id = 0
   n391-val = 6
   n392-val = 3
   n393-val = 4
   t391-val = 10
   t392-val = 15
   MRU = 1532
   fr-type-val = dte
   dceN392-val = 3
   dceN393-val = 4
```

The Frame-Relay profile name (fr-name) must be unique and cannot exceed 15 characters. The active parameter must be set to Yes to make this profile available for use.

The following subsections discuss various options you specify in the Frame-Relay profile. This section closes with examples of configured profiles.

Defining the nailed connection to the Frame Relay network

Nailed is the default for Frame Relay connections. When the call type is nailed, dial numbers and other telco options (called-number-type, switched-call-type, phone-number, billing-number, and transit-number) do not apply, and the nailed-mode and nailed-up-group parameters are required.

The nailed-up-group parameter specifies the group number assigned to the nailed channels. The nailed-mode parameter is set to ft1 by default, indicating that the connection uses all nailed channels. If you are using a combination of nailed and switched channels, you can specify ft1-mpp or ft1-bo.

You can specify a switched call-type if the Frame Relay switch allows dial-in. However, Frame Relay networks currently have no dial-out connection capability. The two types of data service available are 64K and 56K.

Specifying the switch's link management protocol

The Link-Mgmt settings are none (no link management), ansi-t1.617 (the link management protocol used by the switch is ANSI T1.617), and ccitt-q.933a (the link management protocol used by the switch is CCITT Q.933 Annex A).

Setting Frame Relay timers and event counts

You configure Frame Relay timers and event counts by means of the parameters in Table 3-1:

Table 3-1. Frame Relay timers and event counts

Parameter	Effect
n391-val	Specifies the interval at which the MAX TNT requests a Full Status Report (between 1 and 255 seconds). It doesn't apply when fr-type-val is DCE.
n392-val	Specifies the number of errors during DTE N393 monitored events which cause the user side to declare the network side procedures inactive. Its value should be less than DTE N393 (between 1 and 10). It doesn't apply when fr-type-val is DCE.
n393-val	Specifies the DTE monitored event count (between 1 and 10). It doesn't apply when fr-type-val is DCE.
t391-val	Specifies the Link Integrity Verification polling timer (between 5 and 30 seconds). Its value should be less than T392. It doesn't apply when fr-type-val is DCE.

Table 3-1. Frame Relay timers and event counts (continued)

Parameter	Effect
t392-val	Specifies the time for Status Enquiry messages (between 5 and 30 seconds). An error is recorded if no Status Enquiry is received within T392 seconds. This parameter doesn't apply when fr-type-val is DTE.
dceN392-val	Specifies the number of errors during DCE N393 monitored events which causes the network side to declare the user side procedures inactive. Its value should be less than dceN393-val (between 1 and 10). It doesn't apply when fr-type-val is DTE.
dceN393-val	Specifies the DCE monitored event count (between 1 and 10). It doesn't apply when fr-type-val is DTE.

Specifying the maximum receive units

The MRU parameter specifies the maximum number of bytes the MAX TNT can receive in a single packet across this link. Usually the default of 1532 is the right setting, unless the far-end device requires a lower number.

Specifying the kind of Frame Relay interface

The fr-type-val parameter specifies the kind of logical interface between the MAX TNT and the Frame Relay network on this data link. You can specify UNI-DCE or UNI-DTE. For information about how these interfaces differ, see "Kinds of logical interfaces (data links) to a Frame Relay network" on page 3-2.

Example of a UNI-DCE interface

In the data link shown in Figure 3-4, the MAX TNT has a nailed connection to Customer Premise Equipment (CPE) and a DCE interface to that equipment.

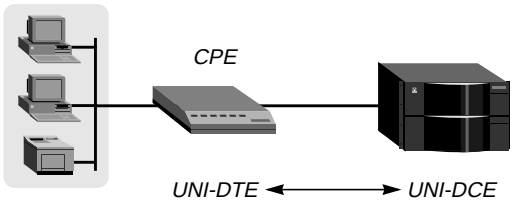


Figure 3-4. Frame-Relay profile defines a UNI-DCE interface

The following example shows the commands used to configure the Frame-Relay profile for the UNI-DCE interface shown in Figure 3-4, and the system's responses:

1 Create a Frame-Relay profile and list its contents:

```
admin> new frame-relay att-dce
FRAME-RELAY/att-dce read

admin> list
fr-name* = att-dce
```

```
active = no
nailed-up-group = 32769
nailed-mode = ft1
called-number-type = 2
switched-call-type = 56k-restricted
phone-number = ""
billing-number = ""
transit-number = ""
Link-Mgmt = none
call-by-call-id = 0
n391-val = 6
n392-val = 3
n393-val = 4
t391-val = 10
t392-val = 15
MRU = 1532
fr-type-val = dte
dceN392-val = 3
dceN393-val = 4
```

2 Activate the profile and set the fr-type-val to DCE:

```
admin> set active = yes
admin> set fr-type-val = dce
```

3 Specify the group number of the nailed channels to use (the number assigned to a group of nailed channels in a line profile, as described in the *MAX TNT Hardware Installation Guide*. For example:

```
admin> set nailed-up-group = 6
```

4 Specify the link management protocol and its configuration parameters. For example:

```
admin> set link-mgmt = ansi
admin> set dceN392 = 3
admin> set dceN393 = 4
admin> set t392 = 15
```

5 Write the Frame-Relay profile:

```
admin> write
FRAME-RELAY/att-dce written
```

Example of a UNI-DTE interface

In the data link shown in Figure 3-5, the MAX TNT has a nailed connection to a Frame Relay switch configured as a DCE, and will be configured with a UNI-DTE interface to that switch.

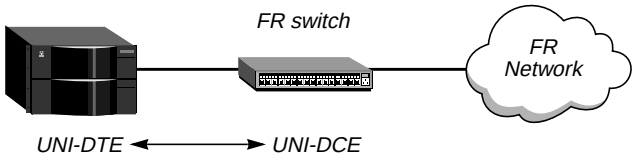


Figure 3-5. Frame-Relay profile defines a UNI-DTE interface

The following example shows the commands entered to configure the Frame-Relay profile for this interface, and the system's responses:

1 Create a Frame-Relay profile, activate it, and set the fr-type-val to DTE:

```
admin> new frame-relay att-dte
FRAME-RELAY/att-dte read

admin> set active = yes

admin> set fr-type-val = dte
```

2 Specify the group number of the nailed channels to use (the number assigned to a group of nailed channels in a line profile, as described in the *MAX TNT Hardware Installation Guide*. For example:

```
admin> set nailed-up-group = 7
```

3 Specify the link management protocol and its configuration parameters. For example:

```
admin> set link-mgmt = ccitt

admin> set n391 = 6

admin> set n392 = 3

admin> set n393 = 4

admin> set t391 = 10
```

4 Write the Frame-Relay profile:

```
admin> write
FRAME-RELAY/att-dte written
```

Configuring dial-in connections that use Frame Relay

Dial-in connections that are forwarded or routed over the Frame Relay link use parameters in the Answer-Defaults and Connection profiles. By default, the Answer-Defaults profile enables all types of encapsulation and routing, and the basic call-setup parameters use the lowest common denominator settings. Following are the related parameters, shown with their default settings:

```
ANSWER-DEFAULTS
   fr-answer
      enabled = yes
   ppp-answer
      enabled = yes

CONNECTION station
   encapsulation-protocol = frame-relay
   fr-options
      frame-relay-profile = ""
      dlci = 16
      circuit-name = ""
      fr-direct-enabled = no
      fr-profile = ""
      fr-dlci = 16
   ip-options
      ip-routing-enabled = yes
      remote-address = 10.1.2.3/24
```

The Connection profile parameters are used in different combinations to define three kinds of connections that may use the Frame Relay data link.

The following subsections discuss various options you specify in Connection profiles and provide examples of the configured profiles.

Example of a gateway connection

Gateway connections require Frame Relay encapsulation, a Frame-Relay profile name, and a DLCI. Your Frame Relay provider tells you the DLCI to assign to each connection.

The far end specified in a Frame Relay encapsulated Connection profile lies at the end of a Permanent Virtual Circuit (PVC), whose first hop is known by the DLCI named in the Connection profile. The MAX TNT does not allow you to enter duplicate DLCIs, except when they are carried by separate physical links specified in different Frame Relay profiles.

Figure 3-6 show callers who dial into the MAX TNT to reach a distant IP network across the Frame Relay network. The MAX TNT communicates with the Frame Relay switch by means of a Frame-Relay profile named ATT-DTE.

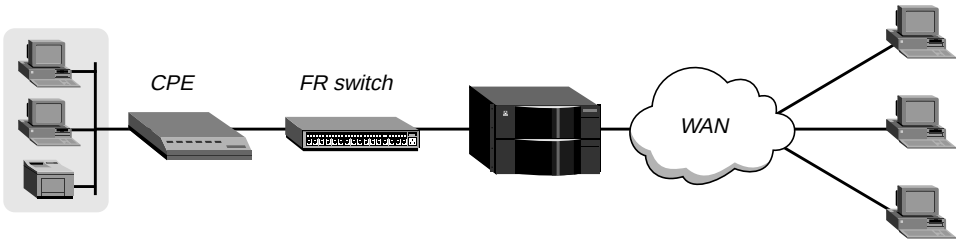


Figure 3-6. Frame Relay gateway connections

The following example shows the commands entered to configure a Connection profile for one of the dial-in hosts in Figure 3-6, and the system's responses:

1 Create a Connection profile and specify Frame Relay encapsulation:

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

admin> set active = yes

admin> set encaps = frame-relay
```

2 Enable IP routing and specify the address of the remote IP router:

```
admin> set ip ip-routing-enabled = yes

admin> set ip remote-address = 10.1.2.3/24
```

3 Open the fr-options subprofile. Then, specify the name of the Frame-Relay profile with a nailed connection to the Frame Relay switch, and a DLCI number:

```
admin> list fr
frame-relay-profile = ""
dlci = 16
circuit-name = ""
fr-direct-enabled = no
fr-profile = ""
fr-dlci = 16
```

```
admin> set frame-relay-profile = att-dte

admin> set dlci = 55
```

4 Write the Connection profile:

```
admin> write
CONNECTION/ted written
```

Example of a Frame Relay circuit

A circuit is a PVC segment configured in two Connection profiles. Data coming in on the DLCI configured in one Connection profile is switched to the DLCI configured in the other. Data is dropped if the circuit has only one DLCI. If more than two Connection profiles specify the same circuit name, only two of them are used.

In a circuit, both Connection profiles must specify Frame Relay encapsulation and the same circuit name. Each profile must specify a unique DLCI. The MAX TNT does not allow you to enter duplicate DLCIs, except when they are carried by separate physical links specified in different Frame Relay profiles.

Figure 3-7 shows a circuit between UNI-DCE and UNI-DTE interfaces. A circuit between any two interfaces within the MAX TNT would be configured in much the same way. In Figure 3-7, the interface to CPE A is UNI-DCE, and the sample Frame Relay profile for that interface is named ATT-DCE. The interface to the Frame Relay switch in Figure 3-7 is UNI-DTE, and the sample Frame-Relay profile for that interface is named ATT-DTE.

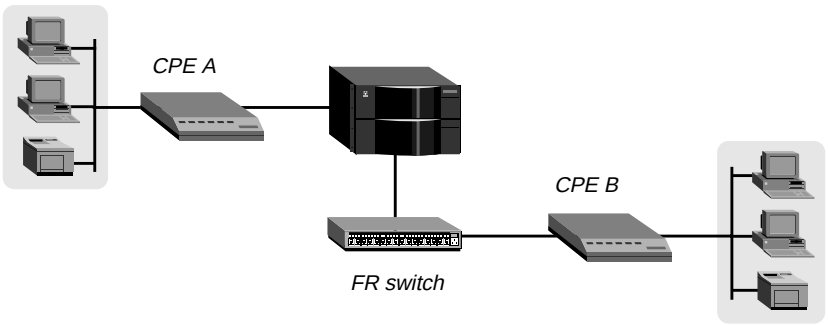


Figure 3-7. Frame Relay circuit

The following example shows the commands entered to configure the Connection profiles for a Frame Relay circuit, and the system's responses.

1 Create the first Connection profile, and specify Frame Relay encapsulation:

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

admin> set active = yes

admin> set encaps = frame-relay
```

2 Open the fr-options subprofile. Then, specify the name of the Frame-Relay profile with a nailed connection to the Frame Relay switch, a DLCI assigned by the Frame Relay administrator, and a name for the Frame Relay circuit:

```
admin> list fr
frame-relay-profile = ""
```

```
dlci = 16
circuit-name = ""
fr-direct-enabled = no
fr-profile = ""
fr-dlci = 16
```

```
admin> set frame-relay-profile = att-dce
```

```
admin> set dlci = 18
```

```
admin> set circuit-name = circuit-1
```

3 Write the Connection profile:

```
admin> write
CONNECTION/victor written
```

4 Create the second Connection profile, and specify Frame Relay encapsulation:

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

```
admin> set active = yes
```

```
admin> set encaps = frame-relay
```

5 Open the fr-options subprofile. Then, specify the name of the Frame-Relay profile with a nailed connection to the Frame Relay switch, a DLCI assigned by the Frame Relay administrator, and a name for the Frame Relay circuit:

```
admin> list fr
frame-relay-profile = ""
dlci = 16
circuit-name = ""
fr-direct-enabled = no
fr-profile = ""
fr-dlci = 16
```

```
admin> set frame-relay-profile = att-dte
```

```
admin> set dlci = 23
```

```
admin> set circuit-name = circuit-1
```

6 Write the Connection profile:

```
admin> write
CONNECTION/marty written
```

Example of a Frame Relay redirect connection

In a redirect connection, the MAX TNT associates a Frame Relay PVC with multiple Connection profiles. It does so in the fr-options subprofile, by enabling Frame Relay Direct, specifying a Frame-Relay profile, and setting the fr-dlci parameter to a DLCI for the PVC endpoint. Any packet coming into the MAX TNT on these connections gets switched out on the specified DLCI. In this mode, the MAX TNT allows multiple Connection profiles to specify the same DLCI.

In Frame Relay redirect mode, the MAX TNT ignores the destination of the affected packets. It assumes that some device at the far end of the PVC makes the routing decisions. However, the

Connection profile must use IP routing to enable the MAX TNT to route data back to the client.

Note: A Frame Relay redirect connection is not a full-duplex tunnel between the PPP dial-in and the switch. The IP packets coming back from the Frame Relay switch are handled by the MAX TNT router software, so they must contain the PPP caller's IP address to be routed correctly back across the WAN.

Figure 3-8 shows two incoming PPP connections which are redirected out to the Frame Relay network. In this example, the MAX TNT communicates with a Frame Relay switch by means of a Frame-Relay profile named ATT-DTE. Both redirect connections (shown at the right of Figure 3-8) use the same DLCI number (72).

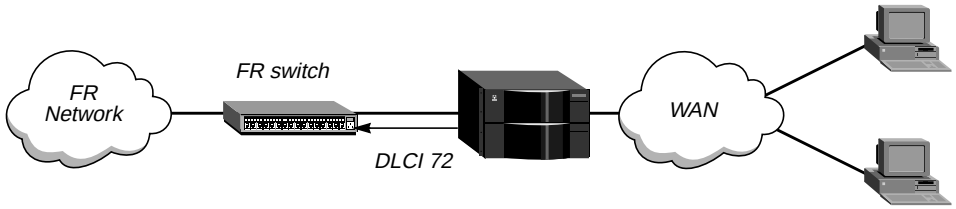


Figure 3-8. Frame Relay redirect connections using the same DLCI

The following example shows the commands entered to configure Connection profiles for the Frame Relay redirect connections in Figure 3-8, and the system's responses:

1 Create the first Connection profile, and specify PPP encapsulation:

```
admin> new conn caller-1
CONNECTION/caller-1 read

admin> set active = yes

admin> set encaps = ppp
```

2 Make sure that IP routing is enabled.

```
admin> set ip ip-routing-enabled = yes
```

3 Open the fr-options subprofile. Then, enable Frame Relay direct, and use the fr-profile parameter to specify the name of the Frame-Relay profile.

```
admin> list fr
frame-relay-profile = ""
dlci = 16
circuit-name = ""
fr-direct-enabled = no
fr-profile = ""
fr-dlci = 16

admin> set fr-direct-enabled = yes

admin> set fr-profile = att-dte
```

4 Assign a DLCI to be available for this redirect connection (it can already be in use by other redirect Connection profiles):

```
admin> set fr-dlci = 72
```

5 Write the Connection profile:

```
admin> write
CONNECTION/caller-1 written
```

6 Create the second Connection profile, and specify PPP encapsulation:

```
admin> new conn caller-2
CONNECTION/caller-2 read

admin> set active = yes

admin> set encaps = ppp
```

7 Make sure that IP routing is enabled.

```
admin> set ip ip-routing-enabled = yes
```

8 Open the fr-options subprofile. Then, enable Frame Relay direct, and use the fr-profile parameter to specify the name of the Frame-Relay profile:

```
admin> list fr
frame-relay-profile = ""
dlci = 16
circuit-name = ""
fr-direct-enabled = no
fr-profile = ""
fr-dlci = 16

admin> set fr-direct-enabled = yes

admin> set fr-profile = att-dte
```

9 Assign a DLCI to be available for this redirect connection (it can already be in use by other redirect Connection profiles):

```
admin> set fr-dlci = 72
```

10 Write the Connection profile:

```
admin> write
CONNECTION/caller-2 written
```

IP Router Configuration

4

This chapter discusses the following topics:

Introduction

The MAX TNT supports a wide range of IP routing configuration options. At a minimum, IP routing requires two interfaces configured for IP. Figure 4-1 shows a MAX TNT that routes IP packets between WAN interfaces (connections) and a LAN interface.

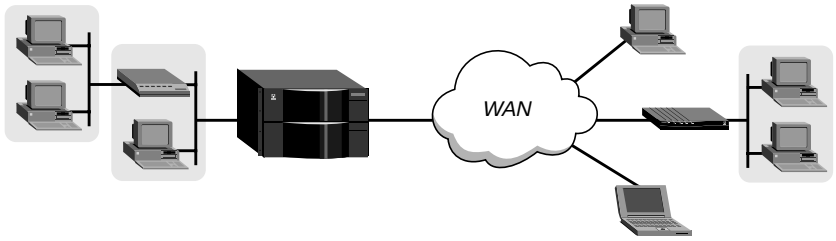


Figure 4-1. Simple IP routing configuration

What are your options?

The MAX TNT configuration options for IP routing can be separated into four areas of activity, as shown in Table 4-1:

Table 4-1. IP configuration overview

Configuration activity	Options
LAN interfaces	You can assign each local interface one or more IP addresses. In addition, you can enable the unit to respond to ARP requests for remote hosts that have an IP address on the local network, and specify how the MAX TNT handles dynamic route updates on each local interface.
IP router	The router software has many possible configuration settings that determine how it handles routes, which source address it uses, how WAN clients access DNS servers, how pools of local addresses are allocated for dynamic assignment to dial-in hosts, and so forth.
WAN interfaces	The MAX TNT connects across the WAN to a remote IP network or host, and different routing policies or preferences might be appropriate, depending on the caller.
Static IP routes	If the router cannot find a route to a packet's destination address, it either drops the packet or sends it to the configured default route. At a minimum, most sites configure a default route in the MAX TNT. Many sites choose to provide multipath static routes to balance traffic to a site across several connections.

For information about OSPF routing, see Chapter 5, "OSPF Router Configuration." For information about multicast forwarding, see Chapter 6, "Multicast Forwarding."

Which profiles do you need?

To configure the MAX TNT as an IP router, you use the profiles shown in Table 4-2:

Table 4-2. Profiles used for IP routing configuration

Profile	Description
IP-Interface	Configure local Ethernet interfaces for IP. Each packet-handling slot card operates as a router subsystem with its own local interface table and route cache. Each LAN interface has its own IP address. If you are using numbered interfaces, you can assign multiple IP addresses to a single interface by creating additional IP-Interface profiles that specify the same physical slot address.
IP-Global	The MAX TNT master shelf-controller manages the global interface and routing tables and defines general router behavior. Most of the system routing options are configured in this profile.
Connection	WAN interfaces can be configured locally, in Connection profiles, or in a central location on an external authentication server, such as RADIUS, TACACS, or TACACS+. For details about RADIUS, see the *MAX TNT RADIUS Guide*. For information about TACACS or TACACS+, see the documentation for the server.
IP-Route	Each IP-Route profile defines a static route.

In addition to these IP-specific profiles, you might also need to change some settings in the Answer-Defaults profile. By default, Answer-Defaults enables IP calls, but does not enable dynamic address assignment. To assign addresses to dial-in hosts from a pool of local addresses, you must enable address assignment in this profile.

For information about setting IP packet filters, see Chapter 7, “Packet and Route Filters.”

Diagnostic commands

The MAX TNT command-line interface supports several network administration commands, which the *MAX TNT Reference Guide* describes in detail. This section provides a brief overview of the commands you’ll need for verifying that the MAX TNT is working as an IP router.

- Netstat (Display routing or interface tables)
- Nslookup (Perform DNS lookup)
- Ping (Bounce a packet off the specified host)
- Traceroute (Display route statistics)

Displaying the routing and interface tables

To view the routing table, enter the Netstat command with the –rn argument, as shown in the following example:

```
admin> netstat -rn
```

```
Destination       Gateway        IF       Flg   Pref Met     Use        Age
0.0.0.0/0         206.65.212.1   ie0      SG     100   1    4891      48630
10.0.0.0/24       11.168.6.249   ie1-12-1 RGT    100   3       0       9236
10.0.100.0/24     11.168.6.86    ie1-12-1 RGT    100   2       0      48601
10.0.200.0/24     11.168.6.86    ie1-12-1 RGT    100   2       0      48601
10.122.72.0/24    -              ie1-12-2 C        0   0    3141      48630
10.122.72.1/32    -              lo0      CP       0   0       0      48630
10.122.73.0/24    -              ie1-12-3 C        0   0    3140      48630
10.122.73.1/32    -              lo0      CP       0   0       0      48630
10.122.74.0/24    -              ie1-12-4 C        0   0    3139      48630
10.122.74.1/32    -              lo0      CP       0   0       0      48630
10.122.99.0/24    10.122.99.1    wan4     SG     100   7       0      48630
10.122.99.1/32    10.122.99.1    wan4     S      100   7       1      48630
127.0.0.1/32      -              lo0      CP       0   0       0      48672
127.0.0.2/32      -              rj0      CP       0   0       0      48672
127.0.0.3/32      -              bh0      CP       0   0       0      48672
11.0.2.0/24       11.168.6.249   ie1-12-1 RGT    100   2       0      48626
11.168.6.0/24     -              ie1-12-1 C        0   0   14589      48630
11.168.6.0/24     11.168.6.116   ie1-12-1 *RGTM  100   8       0      48606
11.168.6.0/24     11.168.6.142   ie1-12-1 *RGTM  100   8       0      48610
11.168.6.0/24     11.168.6.96    ie1-12-1 *RGTM  100   8       0      48624
11.168.6.102/32   11.168.6.86    ie1-12-1 RGT    100   8       0      48601
11.168.6.115/32   11.168.6.116   ie1-12-1 RGT    100   8       0      48606
11.168.6.116/32   11.168.6.96    ie1-12-1 RGT    100   8       0      48624
11.168.6.141/32   11.168.6.142   ie1-12-1 RGT    100   8       0      48610
11.168.6.145/32   11.168.6.86    ie1-12-1 RGT    100   8       0      48601
11.168.6.227/32   -              lo0      CP       0   0       0      48630
12.65.212.0/24    -              ie0      C        0   0   54432      48672
12.65.212.227/32  -              lo0      CP       0   0    4863      48672
255.255.255.255/32-              ie0      CP       0   0       0      48630
```

For details about the subnet notation Ascend uses (for example, 12.65.212.227/32, where 32 is the subnet mask), see “Using Ascend notation for IP addresses” on page 4-6.

The Destination and Gateway fields show the destination address and the address of the next-hop router used to reach it. Note that the router will use the most specific route (having the largest netmask) that matches a given destination. Direct routes do not show a gateway address.

The IF field shows the name of the interface through which a packet addressed to this destination will be sent. To view the interface table, enter the Netstat command with the –in argument, as shown in the following example:

```
admin> netstat -in
```

```
Name     MTU  Net/Dest          Address        Ipkts  Ierr   Opkts   Oerr
ie0      1500 12.65.212.0/24    12.65.212.227  107219    0   54351     0
lo0      1500 127.0.0.1/32      127.0.0.1        4867    0    4867     0
```

```
rj0      1500 127.0.0.2/32      127.0.0.2           0    0       0     0
bh0      1500 127.0.0.3/32      127.0.0.3           0    0       0     0
wan4     1500 10.122.99.1       -                   0    0       0     0
ie1-12-1 1500 11.168.6.0/24     11.168.6.227   430276  651       0     0
ie1-12-2 1500 10.122.72.0/24    10.122.72.1         0    0       0  3144
ie1-12-3 1500 10.122.73.0/24    10.122.73.1         0    0    3142     0
ie1-12-4 1500 10.122.74.0/24    10.122.74.1         0    0    3141     0
```

The entries named ie0 or ie*N-N-N*[-*N*] represent Ethernet interfaces.

N-N-N-N represents the shelf-number, slot-number, item-number, and logical-item-number of the interface.When the logical-item-number is zero, it does not appear in the interface name. This sequence of numbers is the same as the address used to index the IP-Interface profile. For example, the default profile for 1-12-1 is indexed as follows:

```
IP-INTERFACE { { 1 12 1 } 0 }
```

When the logical-item-number is zero, it appears in the interface name. Again, the sequence of numbers is identical to the profile index. For example, an IP-Interface profile with the following index:

```
IP-INTERFACE { { 1 12 1 } 3 }
```

is presented by the following interface name:

```
ie1-12-1-3
```

The other names in the interface table, and their significance, are:

- lo0 is the loopback interface.
- rj0 is the reject interface and bh0 is the blackhole interface, used in summarization.
- wan*N* is a WAN connection, entered as the connection becomes active.

Performing a DNS lookup

To retrieve the IP address of the host named techpubs, append the host's name to the Nslookup command:

```
admin> nslookup techpubs
Resolving host techpubs.
IP address for host techpubs is 10.6.212.19.
```

Pinging a host

To ping the host named techpubs, append the name to the Ping command:

```
admin> ping techpubs
PING techpubs (10.65.212.19): 56 data bytes
64 bytes from 10.65.212.19: icmp_seq=0 ttl=255 time=0 ms
64 bytes from 10.65.212.19: icmp_seq=3 ttl=255 time=0 ms
^C
--- techpubs ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0/0/0 ms
```

Displaying route statistics

To trace the route an IP packet takes to a host named cujo, append the host's name to the Traceroute command:

```
admin> traceroute cujo
traceroute to cujo (11.2.3.4), 30 hops max, 0 byte packets
1 cujo.ascend.com (11.2.3.4) 0 ms 0 ms 0 ms
```

Using Ascend notation for IP addresses

In the MAX TNT, IP addresses are specified in dotted decimal format (not hexadecimal). If no subnet mask is specified, the MAX TNT assumes a default mask based on address class. Table 4-3 shows address classes and the number of network bits in the default mask.

Table 4-3. IP address classes

Class	Address range	Network bits
Class A	0.0.0.0 — 127.255.255.255	8
Class B	128.0.0.0 — 191.255.255.255	16
Class C	192.0.0.0 — 223.255.255.255	24

For example, a class C address such as 198.5.248.40 has 24 network bits and an 8-bit host portion of the address. If no subnet mask is specified for a class C address, the MAX TNT assumes the default mask of 24 bits, as shown in Figure 4-2:

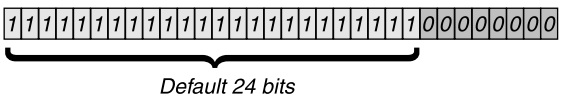


Figure 4-2. Class C IP address

To specify a subnet mask, the MAX TNT appends to the IP address a modifier that specifies the total number of network bits in the address. For example:

```
ip-address = 198.5.248.40/29
```

In this example, the /29 specification indicates that 29 bits of the address are used to specify the network. This is commonly referred to as a 29-bit subnet. The three remaining bits specify unique hosts.

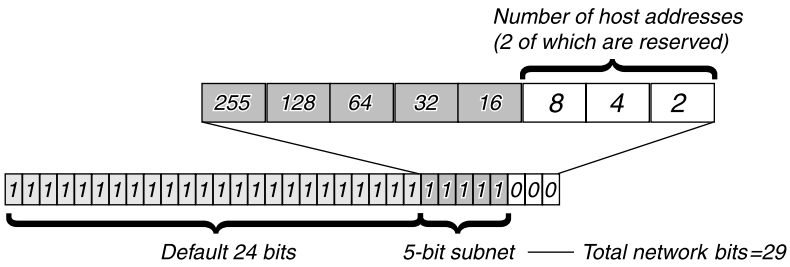


Figure 4-3. 29-bit subnet mask and the number of supported hosts

With three bits used to specify hosts on a 29-bit subnet, eight different bit-combinations are possible. Of those eight possible host addresses, two are reserved:

000 — Reserved for the network (base address)
001
010
100
110
101
011
111 — Reserved for the broadcast address of the subnet

Note: Early implementations of TCP/IP did not allow zero subnets. That is, subnets could have the same base address that a class A, B, or C network would have. For example, the subnet 192.168.8.0/30 was illegal because it had the same base address as the class C network 192.168.8.0/24, while 192.168.8.4/30 was legal. (192.168.8.0/30 is called a zero subnet, because like a class C base address, its last octet is zero.) Modern implementations of TCP/IP allow subnets to have base addresses that might be identical to the class A, B, or C base addresses. Ascend's implementations of RIP 2 and OSPF treat these so-called zero subnetworks the same as any other network. However, it is important that you treat zero subnets consistently throughout your network. Otherwise, you will encounter routing problems!

Table 4-4 shows standard and Ascend subnet formats for a class C network number.

Table 4-4. Standard subnet masks and Ascend notation

Subnet mask	Number of host addresses	Ascend notation
255.255.255.0	254 hosts + 1 broadcast, 1 network base	/24
255.255.255.128	126 hosts + 1 broadcast, 1 network base	/25
255.255.255.192	62 hosts + 1 broadcast, 1 network base	/26
255.255.255.224	30 hosts + 1 broadcast, 1 network base	/27
255.255.255.240	14 hosts + 1 broadcast, 1 network base	/28
255.255.255.248	6 hosts + 1 broadcast, 1 network base	/29
255.255.255.252	2 hosts + 1 broadcast, 1 network base	/30
255.255.255.254	invalid netmask (no hosts)	/31
255.255.255.255	1 host — a host route	/32

The broadcast address of any subnet has the host portion of the IP address set to all ones. The network address (or base address) represents the network itself, because the host portion of the IP address is all zeros. For example, if the MAX TNT configuration assigns this address to a remote router:

```
198.5.248.120/29
```

The Ethernet attached to that router has the following address range:

```
198.5.248.120 – 198.5.248.127
```

A host route is a special-case IP address with a subnet mask of /32. For example:

```
198.5.248.40/32
```

Host routes are required for a dial-in host.

Configuring LAN interfaces

This section describes how to configure the local Ethernet interfaces of the MAX TNT for IP routing. It covers the following topics:

- IP-Interface profile indexes
- Using system-based routing
- Using numbered interfaces
- Enabling proxy ARP on a LAN interface
- Enabling RIP on a LAN interface

For information about OSPF routing, see Chapter 5, “OSPF Router Configuration.” For information about multicast forwarding, see Chapter 6, “Multicast Forwarding.”

IP-Interface profile indexes

The MAX TNT creates a default IP-Interface profile for each local interface when it first detects the shelf-controller Ethernet port or the presence of an installed Ethernet card. For example, the output below shows the default IP-Interface profiles for the shelf-controller and a 100-Mbit Ethernet card installed in slot 12:

```
admin> dir ip-interface
    66  05/02/1997 10:13:24  { { shelf-1 controller 1 } 0 }
     8  05/10/1997 11:36:32  { { shelf-1 slot-12 2 } 0 }
     8  05/10/1997 11:36:32  { { shelf-1 slot-12 3 } 0 }
     8  05/10/1997 11:36:32  { { shelf-1 slot-12 4 } 0 }
     8  05/10/1997 11:36:59  { { shelf-1 slot-12 5 } 0 }
    64  05/10/1997 11:53:12  { { shelf-1 slot-12 1 } 0 }
```

The profile for the first Ethernet port on a card in shelf 1, slot 12, uses the following index:

```
{{1 12 1} 0}
```

This index is composed of a physical address and a logical-item number in the following format:

```
{{ shelf-N slot-N item-N } logical-item-N }
```

The logical item addresses a specific logical interface. It is zero except when multiple interfaces have been configured. The logical-item numbers do not have to be consecutive, but they must be unique. For example, the following command creates another IP-Interface profile for that Ethernet port:

```
admin> new ip-interface {{1 12 1} 1}
IP-INTERFACE/{ { shelf-1 slot-12 1 } 1 } read
```

Note: For IP-Interface profiles, the default profile (with the zero logical-item number) must have an IP address configured, or none of the other IP-Interface profiles for the same port will function. Therefore, do not delete the default profile and expect your other configurations to work!

Assigning local IP addresses

You must specify at least one IP address for each LAN interface that supports TCP/IP, unless the MAX TNT uses reverse ARP to obtain its address for the interface. (To enable reverse ARP, see “Enabling BootP and RARP” on page 4-27). To assign an IP address to a LAN interface, set the ip-address parameter in the IP-Interface profile, shown with its default value:

```
IP-INTERFACE {{shelf-N slot-N N } N }
   ip-address = 0.0.0.0/0
```

For background information about specifying subnet masks, see “Using Ascend notation for IP addresses” on page 4-6.

Using system-based routing

In system-based routing, each interface has a single IP address. If the MAX TNT has an installed four-port Ethernet card, for example, it has up to five local IP addresses. To assign an address, first obtain an IP address that is not in use on the network segment. Then, open the IP-Interface profile and specify the IP address. Following is an example that shows the commands entered to set the IP address to 10.1.2.65/24, and the system’s responses:

```
admin> dir ip-interface
    66  05/02/1997 10:13:24  { { shelf-1 controller 1 } 0 }
admin> read ip-interface { { 1 c 1 } 0}
IP-INTERFACE/{ { shelf-1 controller 1 } 0 } read
admin> set ip-address = 10.1.2.65/24
admin> write
IP-INTERFACE/{ { shelf-1 controller 1 } 0 } written
```

Using numbered interfaces

SNMP and some other types of applications might require that each side of a connection have a unique address associated only with that connection. This arrangement is called *numbered interfaces*. Figure 4-4 shows a local interface with two addresses, one of which is used for a numbered interface connection:

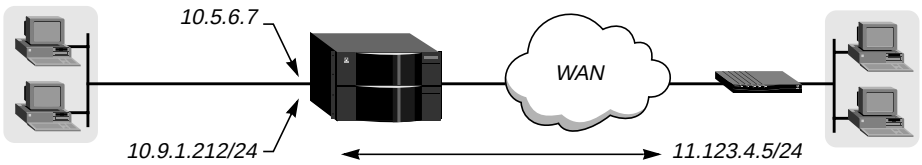


Figure 4-4. How numbered interfaces work

For numbered interfaces, a local interface supports multiple IP addresses. The address assigned in its default profile, and one or more additional addresses to be used for specific numbered-interface connections. (For related information, see “Example of a numbered interface WAN connection” on page 4-34.)

You can configure up to 16 IP-Interface profiles for each Ethernet card as a whole, with each profile specifying one IP address. The system creates the default profile for an interface and assigns it a 0 (zero) logical-item-number. To configure more than one IP address on a local

interface, create an IP-Interface profile for each unique IP address. The following example assigns the IP address 10.5.6.7 to the default IP interface for shelf 1, slot 12, port 1:

```
admin> dir ip-interface
    66  05/02/1997 10:13:24  { { shelf-1 controller 1 } 0 }
     8  05/10/1997 11:36:32  { { shelf-1 slot-12 2 } 0 }
     8  05/10/1997 11:36:32  { { shelf-1 slot-12 3 } 0 }
     8  05/10/1997 11:36:32  { { shelf-1 slot-12 4 } 0 }
     8  05/10/1997 11:36:59  { { shelf-1 slot-12 5 } 0 }
    64  05/10/1997 11:53:12  { { shelf-1 slot-12 1 } 0 }

admin> read ip-int { {1 12 1} 0}
IP-INTERFACE/{ { shelf-1 slot-12 1 } 0 } read

admin> set ip-address = 10.5.6.7

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

The following example creates a second IP-interface profile for the same physical port and assign it the address 10.9.1.212/24:

```
admin> new ip-int { {1 12 1 } 1}
IP-INTERFACE/{ { shelf-1 slot-12 1 } 1 } read

admin> set ip-addr = 10.9.1.212/24

admin> write
IP-INTERFACE/{ { shelf-1 slot-12 1 } 1 } written
```

Note: For IP-Interface profiles, the default profile (with the zero logical-item number) must have an IP address configured, or none of the other IP-Interface profiles for the same port will function. Do not delete the default IP-Interface profiles.

Note the following differences in the way the MAX TNT operates when it is using a numbered interface instead of system-based routing (where each Ethernet interface has a single IP address):

- IP packets generated in the MAX TNT and sent to the remote address will have an IP source address corresponding to the numbered interface.
- During authentication of a call placed from a MAX TNT using a numbered interface, the MAX TNT reports the address of the interface as its IP address.
- The MAX TNT adds all numbered interfaces listed in Connection profiles to its routing table as host routes.

The MAX TNT accepts IP packets destined for a numbered interface and treats them as destined for the MAX TNT itself. (The packet may arrive on any interface, and the destination numbered interface need not be in the active state.)

Enabling proxy ARP on a LAN interface

When you enable proxy ARP, hosts on the LAN interface can ARP for hosts or subnets that reside across the WAN but have an IP address on the local network. The MAX TNT responds to the local hosts' ARP requests, and then routes the packets for those connections across the WAN. To enable proxy ARP, set the proxy-mode parameter in the IP-Interface profile, shown with its default value:

```
IP-INTERFACE {{shelf-N slot-N N } N }
   proxy-mode = Off
```

You can set proxy-mode to Active (respond for active WAN connections only), Inactive (respond only for inactive WAN connections), or Always (respond for all pool addresses). If the MAX TNT is set to respond to ARP requests for inactive connections (Inactive or Always), it brings up the required WAN connection.

Note: If proxy-mode is enabled in any of the IP-Interface profiles for a given Ethernet port, it is enabled for all ARP requests coming into the physical port.

To enable the MAX TNT to respond as proxy for ARP requests, but only for active WAN connections, set proxy-mode to Active, as shown in the following example:

```
admin> read ip-interface {{1 12 3} 0}
IP-INTERFACE/{ { shelf-1 slot-12 3 } 0 } read

admin> set proxy-mode = Active

admin> write
IP-INTERFACE/{ { shelf-1 slot-12 3 } 0 } written
```

Enabling RIP on a LAN interface

By default, RIP is turned off on local interfaces. You can enable it to send out updates on the interface, receive updates from other routers on that interface, or both. To configure RIP, use the following parameters (shown with their default values):

```
IP-INTERFACE {{shelf-N slot-N N } N }
   rip-mode = routing-off
   rip2-use-multicast = yes
```

For the details of setting each parameter, see the *MAX TNT Reference Guide*.

Figure 4-5 shows the MAX TNT and a local router, connecting to a remote access router with another router on the remote network. Each router maintains its own routing table.

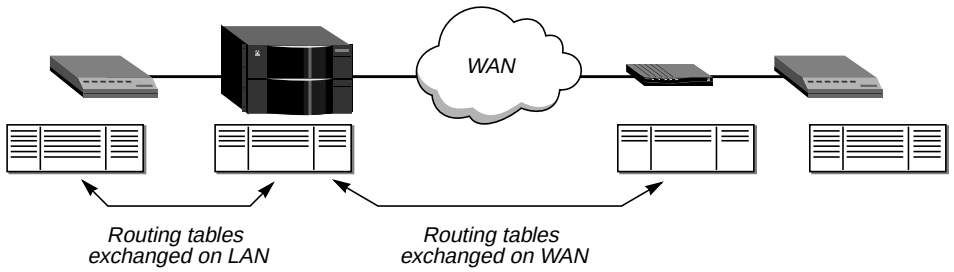


Figure 4-5. Deciding whether to enable RIP

When RIP is turned off in the IP-Interface profile, the MAX TNT does not propagate its routing table to routers on the LAN interface, so local hosts do not have access to the remote network and its routes. When RIP is off, the MAX TNT also does not receive the local routers' updates to its routing table. So, callers do not have access to other routes maintained locally. If you decide to enable RIP on the LAN interface, you have the following options:

- Configure the MAX TNT to receive RIP updates. This increases the size of the MAX TNT routing table, so it can access more networks.

- Configure the MAX TNT to send RIP updates. This propagates information about remote networks to local routers.
- Send and receive RIP updates on the interface. This updates the routing table in both the MAX TNT and other local routers on the interface.
- Use RIP-v2 (recommended) or RIP-v1.

  RIP version 2 is preferable to version 1. Ascend does not recommend running RIP-v2 and RIP-v1 on the same network in such a way that the routers receive each other's advertisements. RIP-v1 guesses subnet masks, while RIP-v2 handles them explicitly. Running the two versions on the same network can result in RIP-v1 guesses overriding accurate subnet information obtained via RIP-v2.
- Use the multicast address (224.0.0.9) rather than the broadcast address for RIP updates.

  By default, RIP updates use the multicast address. If you must use the broadcast address for backward compatibility with other systems that are reached through this interface, you can turn off the use of the multicast address.

Following is an example that enables the MAX TNT to send and receive RIP-v2 updates on the multicast address:

```
admin> read ip-interface {{1 12 3} 0}
IP-INTERFACE/{ { shelf-1 slot-12 3 } 0 } read

admin> set rip-mode = routing-send-and-recv-v2

admin> write
IP-INTERFACE/{ { shelf-1 slot-12 3 } 0 } written
```

For information about filtering routes or configuring route metrics in RIP update packets, see Chapter 7, "Packet and Route Filters."

Configuring the IP router

This section describes how to configure the MAX TNT IP router. It covers the following topics:

- Accessing the IP-Global profile
- Specifying a system address
- Setting an interface-independent IP address
- Providing access to DNS
- Configuring address pools for dynamic assignment to dial-in hosts
- Sharing profiles
- Configuring Telnet access to the system
- Configuring system-level routing policies and preferences
- Enabling BootP and RARP
- Enabling UDP checksums
- Using SNTP to set and maintain the MAX TNT system time

For information about OSPF routing, see Chapter 5, "OSPF Router Configuration." For information about multicast forwarding, see Chapter 6, "Multicast Forwarding."

Accessing the IP-Global profile

System-level configuration of the IP router consists largely of configuring the IP-Global profile. To read the profile into the edit buffer, use the Read command as follows:

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

The following sections describe parameters in the IP-Global profile. For detailed information about each parameter, see the *MAX TNT Reference Guide*.

Specifying a system address

By default, the MAX TNT uses the IP address assigned to the shelf-controller Ethernet interface as the source address for packets it generates, such as RADIUS or TACACS+ requests, or a Telnet, Traceroute, or Ping command originating from the unit. (See the description of the system-ip-addr parameter in the *MAX TNT Reference Guide* for more information about generated packets.)

You can use the following parameter to specify a different source address for these packets (shown with a sample setting):

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

If you specify an IP address in the system-ip-addr parameter, the MAX TNT uses that as the source address for packets it generates. For return packets to reach this address, the remote host must have a route to the address (or an ARP entry).

The most common reason for setting a system address other than the shelf-controller address is that doing so simplifies access control. For example, most RADIUS servers keep a database of known RAS clients and their authentication keys. If you don’t specify a system address, you must include a complete list of all the system's interface addresses in this database. If you specify a system address, it is used for all RADIUS request packets.

Another reason to set a system address is to force the MAX TNT to use an IP address assigned to an Ethernet interface on a slot card rather than the shelf-controller address as the source address for packets it originates.

In addition, some tunneling protocols, such as Ascend Tunnel Management Protocol (ATMP) require that you specify a system address.

Following is an example that sets the system-ip-addr parameter to an address assigned to a port on a slot card:

```
admin> dir ip-interface
    66  05/02/1997 10:13:24  { { shelf-1 controller 1 } 0 }
     8  05/10/1997 11:36:32  { { shelf-1 slot-12 2 } 0 }
     8  05/10/1997 11:36:32  { { shelf-1 slot-12 3 } 0 }
     8  05/10/1997 11:36:32  { { shelf-1 slot-12 4 } 0 }
     8  05/10/1997 11:36:59  { { shelf-1 slot-12 5 } 0 }
    64  05/10/1997 11:53:12  { { shelf-1 slot-12 1 } 0 }

admin> get ip-int { {1 12 1} 0} ip-address
ip-address = 10.2.3.4
```

```
admin> read ip-global
IP-GLOBAL read
admin> set system-ip-addr = 10.2.3.4
admin> write
IP-GLOBAL written
```

If the system address becomes unreachable due to a change in the network topology, the MAX TNT might still be reachable by Telnet at any of its other interface addresses. (Of course, this is subject to packet filtering throughout the network.)

Setting an interface-independent IP address

To make inbound IP traffic independent of physical addresses in the MAX TNT, you can define an interface-independent (soft) IP address. Because it is not associated with a physical port, this address is always accessible if any one of the MAX TNT physical interfaces is up.

You can use the following parameter to set a soft interface address (shown with a sample setting):

```
IP-GLOBAL
   soft-ip-interface-addr = 11.168.7.100
```

Note: The interface-independent address must have a unique IP address, just like other interfaces in the unit.

Following is an example that sets the soft interface address to 11.168.7.100:

```
admin> read ip-global
IP-GLOBAL read
admin> set soft-ip-interface-addr = 11.168.7.100
admin> write
IP-GLOBAL written
```

This address is advertised in RIP and OSPF as a host route with a mask of /32 using the loopback interface. To enable hosts on the network to reach this address, you must either enable routing protocols (RIP or OSPF) or configure static routes in routers one hop away from the MAX TNT. To verify that other hosts in your network have a route to the soft address, use Ping or Traceroute from the other hosts. For example:

```
host1% ping 11.168.7.100
PING 11.168.7.100 (11.168.7.100): 56 data bytes
64 bytes from 11.168.7.100: icmp_seq=0 ttl=255 time=0 ms
64 bytes from 11.168.7.100: icmp_seq=7 ttl=255 time=0 ms
^C
--- 11.168.7.100 ping statistics ---
8 packets transmitted, 8 packets received, 0% packet loss
round-trip min/avg/max = 0/0/0 ms
```

Providing access to DNS

There are three aspects to DNS configuration in the MAX TNT:

- Specifying domain names to use for name lookups

- Specifying which name servers are accessible to clients (DNS or NetBIOS)
- Supporting DNS list for DNS servers that use this feature

For information about configuring client DNS, which enables the MAX TNT to direct connections to DNS servers owned by a client, or to DNS servers owned by several different client on a per-connection basis, see Appendix B, "Authorization Options."

Specifying domain names for name lookups

When the MAX TNT is given a host name to look up, it tries various combinations. For example, it appends the domain name specified in the IP-Global profile. You can specify a primary and secondary domain name for DNS lookups by using the following parameters, shown with sample settings:

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

The secondary domain name specifies another domain name the MAX TNT can search if the host name is not found in the primary domain. The following example shows the commands entered to set the primary and secondary domain names, and the system's responses:

```
admin> read ip-global
IP-GLOBAL read
admin> set domain-name = abc.com
admin> set sec-domain-name = eng.abc.com
admin> write
IP-GLOBAL written
```

Specifying which name servers are accessible

In the IP-Global profile, administrators can specify addresses for two local DNS servers, and additional DNS servers referred to as *client* servers. Client servers are accessed only if a caller's configured profile does not include DNS server addresses specific to that connection. Following are the related parameters in the IP-Global profile, shown with sample settings:

```
IP-GLOBAL
   dns-primary-server = 10.65.212.178
   dns-secondary-server = 10.65.212.10
   client-primary-dns-server = 0.0.0.0
   client-secondary-dns-server = 0.0.0.0
   allow-as-client-dns-info = True
```

For information about configuring client DNS, see Appendix B, "Authorization Options."

If you are using NetBIOS

If the local network supports NetBIOS instead of DNS, you can configure the MAX TNT to access NetBIOS servers by setting the following parameters (shown with sample values):

```
IP-GLOBAL
   netbios-primary-ns = 10.1.2.3/24
   netbios-secondary-ns = 10.2.3.4/24
```

Following is an example that specifies NetBIOS server addresses:

```
admin> read ip-global
IP-GLOBAL read
admin> set netbios-primary-ns = 10.1.2.3/24
admin> set netbios-secondary-ns = 10.2.3.4./24
admin> write
IP-GLOBAL written
```

The system accesses the secondary NetBIOS server only if the primary server is not found.

If you are using DNS

If you inform the MAX TNT about name servers on the local network, callers can access the databases on those hosts. The following example specifies DNS server addresses:

```
admin> read ip-global
IP-GLOBAL read
admin> set dns-pri = 10.2.3.56
admin> set dns-sec = 10.2.3.107
admin> write
IP-GLOBAL written
```

The secondary server is accessed only if the primary one is inaccessible.

Supporting DNS list

Some DNS servers support a list feature that enables them to return multiple addresses for a host name in response to a DNS query. However, these responses do not include information about availability of the hosts in the list. Users typically attempt to access the first address in the list. If that host is unavailable, the user must try the next host, and so forth.

When the DNS list is used for an immediate connection by a dial-in user (for example, an immediate Telnet connection to a local host), and the first attempt fails, the physical connection is torn down.To avoid tearing down physical links when the host is unavailable, you can support DNS list in the MAX TNT by setting the following parameters, shown with default settings:

```
IP-GLOBAL
   dns-list-attempt = no
   dns-list-size = 6
   tcp-timeout = 0
```

The dns-list-attempt parameter enables the user to try one entry in the DNS list of hosts, and, if that connection fails, to try the next entry, without losing the WAN session, and so on. The dns-list-size parameter specifies the maximum number of hosts listed.

The DNS list attempt has a default timeout of 170 seconds, which means that the MAX TNT attempts to connect to the first host on the list for that length of time. When the tcp-timeout parameter is set to zero, the default 170-second timeout applies. Some client software times out in less than 170 seconds, which causes it to drop the connection before attempting the second host if the first host does not respond. In that case, you can set the tcp-timeout parameter to a smaller timeout period, such as 30 or 60 seconds.

Following is an example that enables DNS list with a maximum of 5 hosts in the list:

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

admin> set dns-list-attempt = yes

admin> set dns-list-size = 15

admin> write
IP-GLOBAL written
```

Configuring address pools for dynamic assignment to dial-in hosts

For dial-in PPP clients that are not running as IP routers, the MAX TNT can assign the connection a local IP address on a first-come first-served basis. After the connection is terminated, the address that was assigned to that connection is returned to the pool for reassignment to another connection.

To enable the MAX TNT to handle local IP addresses in this way, you configure the following parameters in the IP-Global profile, shown with their default values:

```
IP-GLOBAL
   must-accept-address-assign = no
   pool-summary = no
   pool-ospf-adv-type = type-1
   pool-base-address = [ 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 0.+
   assign-count = [ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0+
   pool-name = [ "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""+
```

Requiring acceptance of the pool address

During PPP negotiation, a caller could reject the IP address offered by the MAX TNT and present its own IP address for consideration. For security reasons, you might want to set the must-accept-address-assign parameter to ensure that the MAX TNT terminates such a call:

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

admin> set must-accept-address-assign = yes

admin> write
IP-GLOBAL written
```

The Answer-Defaults profile must enable dynamic assignment, the caller's configured profile must specify dynamic assignment, and the caller's PPP dial-in software must be configured to acquire its IP address dynamically. For more details, see "Example of a dial-in host requiring address assignment" on page 4-32.

Importing pools into OSPF as internal routes

The MAX TNT can import summarized pool addresses into OSPF as internal routes, which enables them to work properly with stub areas. When pool-summary is set to Yes and OSPF is enabled, OSPF looks at the pool-ospf-adv-type parameter to decide how to import pool addresses into OSPF:

- Type-1 (the default), to import the pool addresses into OSPF as external Type-1 routes.
- Type-2, to import the pool addresses into OSPF as external Type-2 routes.
- Internal, to import the pool addresses into OSPF as intra-area routes.

Note: If you change the value of this parameter, you must reset the MAX TNT for the change to take effect.

Pool names

Each pool configuration consists of a base address, address count, and name. A pool name can contain up to 11 characters. If TACACS+ PPP authentication is performed, the appropriate string is returned and matched against the pool names to obtain the pool number to be used in allocating an address for the connection. If all addresses in a named pool have been allocated, an address is taken from the next pool with the same name.

What is pool summary?

The pool-summary feature is designed to reduce the routing overhead associated with address pools. Originally, each address assigned from a pool was advertised as a host route with a subnet mask of 32. As the number of supported pool addresses grew (the MAX TNT can support up to 32,512 pool addresses), the possible overhead associated with advertising these host routes became an issue.

The pool-summary feature sets a flag that enables the MAX TNT to advertise a route to the entire pool of addresses rather than advertise each address within the pool. However, to use this feature, the pools must be *network aligned*, which requires a special pool base address and adherence to other specifications. For details, see “Setting up summarized address pools (pool summary)” on page 4-19.

If you do not use the pool-summary feature, each address in a pool is advertised as a host route with a subnet mask of 32. In that case, the pool does not have to be network-aligned, so any IP address that begins a block of free addresses can serve as the pool base address.

Setting up address pools (no pool summary)

You can define up to 128 address pools, with each pool containing up to 254 contiguous IP addresses. A *non-aligned* pool can start at any pool base address. Addresses do not accept a subnet mask component, because they are always advertised as host routes.

- The pool-base-address parameter specifies the first address in a block of contiguous addresses on the local network or subnet.
- The assign-count parameter specifies how many addresses are in the pool (up to 254).
- The pool-name parameter is optional unless you are using TACACS+ authentication.

  If you specify a pool-name parameter when TACACS+ authentication is not in use, the name is treated as a comment.

The following example specifies three pools, each containing 50 contiguous free IP addresses:

```
admin> read ip-global
IP-GLOBAL read
admin> set pool-base-address 1 = 10.2.3.4
admin> set pool-base-address 2 = 11.5.7.51
```

```
admin> set pool-base-address 3 = 12.7.112.15
admin> set assign-count 1 = 50
admin> set assign-count 2 = 50
admin> set assign-count 3 = 50
admin> write
IP-GLOBAL written
```

With these commands, the following addresses are allocated for dynamic assignment to callers:

- Pool #1: 10.2.3.4 through 10.2.3.54
- Pool #2: 11.5.7.51 through 11.5.7.101
- Pool #1: 12.7.112.15 through 12.7.112.65

Setting up summarized address pools (pool summary)

This section shows how to set up address pools that comply with all pool-summary requirements, which enables the MAX TNT to summarize the host routes into one advertisement for the pool as a whole. For this feature to work, your configuration must meet the following requirements:

- Each pool to be summarized must be network aligned (explained below).
- The MAX TNT must have a static route to the pool subnet.
- Connection profiles that use one of the pool addresses must be private routes.

Network-aligned address pools

Following are the rules for network-aligned address pools:

1. The assign-count value must be two less than the total number of addresses in the pool. (Add two to assign-count for the total number of addresses in the subnet, and calculate the mask for the subnet on the basis of this total.)
2. The pool-base-address value must be the first host address. (Subtract 1 from the pool-base-address for the base address for the subnet.)

For example, the following configuration is network aligned:

```
admin> read ip-global
IP-GLOBAL read
admin> set pool-base-address 1 = 10.12.253.1
admin> set assign-count 1 = 62
admin> write
IP-GLOBAL written
```

In this example, the pool-base-address is set to 10.12.253.1. When you subtract one from this address, you get 10.12.253.0, which is a valid base address for the 255.255.255.192 subnet mask. Note that 10.12.253.64, 10.12.253.128, and 10.12.253.192 are also valid zero addresses for the same mask. The resulting address pool subnet is 10.12.253.0/26.

The assign-count is set to 62. When you add two to the assign-count, you get 64. The subnet mask for 64 addresses is 255.255.255.192 (256–64 = 192). The Ascend subnet notation for a 255.255.255.192 mask is /26.

Configuring a static route to the summarized subnet

After verifying that *every one* of the configured address pools is network aligned, you must enter a static route for each of them. These static routes handle all IP address that have not been given to users, by routing them to the reject interface or the blackhole interface.

The reject (rj0) interface address is 127.0.0.2. Packets routed to this interface are bounced back to the sender with an ICMP unreachable message.

The blackhole (bh0) interface address is 127.0.0.3. Packets routed to this interface are silently discarded.

The MAX TNT creates a host route for each assigned address in the pool, and host routes override subnet routes. So packets whose destination matches an assigned IP address from the pool are properly routed and not discarded or bounced. But because the MAX TNT advertises the entire pool as a route, and only privately knows which IP addresses in the pool are active, a remote network might improperly send the MAX TNT a packet for an inactive IP address. Depending on the static-route gateway address, these packets are either bounced with an ICMP unreachable message (reject interface) or silently discarded (blackhole interface).

For example, you could use a procedure similar to the following to set up the destination and gateway parameters that define the pool. This example uses the blackhole interface for inactive host addresses, and sets the other required values for the route. For information about configuring static routes, see “Working with static IP routes” on page 4-38.

1 Create a new IP-Route profile:

```
admin> new ip-route pool-net
IP-ROUTE/pool-net read
```

2 Specify the base network address of the network-aligned pool:

```
admin> set dest-address = 10.12.253.0/26
```

3 Specify the blackhole interface (for example) as the gateway-address:

```
admin> set gateway-address = 127.0.0.3
```

4 Set the metric, cost, and preference values to zero:

```
admin> set metric = 0
admin> set cost = 0
admin> set preference = 0
```

5 Make sure the route is not private:

```
admin> set private-route = no
```

6 Write the IP-Route profile:

```
admin> write
IP-ROUTE/pool-net written
```

The routing table will contain the following lines:

```
Destination         Gateway    IF      Flg       Pref   Met     Use    Age
10.12.253.0/26        -          bh0     C          0      0       0   172162
127.0.0.1/32          -          lo0     CP         0      0       0   172163
127.0.0.2/32          -          rj0     CP         0      0       0   172163
127.0.0.3/32          -          bh0     CP         0      0       0   172163
```

Making each Connection profile that uses the pool private

When you configure Connection profiles that will be assigned an IP address from the summarized pool, make sure the private-route parameter is set to Yes. For example:

1 Open a Connection profile that will use dynamic address assignment:

```
admin> read conn vikki
CONNECTION/vikki read

admin> list ip-options
ip-routing-enabled = yes
vj-header-prediction = yes
remote-address = 0.0.0.0/0
local-address = 0.0.0.0/0
routing-metric = 1
preference = 60
down-preference = 120
private-route = no
multicast-allowed = no
address-pool = 0
ip-direct = 0.0.0.0
rip = routing-off
ospf-options = { no 0.0.0.0 normal 30 120 5 simple ******* 10 1000+
multicast-rate-limit = 100
client-dns-primary-addr = 0.0.0.0
client-dns-secondary-addr = 0.0.0.0
client-dns-addr-assign = yes
client-default-gateway = 0.0.0.0/0
```

2 Specify the number of the summarized pool:

```
admin> set address-pool = 1
```

3 Make the WAN connection route private:

```
admin> set private-route = yes
```

4 Write the profile:

```
admin> write
CONNECTION/vikki written
```

This example assumes that address-pool 1 is network aligned and has a configured route, as shown in the preceding sections. In addition, the Answer-Defaults profile must enable dynamic assignment, and the caller's PPP dial-in software must be configured to acquire its IP address dynamically. For more information about configuring Connection profiles, see "Configuring WAN interfaces" on page 4-29.

Sharing profiles

The following parameter specifies whether the MAX TNT will allow more than one incoming call to share the same Connection profile:

```
IP-GLOBAL
   shared-prof = no
```

For routed IP callers, shared profiles cannot result in two IP addresses reached through the same profile.

In low-security situations, more than one caller can share a name and password for accessing the local network. This would require sharing a single Connection profile that does not assign an IP address, or one that specifies dynamic IP address assignment. If a shared profile uses an IP address, it must be assigned dynamically, because multiple hosts cannot share a single IP address. When the shared profile uses dynamic address assignment, each call is a separate connection that shares the same name and password. A separate IP address is assigned dynamically to each caller.

Following is an example that enables shared profiles:

```
admin> read ip-global
IP-GLOBAL read
admin> set shared-prof = yes
admin> write
IP-GLOBAL written
```

Configuring Telnet access to the system

The following parameters configure Telnet access to the MAX TNT (shown with their default values):

```
IP-GLOBAL
   telnet-password = ""
   user-profile = ""
```

All users attempting to access the MAX TNT unit via Telnet are prompted for the Telnet password. They are allowed three tries, each with a 60-second time limit, to enter the correct password. If all three tries fail, the connection attempt times out.

You can also associate a User profile with Telnet sessions. By default, no profile is specified, which means that each Telnet user must supply the name and password of a User profile. If you specify a User profile for Telnet sessions, the system uses that profile for any Telnet login. If the profile has a password, it prompts the Telnet user for an additional password following the Telnet password. If not, supplying the Telnet password alone will allow access to the unit.

Following is an example that sets the Telnet password and specifies the Default User profile for Telnet logins. The Default profile enables minimal permissions and requires no password. This example shows the commands entered and the system's responses:

```
admin> read ip-global
IP-GLOBAL read
admin> set telnet-password = Ascend
admin> set user-profile = default
admin> write
IP-GLOBAL written
```

Configuring system-level routing policies and preferences

The following parameters configure routing policies and route preferences in the MAX TNT:

```
IP-GLOBAL
   rip-policy = Poison-Rvrs
   summarize-rip-routes = no
```

```
ignore-icmp-redirects = no
drop-source-routed-ip-packets = no
ignore-def-route = yes
dialout-poison = no
static-pref = 100
rip-pref = 100
rip-queue-depth = 0
ospf-pref = 10
ospf-ase-pref = 150
rip-tag = c8:00:00:00
rip-ase-type = 1
iproute-cache-enable = yes
iproute-cache-size = 0
```

For complete information about each parameter, see the *MAX TNT Reference Guide*.

RIP-v1 issues

The rip-policy and summarize-rip-routes parameters have no effect on RIP-v2. The IETF has voted to move RIP-v1 into the Historic category, and its use is no longer recommended. You should upgrade all routers and hosts to RIP-v2. If you must maintain RIP-v1, Ascend recommends that you create a separate subnet for all RIP-v1 routers and hosts.

If the MAX TNT is running RIP-v1, the rip-policy parameter must specify a split-horizon or poison-reverse policy for outgoing update packets that include routes that were received on the same interface on which the update is sent. Split-horizon means that the MAX TNT does not propagate routes back to the subnet from which they were received. Poison-reverse means that it propagates routes back to the subnet from which they were received, but with a metric of 16.

The summarize-rip-routes parameter specifies whether to summarize subnet information when advertising routes. If the MAX TNT summarizes RIP routes, it advertises a route to all the subnets in a network of the same class. For example, the route to 200.5.8.13/28 (a class C address) would be advertised as a route to 200.5.8.0. When the MAX TNT does not summarize information, it advertises each route in its routing table as-is.

Ignoring ICMP redirects

ICMP was designed to find the most efficient IP route to a destination. ICMP redirect packets are one of the oldest route-discovery methods on the Internet. They are also one of the least secure, because it is possible to counterfeit ICMP redirects and change the way a device routes packets. Following is an example that protects the router from ICMP redirects:

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

admin> set ignore-icmp-redirects = yes

admin> write
IP-GLOBAL written
```

Dropping source-routed packets

The drop-source-routed-ip-packets parameter specifies whether the MAX TNT will forward IP packets with the source route option set. The default is No, which causes the MAX TNT to

forward all source routed packets as described in RFC1812, *Requirements For Routers*. When the parameter is set to Yes, the MAX TNT drops all packets that have either a Loose or a Strict source route among their IP options. Following is an example that instructs the router to drop source-routed packets:

```
admin> read ip-global
IP-GLOBAL read
admin> set drop-source-routed-ip-packets = yes
admin> write
IP-GLOBAL written
```

Ignoring the default route

You can configure the MAX TNT to ignore default routes advertised by routing protocols. This configuration is recommended. The default route specifies a static route to another IP router, which is often a LAN router. When the MAX TNT is configured to ignore the default route, RIP updates do not modify the default route in the MAX TNT routing table. Following is an example that protects the default route from dynamic modifications:

```
admin> read ip-global
IP-GLOBAL read
admin> set ignore-def-route = yes
admin> write
IP-GLOBAL written
```

Poisoning routes to force the use of a redundant Ascend unit

If you have another Ascend unit backing up the MAX TNT in a redundant configuration on the same network, you can use the poison-dialout parameter to let the redundant unit take over when necessary. If you set the parameter to Yes, it instructs the MAX TNT to stop advertising IP routes that use dial services if for any reason its trunks are in the alarm condition. Otherwise, it continues to advertise its dialout routes, which prevents the redundant unit from taking over the routing load. Following is an example that makes use of this feature for redundant units:

```
admin> read ip-global
IP-GLOBAL read
admin> set dialout-poison = yes
admin> write
IP-GLOBAL written
```

Static and RIP preferences

RIP is a distance-vector protocol, which uses a hop count to select the shortest route to a destination network. OSPF is a link-state protocol, which means that OSPF can take into account a variety of link conditions, such as the reliability or speed of the link, when determining the best path to a destination network. Because these two metrics are incompatible, the MAX TNT supports route preferences.

By default, static routes and RIP routes have the same preference, so they compete equally. ICMP redirects take precedence over both and OSPF takes precedence over everything. If a dynamic route's preference is lower than that of the static route, the dynamic route can hide

(temporarily overwrite) a static route to the same network. However, dynamic routes age, and if no updates are received, they eventually expire. In that case, the hidden static route reappears in the routing table.

In the following example, the administrator increases the preference value of RIP routes, instructing the router to use static routes first if they exist:

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

admin> set rip-pref = 150

admin> write
IP-GLOBAL written
```

For information about filtering routes or configuring route metrics in RIP update packets, see Chapter 7, "Packet and Route Filters."

Specifying UDP packet queues

When the router is very busy and receives a flood of UDP packets from SNMP requests or RIP updates, a backlog of packets waiting for processing can create enough delay in routing to cause sporadic problems with time-sensitive packets, such as LCP negotiation or Frame Relay management packets.

To prevent these possible problems, the UDP processing logic runs at a lower priority than routed packets. On a system busily routing packets, this could mean that UDP processing is delayed, and that a backlog of UDP packets may build up. Following are the parameters that specify the maximum size of this backlog, shown with sample settings:

```
IP-GLOBAL
   rip-queue-depth = 50

SNMP
   queue-depth = 0
```

The specified number of packets will be held for processing, and if additional packets are received when the queue is full, those packets are dropped.

Note: A queue depth of zero means the MAX TNT does not drop UDP packets, no matter how far behind it gets in processing them. When you configure a queue depth, the MAX TNT is more likely to drop UDP packets when it is busy routing packets. However, time-sensitive routed packets are less likely to be delayed and system memory is used more efficiently.

Specifying a RIP queue depth

The default queue depth for RIP packets is 50. Valid values for the RIP queue depth are 0–1024. A value of zero means the packets will not be dropped, no matter what the state of the routing subsystem or system memory. Following is an example that sets the RIP queue depth to 128:

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

admin> set rip-queue-depth = 128

admin> write
IP-GLOBAL written
```

Specifying a queue for SNMP requests

The default queue depth for SNMP requests is zero, which means the packets will not be dropped, no matter no matter what the state of the SNMP subsystem or system memory. If the queue grows too large in an extremely loaded routing environment, the system could ultimately run out of memory. Valid values for the queue depth are 0–1024. Following is an example that shows the commands entered to set the SNMP queue to 32, and the system's responses:

```
admin> read snmp
SNMP read

admin> set queue-depth = 32

admin> write
SNMP written
```

Displaying information about UDP queues

To view information about UDP sockets, append UDP to the Netstat command. The command output shows the queue depth of various UDP ports, as well as the total packets received and total packets dropped on each port. The total packets received count includes the total packets dropped. For this sample output, the SNMP queue depth was set to 32:

```
admin> netstat udp
udp:
Socket  Local Port  InQLen  InQMax    InQDrops    Total Rx
  0        1023       0       1           0           0
  1       route       0      50           0         509
  2        echo       0      32           0           0
  3         ntp       0      32           0           0
  4        1022       0     128           0           0
  5        snmp      32      32        5837       20849
```

OSPF ASE preferences and handling

Because OSPF typically involves a complex environment, its router configuration is described in Chapter 5, "OSPF Router Configuration."For information about the following parameters:

```
IP-GLOBAL
   ospf-ase-pref = 150
   rip-tag = c8:00:00:00
   rip-ase-type = 1
```

see Chapter 5, "OSPF Router Configuration."

Route caches

The global routing table is maintained on the shelf-controller and is used to route packets internally to the correct interface. To offload some of the routing overhead and improve performance, the MAX TNT uses route caches on each slot card. Route caches work as follows:

- When a modem or HDLC card receives an IP packet, it forwards the packet to the shelf-controller, which routes it to the proper slot (an Ethernet card, for example).
- When the shelf-controller routes the packet, it writes a cache entry that is downloaded to the route cache of all slot cards.

- When the modem or HDLC card receives another IP packet with the same destination address, it checks its route cache and forwards the packet directly to the proper slot, without involving the shelf-controller.

The shelf-controller retains responsibility for managing routing protocols, the global routing table, and the route caches themselves. But each slot card is able to check a small IP cache and route packets to a destination interface without involving the shelf-controller. When a slot card receives an IP packet for which it has no cache entry, it forwards that packet to the shelf-controller, which routes the packet and writes a cache entry to all slot cards.

If you must control memory usage for the card, you can restrict the cache size or disable the route cache with the following parameters (shown with their default values, which are recommended):

```
IP-GLOBAL
   iproute-cache-enable = yes
   iproute-cache-size = 0
```

Route caches are enabled by default, and Ascend recommends that you do not disable route caches or change their size.The iproute-cache-size parameter is set to 0 by default, which sets no limit on the size of the cache. If you set a higher number, it represents the number of cache entries. Usually, no limit is required.

Enabling BootP and RARP

The following parameters configure Bootstrap Protocol (BootP) and reverse-ARP (RARP) in the MAX TNT (shown with their default values):

```
IP-GLOBAL
   bootp-enabled = no
   rarp-enabled = no
```

With bootp-enabled set to Yes, the MAX TNT can query a BootP server for new parameters and to check for a new software load. With rarp-enabled set to Yes, the MAX TNT can obtain its own IP addresses from a RARP server. Following is an example that enables both BootP and RARP:

```
admin> read ip-global
IP-GLOBAL read
admin> set bootp-enabled = yes
admin> set rarp-enabled = yes
admin> write
IP-GLOBAL written
```

Enabling UDP checksums

The udp-checksum parameter enables UDP checksums for transmitted packets:

```
IP-GLOBAL
   udp-cksum = yes
```

If data integrity is of the highest concern for your network and redundant checks are important, you can turn on UDP checksums to generate a checksum whenever a UDP packet is

transmitted. UDP packets are transmitted for queries and responses related to ATMP, SYSLOG, DNS, ECHOSERV, RADIUS, TACACS, RIP, SNTP, and TFTP. Following is an example that turns on UDP checksums:

```
admin> read ip-global
IP-GLOBAL read
admin> set udp-cksum = yes
admin> write
IP-GLOBAL written
```

Using SNTP to set and maintain the MAX TNT system time

The MAX TNT can use Simple Network Time Protocol (SNTP—described in RFC 1305) to set and maintain its system time by communicating with an SNTP server. You configure the process in the sntp-info subprofile of the IP-Global profile, by setting the following parameters:

```
IP-GLOBAL
   sntp-info
      enabled = no
      GMT-offset = utc+0000
      host
         host[1] = 0.0.0.0
         host[2] = 0.0.0.0
         host[3] = 0.0.0.0
```

SNTP must be enabled before the MAX TNT can communicate using that protocol. In addition, you must specify at least one IP address of an SNTP server. The host parameter lets you specify up to three server addresses. The MAX TNT always communicates with the first address unless it is inaccessible. In that case, the MAX TNT attempts to communicate with the second address, trying the third address only if the other two are inaccessible.

With the GMT-offset parameter, you specify your time zone as an offset from the Universal Time Configuration (UTC). UTC is in the same time zone as Greenwich Mean Time (GMT), and you specify the offset in hours and minutes using a 24-hour clock. Because some time zones, such as Newfoundland, cannot use an even hour boundary, the offset includes four digits. It requires half-hour increments. For example, in Newfoundland the time is 1.5 hours ahead of UTC, which is represented as follows:

```
UTC +0130
```

For San Francisco, which is 8 hours ahead of UTC:

```
UTC +0800
```

For Frankfurt, which is 1 hour behind UTC:

```
UTC -0100
```

Following is an example that specifies the time zone for San Francisco and the addresses of two SNTP servers:

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

```
admin> list sntp-info
enabled = no
GMT-offset = utc+0000
host = [0.0.0.0 0.0.0.0 0.0.0.0]
admin> set enabled = yes
admin> set gmt = utc+0800
admin> set host 1 = 10.2.3.4
admin> write
IP-GLOBAL written
```

Configuring WAN interfaces

This section describes how to configure Connection profiles for IP routing connections. It covers the following topics:

- Listing the IP subprofile of a Connection profile
- Enabling IP routing for a WAN connection
- Example of a connection to a remote IP router
- Example of a dial-in host requiring a host route
- Example of a dial-in host requiring address assignment
- Example of a numbered interface WAN connection
- Configuring WAN routing policies and preferences
- Using client DNS
- Specifying client default gateways
- Specifying IP-Direct connections

For information about OSPF routing, see Chapter 5, “OSPF Router Configuration.” For information about multicast forwarding, see Chapter 6, “Multicast Forwarding.”

Listing the IP subprofile of a Connection profile

For information on configuring the encapsulation, telco, and session options required to build a connection, see Chapter 2, “WAN Connections.” Following is an example that shows the commands entered to open a Connection profile and list the ip-options subprofile, and the system’s responses:

```
admin> read conn test
CONNECTION/test read
admin> list ip-options
ip-routing-enabled = yes
vj-header-prediction = yes
remote-address = 0.0.0.0/0
local-address = 0.0.0.0/0
routing-metric = 7
preference = 100
down-preference = 255
private-route = no
```

```
multicast-allowed = no
address-pool = 0
ip-direct = 0.0.0.0
rip = routing-off
ospf-options = { no 0.0.0.0 normal 10 30 120 5 simple +
multicast-rate-limit = 100
client-dns-primary-addr = 0.0.0.0
client-dns-secondary-addr = 0.0.0.0
client-dns-addr-assign = yes
client-default-gateway = 0.0.0.0/0
```

For complete information about each parameter, see the *MAX TNT Reference Guide*.

Enabling IP routing for a WAN connection

The following parameters enable IP routing and TCP header compression, shown with their default values:

```
CONNECTION station
   ip-options
      ip-routing-enabled = yes
      vj-header-prediction = yes
```

By default, all new Connection profiles enable the routing of IP data packets (the ip-routing-enabled parameter is set to Yes). The vj-header-prediction parameter is also set to Yes by default, which specifies negotiation of Van Jacobson header prediction for TCP packets on incoming calls using encapsulation protocols that support this feature. You can change the defaults if necessary, but they are appropriate for most IP routing connections.

Example of a connection to a remote IP router

The following parameter specifies the IP address of the remote router, shown with a sample setting:

```
CONNECTION station
   ip-options
      remote-address = 10.7.8.200/24
```

When the remote station calls in, the MAX TNT matches the caller's source IP address to this parameter find the right Connection profile. Figure 4-6 shows the MAX TNT connecting to a remote router, such as an Ascend Pipeline. This could be a telecommuting configuration, for example, where the Pipeline is located at a branch or home office.

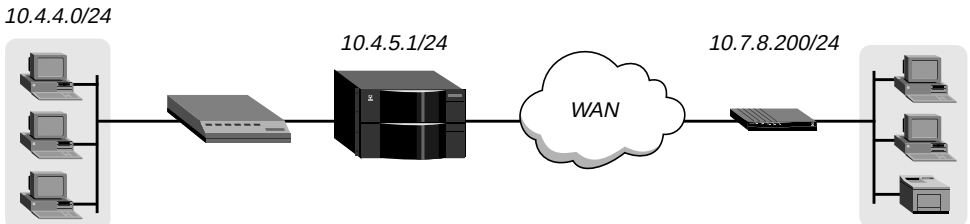


Figure 4-6. Router-to-router IP connection

Because the MAX TNT includes a subnet mask in its own local IP address, it must use other routers to route to local IP addresses outside that subnet. To forward packets to other parts of the corporate network, the MAX TNT must either have a static route configuration to a router in its own subnet (such as the CPE router in Figure 4-6), or it must enable RIP on Ethernet. For related information, see "Example of a default route" on page 4-39.

Note: If you do not specify the subnet mask in the remote-address parameter, the MAX TNT inserts a default mask that assumes the entire far-end network is accessible. Normally, if the far-end router's address includes a mask, you should include it.

The default settings for the ip-options subprofile enable IP routing and Van Jacobsen header compression and turn RIP off. Those are the appropriate settings for the following example, which configures a Connection profile for the Pipeline in Figure 4-6:

```
admin> read conn pipeline-1
CONNECTION/pipeline-1 read
admin> set active = yes
admin> set encapsulation-protocol = ppp
admin> set dial-number = 9-1-333-555-1212
admin> set ppp send-password = remotepw
admin> set ppp recv-password = localpw
admin> set ip-options remote = 10.7.8.200/24
admin> write
CONNECTION/pipeline-1 written
```

To specify the local CPE router as the MAX TNT unit's default route:

```
admin> read ip-route default
IP-ROUTE/default read
admin> set gateway = 10.4.4.133
admin> write
IP-ROUTE/default written
```

For information about configuring other Connection profile parameters, see Chapter 2, "WAN Connections," and Appendix A, "Authentication Methods."

Example of a dial-in host requiring a host route

The following parameter specifies the IP address of a dial-in host running PPP software:

```
CONNECTION station
   ip-options
      remote-address = 10.8.9.10/32
```

A host route is advertised as an IP address with a subnet mask of 32. It represents a single host rather than a remote router. Figure 4-7 shows a sample connection in which a dial-in host with an ISDN modem card calls into the MAX TNT.

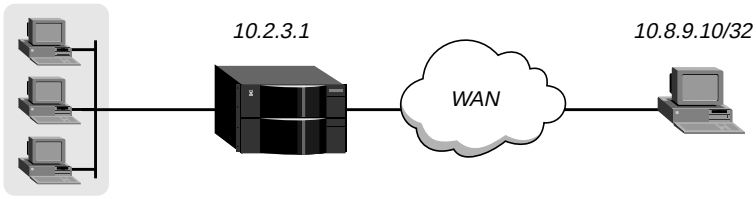


Figure 4-7. Dial-in host requiring a static IP address (a host route)

The PPP software is configured with the host's address, for example:

```
Username=patti
Accept Assigned IP=N/A (or No)
IP address=10.8.9.10
Netmask=255.255.255.255
Default Gateway=N/A (or None)
Name Server=10.7.7.1
Domain suffix=abc.com
VAN Jacobsen compression ON
```

The default settings for the ip-options subprofile enable IP routing and Van Jacobsen header compression and turn RIP off. Those settings are appropriate for the following example, which configures the Connection profile for the host:

```
admin> new conn patti
CONNECTION/patti read
admin> set active = yes
admin> set encapsulation-protocol = ppp
admin> set ppp recv-password = localpw
admin> set ip-options remote = 10.8.9.10/32
admin> write
CONNECTION/patti written
```

Example of a dial-in host requiring address assignment

This section assumes that you have configured address pools in the IP-Global profile, as described in "Configuring address pools for dynamic assignment to dial-in hosts" on page 4-17. Following are the parameters related to assigning an address to a dial-in connection, shown with their default values:

```
ANSWER-DEFAULTS
   ip-answer
      assign-address = no
CONNECTION station
   ip-options
      remote-address = 0.0.0.0/0
      address-pool = 0
```

If the remote device is a dial-in host that accepts dynamic address assignment, leave the remote-address parameter blank and specify the number of the pool from which the MAX TNT can obtain an address for dynamic assignment to the host. Figure 4-8 shows the MAX TNT assigning an address from one of its defined pools to a dial-in host:

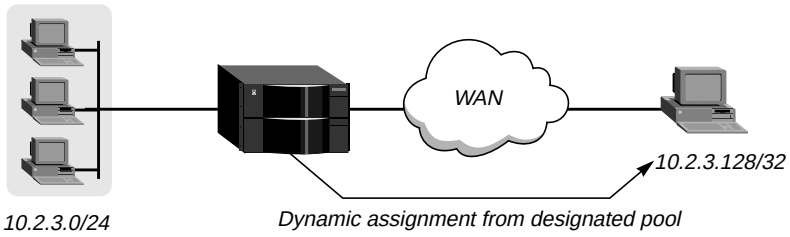


Figure 4-8. Dial-in host requiring assigned IP address

This example does not use the pool summary feature that enables the MAX TNT to advertise the entire pool of addresses rather than individual host routes for each assignment. See “Configuring address pools for dynamic assignment to dial-in hosts” on page 4-17 for details on pool summary.

The PPP software on the dial-in host in Figure 4-8 is configured to acquire its IP address dynamically. For example:

```
Username=victor
Accept Assigned IP=Yes
IP address=Dynamic (or Assigned or N/A)
Netmask=255.255.255.255 (or None or N/A)
Default Gateway=None or N/A
Name Server=10.2.3.55
Domain suffix=abc.com
Baud rate=38400
Hardware handshaking ON
VAN Jacobsen compression ON
```

Following is an example that configures a Connection profile to assign an IP address dynamically when the host dials in:

```
admin> read answer
ANSWER-DEFAULTS read
admin> set ip-answer assign-address = yes
admin> write
ANSWER-DEFAULTS written
admin> new conn victor
CONNECTION/victor read
admin> set active = yes
admin> set encapsulation-protocol = ppp
admin> set ppp recv-password = localpw
admin> set ip-options address-pool = 0
admin> write
CONNECTION/victor written
```

When address-pool is 0, the MAX TNT gets an IP address for this host from the first defined address pool. To assign an address within a specific range, specify a pool number instead of from 1 to 128.

Example of a numbered interface WAN connection

This section assumes that you have configured a local interface with more than one IP address, as described in "Using numbered interfaces" on page 4-9. To configure the WAN side of a numbered interface, use the following parameter (shown with sample settings):

```
CONNECTION station
   ip-options
      remote-address = 11.123.4.5/24
      local-address = 10.9.1.212/24
```

Figure 4-9 shows a numbered interface connection. Note that unless the connection is configured as a private route, the route to this local subnet will be added to the routing table.

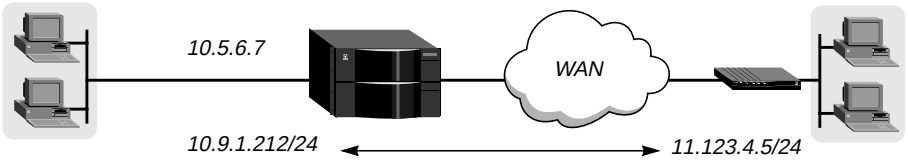


Figure 4-9. Example of a numbered interface connection

Following is an example that configures a numbered interface connection:

```
admin> new conn numbered
CONNECTION/numbered read
admin> set active = yes
admin> set encapsulation-protocol = ppp
admin> set ppp recv-password = localpw
admin> set ip-options remote-address = 11.123.4.5/24
admin> set ip-options local-address = 10.9.1.212/24
admin> write
CONNECTION/numbered written
```

Configuring WAN routing policies and preferences

At system startup, the MAX TNT adds a static route to the routing table for each LAN IP interface (configured IP-Interface profile) and for each WAN IP interface (configured Connection profile). For example, for a Connection profile with the following remote address:

```
remote-address = 10.9.8.10/22
```

the MAX TNT creates a static route with the following addresses:

```
dest-address = 10.9.8.10/22
gateway-address = 10.9.8.10
```

Because each WAN connection is a routing table entry, you can configure routing policies and preferences for the route, just as you would for a static IP-Route profile. The following parameters configure routing policies and route preferences for the WAN connection:

```
CONNECTION station
   ip-options
      routing-metric = 1
```

```
preference = 100
down-preference = 120
private-route = no
rip = routing-off
```

For complete information about each parameter, see the *MAX TNT Reference Guide*.

Assigning a metric to the connection

Connection profiles often represent switched connections, which have an initial cost that you avoid if you use a nailed-up link to the same destination. The lower the metric assigned to a route (a connection), the more likely it is to be used as a means to a destination address. To favor nailed-up links, you can assign a higher metric to switched connections than to any of the nailed-up links that can go to the same place. Following is an example that assigns a high metric to a connection:

```
admin> read conn david
CONNECTION/david read
admin> set ip-options routing-metric = 7
admin> write
CONNECTION/david written
```

For information about configuring route metrics in RIP packets, see Chapter 7, “Packet and Route Filters.”

Assigning a preference and down-preference

When choosing which route to use, the router first compares the preference values, preferring the lower number. If the preference values are equal, the router compares the metric values, using the route with the lower metric. The value of 255 means “Don’t use this route.” For a discussion of route preferences see “Configuring system-level routing policies and preferences” on page 4-22.

The down-preference is set to a high number, so the MAX TNT will look for other routes when this connection is down. Following is an example that assigns preference values:

```
admin> read conn david
CONNECTION/david read
admin> set ip-options preference = 50
admin> set ip-options down-preference = 255
admin> write
CONNECTION/david written
```

Making the connection route private

The private-route parameter specifies whether the MAX TNT will disclose the existence of this route when queried by RIP or another routing protocol. Private routes are used internally but are not advertised.

In some cases, making a route private is recommended. In the case of the pool summary feature, it is required. See “Setting up summarized address pools (pool summary)” on page 4-19. Following is an example that makes the Connection profile route private:

```
admin> read conn david
CONNECTION/david read
admin> set ip-options private-route = yes
admin> write
CONNECTION/david written
```

Enabling RIP on the connection

When you enable RIP in a Connection profile, you can specify whether the MAX TNT sends RIP updates across the WAN connection (informing other routers on the remote network of its routes), receives RIP updates from the remote router (including those routes in its routing table), or both. For related information, see “Enabling RIP on a LAN interface” on page 4-11. Following is an example that sets the unit to both send and receive RIP update packets on this interface:

```
admin> read conn david
CONNECTION/david read
admin> set ip-options rip = routing-send-and-recv-v2
admin> write
CONNECTION/david written
```

You should run RIP version 2 (RIP-v2) if possible. Ascend does not recommend running RIP-v2 and RIP-v1 on the same network in such a way that the routers receive each other’s advertisements. RIP-v1 guesses subnet masks, while RIP-v2 handles them explicitly. Running the two versions on the same network can result in RIP-v1 guesses overriding accurate subnet information obtained via RIP-v2.

For information about filtering routes in RIP update packets, see Chapter 7, “Packet and Route Filters.”

Using client DNS

Client DNS configurations define DNS servers that will be presented to WAN connections during IPCP negotiation. They provide a way to protect your local DNS information from WAN users. For details, see Appendix B, “Authorization Options.”

Specifying client default gateways

A client default gateway is a connection-specific next-hop router. All packets received across the WAN connection are forwarded to the specified router. Client default gateways are typically used to ensure that traffic associated with a particular on-line service is sent through the router operated by that service.

The MAX TNT must be able to reach the specified router directly, in one hop. If the specified router is not a legitimate next-hop router, the MAX TNT drops packets received on the connection. It is important to bear this in mind if you encounter routing problems after configuring a client default gateway, because an error in the next-hop router address will not be apparent by checking the global routing table.

You configure a client default gateway by specifying its address in this parameter:

```
CONNECTION station
   ip-options
      client-default-gateway = 0.0.0.0
```

For example:

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

admin> set ip-options client-default-gateway = 17.1.1.1

admin> write
CONNECTION/test written
```

When a client default gateway is specified for a WAN connection, all packets received across the connection are forwarded to the specified next-hop router. If the remote device is another access router, the default gateway is used for packets sent by all hosts behind that router.

While all packets arriving on the interface using this profile are affected, packets from other users or from the Ethernet are handled normally. The global routing table is not altered by use of this feature.

When a client default gateway is specified, the MAX TNT receives packets across this connection and consults the routing table in the usual way. It looks first for a specific route that matches the destination. If it finds no explicit route for packets received across this WAN connection, it uses the client default gateway instead of using the system default route (destination 0.0.0.0), or instead of dropping the packet, if it finds no explicit route and no system default route has been configured.

Specifying IP-Direct connections

An IP Direct configuration enables IP packets received from an incoming connection to bypass the routing tables and be redirected instead to a specified next-hop destination IP address. Outbound packets are routed as usual. At this time, the feature is implemented only for data calls. Following is the related parameter, shown with a sample setting:

```
CONNECTION station
   ip-options
      ip-direct = 10.1.2.3/24
```

The ip-direct parameter specifies the IP address of a next-hop destination to which all IP packets received across this link will be directed. The default is 0.0.0.0, which means that IP-Direct is disabled. Figure 4-10 shows an example of the IP-Direct traffic flow.

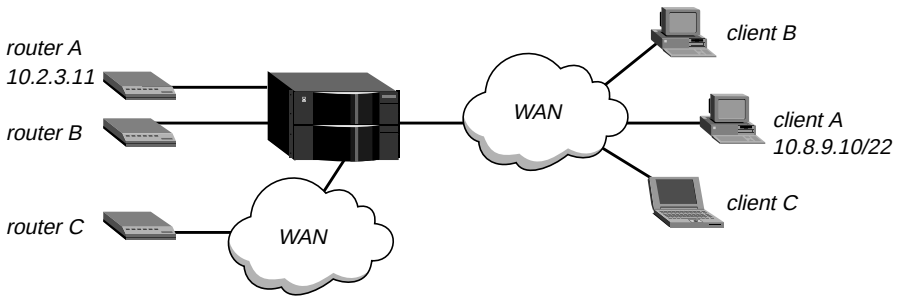


Figure 4-10. IP Direct connections

In Figure 4-10:

- The Connection profile for client A uses ip-direct to redirect inbound packets to router A on the LAN side of the MAX TNT.
- The Connection profile for client B uses ip-direct to redirect inbound packets to router B on the LAN side of the MAX TNT.
- The Connection profile for client C uses ip-direct to redirect inbound packets to router C via a switched connection.

Packets destined for the clients A, B, or C are routed normally by the MAX TNT, which means that these client connections can *receive* packets from any source, not just from the IP address to which their packets are sent.

Following is an example that configures an IP-Direct Connection profile for client A in Figure 4-10:

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

admin> set active = yes

admin> set encapsulation-protocol = ppp

admin> set ppp recv-password = localpw

admin> set ip-options remote = 10.8.9.10/22

admin> set ip-options ip-direct = 10.2.3.11

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

IP-Direct connections require the following special handling:

- If the profile enables the receipt or receipt-transmission of RIP updates, all RIP packets from an incoming connection are kept locally and forwarded to the ip-direct address, so that the MAX TNT can correctly forward packets *destined* for the client.
- ARP requests received from the incoming connection are ignored.
- The caller cannot Telnet to the MAX TNT.

All incoming packets on this connection are forwarded to the ip-direct address. As a side-effect, a user on the remote network cannot Telnet to the MAX TNT, because the connection is passed through to the ip-direct host.

Working with static IP routes

When the MAX TNT starts up, it initially builds the routing table from its known static routes, which include those defined in IP-Interface profiles, Connection profiles, and IP-Route profiles. The routes in IP-Route profiles are also passed to the router whenever a route changes. Following are the related parameters, shown with sample settings:

```
IP-ROUTE name
   name* = default
   dest-address = 0.0.0.0/0
   gateway-address = 10.2.3.17
   metric =1
   cost =1
```

```
   preference = 100
   third-party = no
   ase-type = type-1
   ase-tag = c0:00:00:00
   private-route = yes
   active-route = yes
   ase7-adv = N/A
```

Where to find information about OSPF-related settings

The following parameters (shown with default values) apply only when OSPF is enabled:

```
IP-ROUTE name
   cost =1
   third-party = no
   ase-type = type-1
   ase-tag = c0:00:00:00
   ase7-adv = N/A
```

For information about how to configure IP-Route profiles for OSPF, see Chapter 5, "OSPF Router Configuration."

Example of a default route

When the MAX TNT consults its routing table, if it does not find a specific match for the packet's destination address, it looks for a default route. The default route specifies a zero destination address (0.0.0.0), which is interpreted as "any" destination. If the MAX TNT finds a default route, it forwards the packet to the specified gateway address (next-hop router). If it finds no specific destination match and no default route, the MAX TNT drops the packet.

Following are the parameters that define the default route, shown with an example gateway address:

```
IP-ROUTE default
   dest-address = 0.0.0.0/0
   gateway-address = 10.4.4.133/24
```

Figure 4-11 shows a router on a local subnet configured as the default route in a MAX TNT. This type of configuration enables the MAX TNT to turn off RIP on its local interfaces, and forward all local packets to the default route.

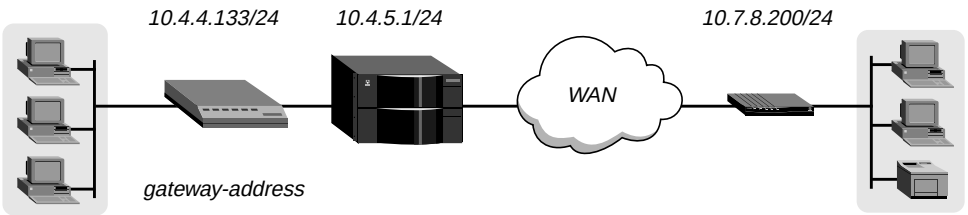


Figure 4-11. Default route to a local IP router

Note: If you do not configure a default route, the MAX TNT drops packets for which it has no route. Many sites configure a local router as the default route to offload routing overhead to local networks and subnets.

The name of the default IP-Route profile is always “default” and its destination is always 0.0.0.0. The following example shows commands entered to configure the default route, and the system’s responses:

```
admin> read ip-route default
IP-ROUTE/default read
admin> set gateway-address = 10.4.4.133/24
admin> write
IP-ROUTE/default written
```

Example of a static route

When dynamic route-discovery protocols such as RIP or OSPF are turned off on an IP interface, the router does not learn about routers on that interface unless it has a static route. Following are the parameters that define a static route, shown with sample settings:

```
IP-ROUTE name
   dest-address = 10.2.3.56/28
   gateway-address = 10.2.3.17/28
```

For example, if a Connection profile specifies the destination address of a host on a remote subnet, but the packets must be routed through an intermediary device to reach that host (and RIP or OSPF is not enabled), you must configure a static route specifying the intermediary device as the gateway. Figure 4-12 shows an example.

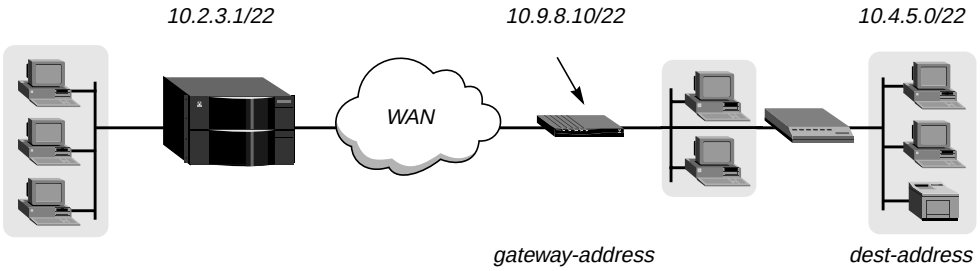


Figure 4-12. Static route to a remote subnet

Following is an example that configures a static route to the remote subnet in Figure 4-12:

```
admin> new ip-route subnet
IP-ROUTE/subnet read
admin> set dest = 10.4.5.0/22
admin> set gateway = 10.9.8.10
admin> write
IP-ROUTE/subnet written
```

Assigning a metric and preference to a static route

The metric parameter is a virtual hop count (a number between 1 to 15) for the specified route. The higher the metric, the less likely that the MAX TNT will use a route.

The preference parameter specifies a route preference. Zero is the default for connected routes (such as the Ethernet). When choosing which route to use, the router first compares the

preference values, preferring the lower number. If the preference values are equal, the router compares the metric values, using the route with the lower metric. The value of 255 means "Don't use this route." For a discussion of route preferences see "Configuring system-level routing policies and preferences" on page 4-22.

Following is an example that sets a relatively low metric and preference value for a route:

```
admin> read ip-route subnet
IP-ROUTE/subnet read
admin> set metric = 2
admin> set preference = 50
admin> write
IP-ROUTE/subnet written
```

For information about configuring route metrics in RIP packets, see Chapter 7, "Packet and Route Filters."

Making a static route private

The private-route parameter specifies whether the MAX TNT will disclose the existence of this route when queried by RIP or another routing protocol. Private routes are used internally but are not advertised. Following is an example that makes the route private:

```
admin> read ip-route subnet
IP-ROUTE/subnet read
admin> private-route = yes
admin> write
IP-ROUTE/subnet written
```

Making a static route temporarily inactive

The active-route parameter is set to Yes by default, indicating that the route should be entered in the routing table. If you set the parameter to No, the MAX TNT leaves the route out of its routing table. This is a useful alternative to deleting the profile, if you might want to reinstate the route later.

Example of static multipath routes

Multipath static routes distribute traffic to one destination across the aggregated bandwidth of multiple interfaces. A multipath route requires static routes that meet the following criteria:

- The routes have the same destination address and subnet mask, but different gateway addresses.
- The routes have the same route metric or OSPF cost.
- The routes have the same route preference.

If more than one IP-Route profile has a destination of 0.0.0.0, the MAX TNT creates a multipath default route. Following is an example that configures a multipath route to the network 10.76.109.0/24:

```
admin> new ip-route bdvnet-1
IP-ROUTE/bdvnet-1 read
```

```
admin> set dest = 10.76.109.0/24
admin> set gateway = 206.65.212.1
admin> set metric = 2
admin> write
IP-ROUTE/bdvnet-1 written
admin> new ip-route bdvnet-2
IP-ROUTE/bdvnet-2 read
admin> set dest = 10.76.109.0/24
admin> set gateway = 206.65.210.1
admin> set metric = 2
admin> write
IP-ROUTE/bdvnet-2 written
```

The multipath routes appear in the routing table with the M (multipath) flag, for example:

```
admin> netstat -rn
Destination      Gateway        IF       Flg   Pref Met    Use     Age
...
10.76.109.0/24   206.65.212.1   ie1-12-2 SGM    100   2    20     7772
10.76.109.0/24   206.65.210.1   ie1-12-3 SGM    100   2     24    7772
```

OSPF Router Configuration

5

This chapter covers the following topics:

Introduction to OSPF

OSPF (Open Shortest Path First) is the next generation Internet routing protocol. The “Open” in its name refers to the fact that OSPF was developed in the public domain as an open specification. The “Shortest Path First” refers to an algorithm developed by Dijkstra in 1978 for building a self-rooted shortest-path tree from which routing tables can be derived. This algorithm is described in “The link-state routing algorithm” on page 5-8.

RIP limitations solved by OSPF

The rapid growth of the Internet has pushed RIP (Routing Information Protocol) beyond its capabilities, particularly in the areas of distance-vector metrics, the 15-hop limitation, and excessive routing traffic causing slow convergence.

Distance-vector metrics

RIP is a distance-vector protocol, which uses a hop count to select the shortest route to a destination network. RIP always uses the lowest hop count, regardless of the speed or reliability of a link.

OSPF is a link-state protocol, which means that OSPF can take into account a variety of link conditions, such as the reliability or speed of the link, when determining the best path to a destination network.

15-hop limitation

A destination that requires more than 15 consecutive hops is considered unreachable, which inhibits the maximum size of a network. OSPF has no hop limitation—you can add as many routers to a network as you want.

Excessive routing traffic and slow convergence

RIP creates a routing table and then propagates it throughout the internet of routers, hop by hop. The time it takes for all routers to receive information about a topology change is called “convergence.” A slow convergence can result in routing loops and errors.

A RIP router broadcasts its entire routing table every 30 seconds. On a 15-hop network, convergence can be as high as 7.5 minutes. In addition, a large table may require multiple broadcasts for each updates, which consumes a lot of bandwidth. OSPF uses a topological database of the network and propagates only changes to the database. See “Exchange of routing information” on page 5-4.

Ascend implementation of OSPF

The primary goal of OSPF at this release is to allow the MAX TNT to communicate with other routers within a single autonomous system (AS).

The MAX TNT acts as an OSPF internal router with limited border router capability. At this release, we do not recommend an ABR configuration for the MAX TNT, so the Ethernet interface and all of the MAX TNT WAN links should be configured in the same area.

At this release, the MAX TNT does not function as an IGP gateway, although ASBR calculations are always performed for external routes (such as WAN links that do not support OSPF). The MAX TNT imports external routes into its OSPF database and flags them as ASE (autonomous system external). It redistributes those routes via OSPF ASE advertisements, and propagates its OSPF routes to remote WAN routers running RIP.

The MAX TNT supports null and simple password authentication.

Diagnostic commands

The OSPF diagnostic-level commands enable the administrator to display information related to OSPF routing, including the link state advertisements (LSAs), border routers' routing table, and the OSPF areas, interfaces, statistics, and routing table. The following command displays the usage statement:

```
admin> ospf ?

ospf usage:
ospf ?              Display help information
ospf areas          Display OSPF areas
ospf border-routers Display OSPF border router information
ospf database       Display OSPF link-state database
ospf errors         Display OSPF errors
ospf general        Display OSPF general info
ospf interfaces     Display OSPF interfaces
ospf neighbor       Display OSPF neighbors
ospf rtab           Display OSPF routing tab
ospf timer-queue    Display OSPF timer queue
ospf stats          Display OSPF stats
```

For information about using these commands, see the *MAX TNT Reference Guide*.

OSPF features

This section provides a brief overview of OSPF routing to help you configure the MAX TNT properly. For full details about how OSPF works, see RFC 1583, “OSPF Version 2", 03/23/1994, J. Moy.

An AS (autonomous system) is a group of OSPF routers exchanging information, typically under the control of one company. An AS can include a large number of networks, all of which are assigned the same AS number. All information exchanged within the AS is “interior.”

Exterior protocols are used to exchange routing information between autonomous systems. The are referred to by the acronym EGP (exterior gateway protocol). The AS number may be used by border routers to filter out certain EGP routing information. OSPF can make use of EGP data generated by other border routers and added into the OSPF system as ASEs, as well as static routes configured in the MAX TNT or RADIUS.

Security

All OSPF protocol exchanges are authenticated. This means that only trusted routers can participate in the AS's routing. A variety of authentication schemes can be used; in fact, different authentication types can be configured for each area. In addition, authentication provides added security for the routers that are on the network. Routers that do not have the

password will not be able to gain access to the routing information, because authentication failure prevents a router from forming adjacencies.

Support for variable length subnet masks

OSPF enables the flexible configuration of IP subnets. Each route distributed by OSPF has a destination and mask. Two different subnets of the same IP network number may have different sizes (different masks). This is commonly referred to as variable-length subnet masks (VLSM), or Classless Inter-Domain Routing (CIDR). A packet is routed to the best (longest or most specific) match. Host routes are considered to be subnets whose masks are "all ones" (0XFFFFFFFF).

Note: Although OSPF is very useful for networks that use VLSM, we recommend that you attempt to assign subnets that are as contiguous as possible in order to prevent excessive link-state calculations by all OSPF routers on the network.

Interior gateway protocol (IGP)

OSPF keeps all AS-internal routing information within that AS. All information exchanged within the AS is "interior."

An AS border router (ASBR) is required to communicate with other autonomous systems by using an external gateway protocol (EGP), as shown in Figure 5-1. An EGP acts as a shuttle service between autonomous systems.

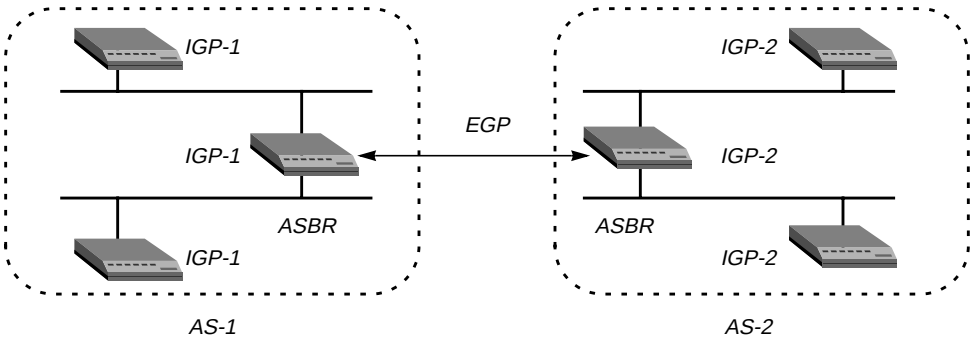


Figure 5-1. Autonomous system border routers

ASBRs perform calculations related to external routes. The MAX TNT imports external routes from RIP—for example, when it establishes a WAN link with a caller that does not support OSPF—and the ASBR calculations are always performed.

Exchange of routing information

OSPF uses a topological database of the network and propagates only changes to the database. Part of the SPF algorithm involves acquiring neighbors, and then forming an adjacency with one neighbor, as shown in Figure 5-2.

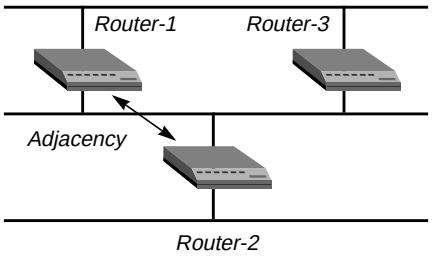


Figure 5-2. Adjacency between neighboring routers

An OSPF router dynamically detects its neighboring routers by sending its Hello packets to the multicast address AllSPFRouters. It attempts to form adjacencies with some of its newly acquired neighbors.

Adjacency is a relationship formed between selected neighboring routers for the purpose of exchanging routing information. Not every pair of neighboring routers become adjacent. Adjacencies are established during network initialization in pairs, between two neighbors. As the adjacency is established, the neighbors exchange databases and build a consistent, synchronized database between them.

When an OSPF router detects a change on one of its interfaces, it modifies its topological database and multicasts the change to its adjacent neighbor, which in turn propagates the change to its adjacent neighbor until all routers within an area have synchronized topological databases. This results in quick convergence among routers. OSPF routes can also be summarized in link-state advertisements (LSAs).

Designated and backup designated routers

In OSPF terminology, a broadcast network is any network that has more than two OSPF routers attached and supports the capability to address a single physical message to all of the attached routers.

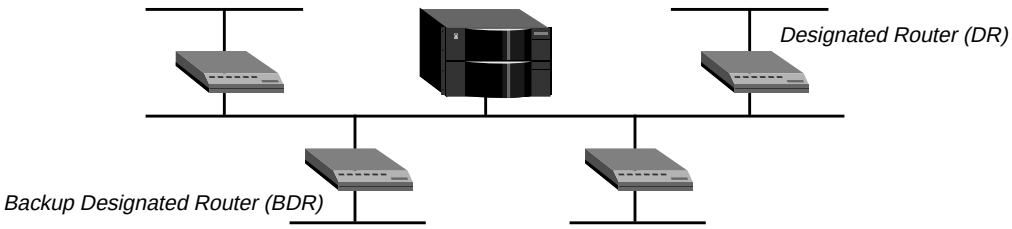


Figure 5-3. Designated and backup designated routers

Note: The MAX TNT can function as a designated router (DR) or backup designated router (BDR). However, many sites choose to assign a LAN-based router for these roles in order to dedicate the MAX TNT to WAN processing. The administrator chooses a DR and BDR based on the device's processing power and reliability.

To reduce the number of adjacencies each router must form, OSPF calls one of the routers the designated router. A designated router is elected as routers are forming adjacencies, and then all other routers establish adjacencies only with the designated router. This simplifies the routing table update procedure and reduces the number of link-state records in the database.

The designated router plays other important roles as well to reduce the overhead of a OSPF link-state procedures. For example, other routers send link-state advertisements it to the designated router only by using the "all-designated-routers" multicast address of 224.0.0.6.

To prevent the designated router from becoming a serious liability to the network if it fails, OSPF also elects a backup designated router at the same time. Other routers maintain adjacencies with both the designated router and its backup router, but the backup router leaves as many of the processing tasks as possible to the designated router. If the designated router fails, the backup immediately becomes the designated router and a new backup is elected.

The administrator chooses which router is to be the designated router based on the processing power, speed, and memory of the system, and then assigns priorities to other routers on the network in case the backup designated router is also down at the same time.

Configurable metrics

The administrator assigns a cost to the output side of each router interface. The lower the cost, the more likely the interface is to be used to forward data traffic. Costs can also be associated with the externally derived routing data.

The OSPF cost can also be used for preferred path selection. If two paths to a destination have equal costs, you can assign a higher cost to one of the paths to configure it as a backup to be used only when the primary path is not available.

Figure 5-4 shows how costs are used to direct traffic over high-speed links. For example, if Router-2 in Figure 5-4 receives packets destined for Host B, it will route them through Router-1 across two T1 links (Cost=20) rather than across one 56kbps B-channel to Router-3 (Cost=240).

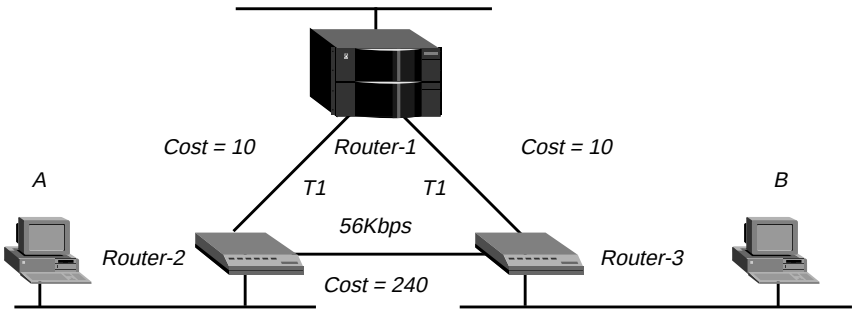


Figure 5-4. OSPF costs for different types of links

The MAX TNT has a default cost of 1 for a connected route (Ethernet) and 10 for a WAN link. If you have two paths to the same destination, the one with the lower cost will be used. You may want to reflect the bandwidth of a connection when assigning costs; for example, for a single B-channel connection, the cost would be 24 times greater than a T1 link.

Note: Be careful when assigning costs. Incorrect cost metrics can cause delays and congestion on the network.

Hierarchical routing (areas)

If a network is large, the size of the database, time required for route computation, and related network traffic become excessive. An administrator can partition an AS into areas to provide

hierarchical routing connected by a backbone. The backbone area is special and always has the area number 0.0.0.0. Other areas are assigned area numbers that are unique within the AS.

Each area acts as its own network: all area-specific routing information stays within the area, and all routers within an area must have a synchronized topological database. To tie the areas together, some routers belong to an area and to the backbone area. These routers are area border routers (ABRs). In Figure 5-5, all of the routers are ABRs:

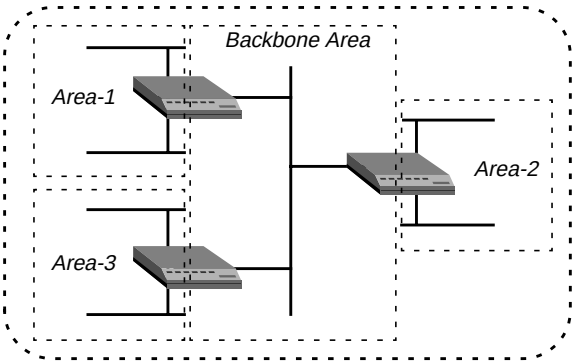


Figure 5-5. Dividing an AS into areas

Note: The MAX TNT does not currently operate as an ABR. Ascend recommends that you use the same area number for the Ethernet interface of the MAX TNT and each of its WAN links. That area number does not have to be the default backbone area (0.0.0.0).

If the ABRs and area boundaries are set up correctly, link-state databases are unique to an area. You can configure MAX TNT to route in three kinds of area, which differ in their handling of external routes.

- normal
- stub
- NSSA (Not So Stubby Area)

Normal areas

AS external routes are originated by AS boundary routers as type-5 link state advertisements (LSAs). An OSPF normal area allows type-5 LSAs to be flooded throughout the area.

Stub areas

For areas that are connected only to the backbone by one ABR (that is, the area has one exit point), there is no need to maintain information about external routes. To reduce the cost of routing, OSPF supports stub areas, in which all external routes are summarized by a default route. A stub area allows no type-5 LSAs to be propagated into or throughout the area, and instead depends on default routing to external destinations.

Because the MAX TNT does not currently operate as an ABR, you should not configure it to route OSPF in a stub area if any of its links are AS-external.

NSSAs

NSSAs are like stub areas in that they do not receive or originate type-5 LSAs. They differ from stub areas in that they can import AS external routes in a limited fashion.

A new type-7 LSA is defined for NSSAs in OSPF version 2. In the MAX TNT, ASE type-7s can be imported only from static route definitions. Type-7 LSAs differ from type-5 LSAs in the following ways:.

- Type-7 LSAs may be originated by and advertised throughout an NSSA.
- Type-7 LSAs are advertised only within a single NSSA—they are not flooded throughout the AS as are type-5 LSAs.

Note: Please see RFC 1587 for detailed information regarding the NSSA specification.

The link-state routing algorithm

Link-state routing algorithms require that all routers within a domain maintain synchronized (identical) topological databases, and that the databases describe the complete topology of the domain. An OSPF router's domain may be an AS or an area within an AS.

Based on the exchange of information among routers, OSPF routers create a link-state database, which is updated based on packet exchanges among the routers. Link-state databases are synchronized between pairs of adjacent routers (see "Exchange of routing information" on page 5-4). In addition, each OSPF router uses its link-state database to calculate a self-rooted tree of shortest paths to all destinations. The routing table is built from these calculated shortest-path trees.

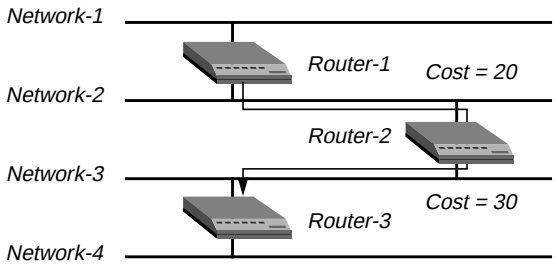


Figure 5-6. Sample network topology

For example, the link-state databases of the three routers shown in Table 5-1, next.

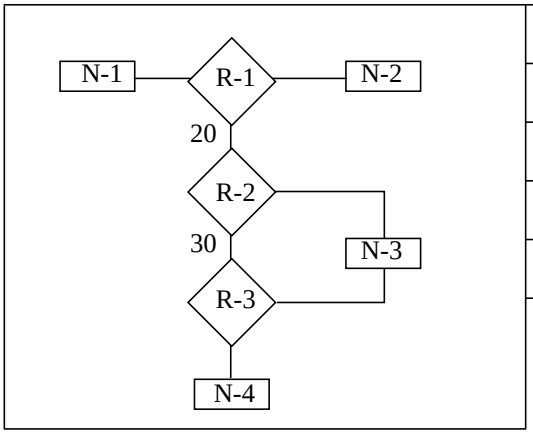
Table 5-1. Link state databases for network topology in Figure 5-6

Router-1	Router-2	Router-3
Network-1/Cost 0	Network-2/Cost0	Network-3/Cost 0
Network-2/Cost 0	Network-3/Cost0	Network-4/Cost 0
Router-2/Cost 20	Router-1/Cost 20	Router-2/Cost 30
	Router-3/Cost 30	

From the link-state database, each router builds a self-rooted shortest-path tree, and then calculates a routing table stating the shortest path to each destination in the AS as well as externally derived routing information. All of the routers calculate a routing table of shortest paths based on the link-state database. Externally derived routing data is advertised throughout the AS but is kept separate from the link-state data. Each external route can also be tagged by

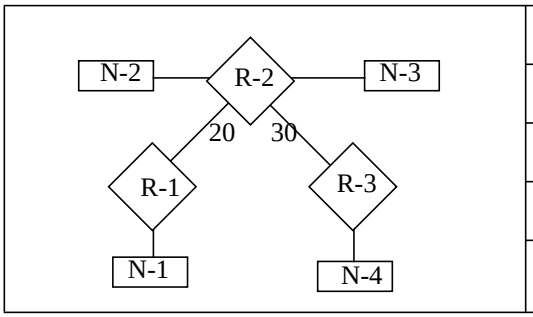
the advertising router, enabling the passing of additional information between routers on the boundary of the AS.

Table 5-2. Shortest-path tree and resulting routing table for Router-1



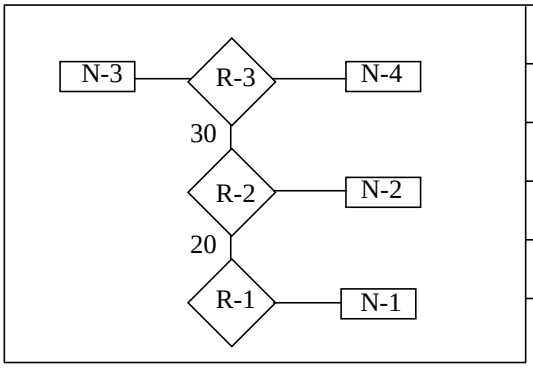
Destination	*Next Hop*	*Metric*
Network-1	Direct	0
Network-2	Direct	0
Network-3	Router-2	20
Network-4	Router-2	50

Table 5-3. Shortest-path tree and resulting routing table for Router-2



Destination	*Next Hop*	*Metric*
Network-1	Router-1	20
Network-2	Direct	0
Network-3	Direct	0
Network-4	Router-2	30

Table 5-4. Shortest-path tree and resulting routing table for Router-3



Destination	*Next Hop*	*Metric*
Network-1	Router-2	50
Network-2	Router-2	30
Network-3	Direct	0
Network-4	Direct	0

Configuring OSPF routing interfaces

This section describes how to add the MAX TNT to an OSPF network. It shows a local OSPF interface in a normal area, and one that routes OSPF across a WAN link. This section assumes that the MAX TNT is configured for IP, as described in Chapter 4, “IP Router Configuration.”

Following are the related parameters, shown with their default values:

```
IP-INTERFACE { { shelf-N slot-N N } N }
   ospf-options
      active = no
      area = 0.0.0.0
      area-type = normal
      hello-interval = 10
      dead-interval = 40
      priority = 5
      authen-type = simple
      auth-key = ascend0
      cost = 1
      down-cost = 16777215
      ase-type = type-1
      ase-tag = c0:00:00:00
      transit-delay = 1
      retransmit-delay = 5
CONNECTION station
   ip-options
      ospf-options
      active = no
      area = 0.0.0.0
      area-type = normal
      hello-interval = 30
      dead-interval = 120
      priority = 5
      authen-type = simple
      auth-key = ascend0
      cost = 10
      down-cost = 1000
      ase-type = type-1
      ase-tag = c0:00:00:00
      transit-delay = 1
      retransmit-delay = 5
```

For information about each parameter, see the *MAX TNT Reference Guide*.

OSPF configuration options

Note that the same parameters appear in the OSPF subprofiles of the IP-Interface and Connection profiles. This section provides an overview of what each parameter represents.

Enabling OSPF on an interface

To run OSPF on an interface, you must set active to Yes in the OSPF subprofile.

Specifying areas and area types

The default area type for a MAX TNT OSPF configuration is the “normal” area type, in which external routes are advertised throughout the AS. If you change this default for one interface in the unit, you must change it for all interfaces, because the MAX TNT does not currently

perform ABR functions. For background information about stub and NSSA areas, see “Hierarchical routing (areas)” on page 5-6.

Note: Because the MAX TNT does not currently operate as an area border router (ABR), all interfaces must be in the same area, and must specify the same area type.

In the MAX TNT, area numbers use dotted-decimal format. They are not IP addresses, although they use a similar format. You can use any area numbering scheme that is consistent throughout the AS and uses this format.

Hello and dead intervals

The hello-interval and dead-interval parameters specify the number of seconds between hello packets, and the number of elapsed seconds without receiving a hello packet the router will wait before considering its neighbor dead and instituting a link-state change. See “Exchange of routing information” on page 5-4.

Priority

The priority value is used to elect a designated router (DR) and backup designated router (BDR). For example, assigning a priority of 1 would place the MAX TNT near the top of the list of possible designated routers, which is generally not recommended, because acting as a DR or BDR significantly increases the amount of OSPF overhead for the router. See “Designated and backup designated routers” on page 5-5.

Area authentication

If authentication is required to get into the router’s area, specify the password. If authentication is not required, set authen-type to none. See “Security” on page 5-3.

Configurable costs

The lower the cost assigned to a route, the higher the likelihood of using that route to forward traffic. If you need to increase the cost of this router’s connection to Ethernet, you can raise the cost value. See “Configurable metrics” on page 5-6.

By default the cost of a connected route (an Ethernet interface) is 1. The main difference between the OSPF configuration on Ethernet and across the WAN is that the default cost and down-cost values are higher in a Connection profile. You can assign higher costs to reflect a slower connection or lower costs to set up a preferred route to a certain destination. If the cost of one route is lower than another to the same destination, the higher-cost route will not be used unless route preferences change that equation. (For information about route preferences, see Chapter 4, “IP Router Configuration.”)

Handling routes learned from RIP

The ase-tag field a hexadecimal number that shows up in management utilities and “flags” this route as external. It may also be used by border routers to filter this record.

The ase-type parameter can be set to a Type-1 metric (expressed in the same units as the interface cost) or a Type-2 metric (which is considered larger than any link-state path). This

parameter assumes that routing outside the AS is the major cost of routing a packet, and eliminates the need for conversion of external costs to internal link-state metrics.

In a Connection profile, the two ASE parameters have the same meaning as in an IP-Interface profile, but they are active only when OSPF is *not* active. When you configure these parameters, the Connection profile route will be advertised whenever the MAX TNT is up.

Transit delay and retransmit interval

The transit-delay parameter sets the estimated number of seconds it takes to transmit a Link State Update Packet over this interface, taking into account transmission and propagation delays. On a connected route, you can leave the default 1.

The retransmit-interval for OSPF packets specifies the number of seconds between retransmissions of Link-State Advertisements, Database Description and Link State Request Packets. On a connected route, you can leave the default 5.

Example of configuring OSPF on a LAN interface

Figure 5-7 shows five OSPF routers in the backbone area of an AS. Because all OSPF routers are in the same area, the units all form adjacencies and synchronize their databases together. This example shows how to configure the LAN interface of the MAX TNT labeled MAX-TNT-2 in Figure 5-7.

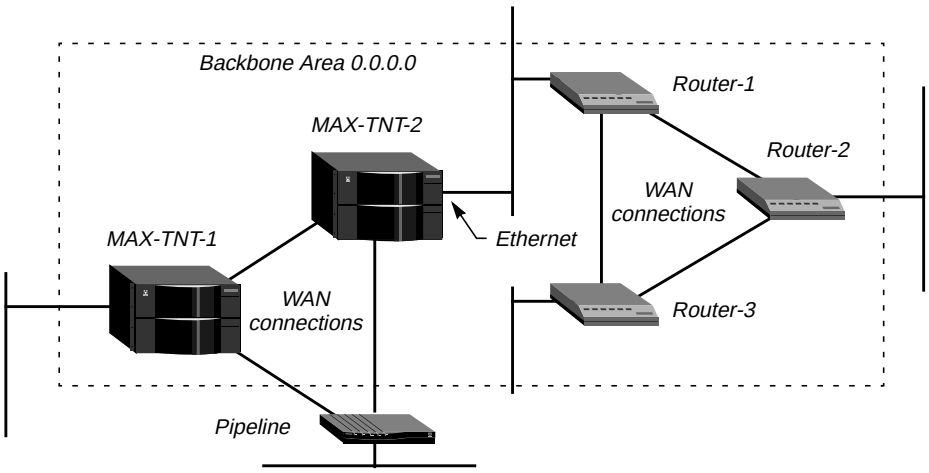


Figure 5-7. OSPF on a LAN interface

Note: All OSPF routers in Figure 5-7 have RIP turned off. It isn't necessary to run both RIP and OSPF, and it reduces processor overhead to turn RIP off. OSPF can learn routes from RIP, incorporate them in the routing table, assign them an external metric, and tag them as external routes.

Although there is no limitation stated in the RFC about the number of routers in the backbone area, it is recommended that you keep the number of routers relatively small, because changes that occur in area zero are propagated throughout the AS. Another way to configure the same units would be to create a second area (such as 0.0.0.1) in one of the existing OSPF routers,

and add the MAX TNT to that area. You can then assign the same area number (0.0.0.1) to all OSPF routers reached through the MAX TNT across a WAN link.

Following is an example that configures MAX TNT-2 in Figure 5-7 as an IP host on Ethernet with the IP address 10.168.8.17/24 on that interface:

```
admin> read ip-int {{ 1 c 1 } 0 }
IP-INTERFACE/{ { shelf-1 controller 1 } 0 } read
admin> set ip-address = 10.168.8.17/24
admin> set rip-mode = routing-off
admin> set ignore-def-route = yes
```

For information about IP configurations, see Chapter 4, “IP Router Configuration.” The following example configures MAX-TNT-2 in Figure 5-7 as an OSPF router in the backbone area:

```
admin> list ospf
active = no
area = 0.0.0.0
area-type = normal
hello-interval = 10
dead-interval = 40
priority = 5
authen-type = simple
auth-key = ascend0
cost = 1
down-cost = 16777215
ase-type = type-1
ase-tag = c0:00:00:00
transit-delay = 1
retransmit-delay = 5
admin> set active = yes
admin> write
IP-INTERFACE/{ { shelf-1 controller 1 } 0 } written
```

Note: When you write the IP-Interface profile, the MAX TNT comes up as an OSPF router on that interface. It forms adjacencies and begins building its routing table.

Example of configuring OSPF on WAN interfaces

This example shows how to configure Connection profiles in the MAX TNT units shown in Figure 5-8 to enable them to route OSPF across the WAN that separates them. In this example, the unit labeled MAX TNT-2 has the IP address 10.2.3.4/24, and the unit labeled MAX TNT-1 has the address 10.168.8.17/24.

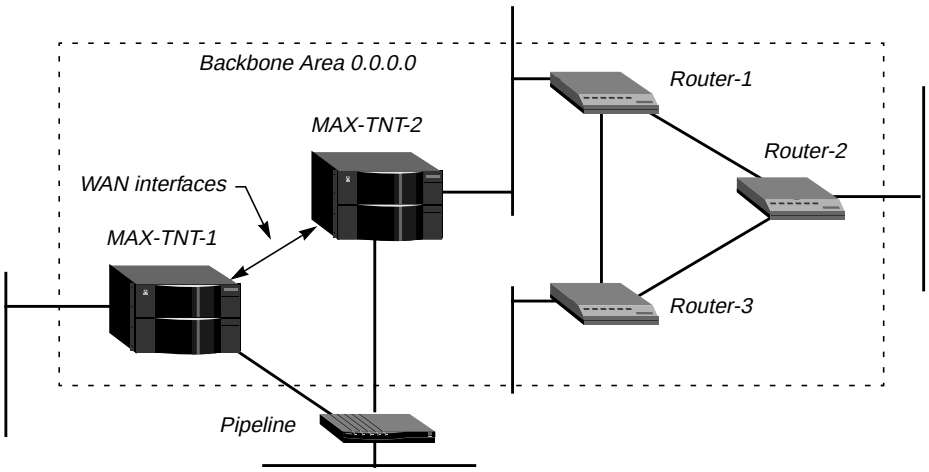


Figure 5-8. OSPF on WAN interfaces

The WAN interface of the MAX TNT is a point-to-point network; that is, it joins a single pair of routers. Point-to-point networks typically do not provide a broadcasting or multicasting service, so all advertisements are sent point to point.

Following is an example that configures the OSPF WAN link in the unit labeled MAX-TNT-1 in Figure 5-8:

```
admin> read conn maxtnt2ink
CONNECTION/maxtnt2link read
admin> set ip remote = 10.2.3.4/24
admin> set ip rip = routing-off
admin> set ip ospf active = yes
admin> write
CONNECTION/maxtnt2link written
```

Following is an example that configures the OSPF WAN link in the unit labeled MAX-TNT-2 in Figure 5-8:

```
admin> read conn maxtnt1link
CONNECTION/maxtnt1link read
admin> set ip remote = 10.168.8.17/24
admin> set ip rip = routing-off
admin> set ip ospf active = yes
admin> write
CONNECTION/maxtnt1link written
```

Example of integrating a RIP-v2 interface

In Figure 5-9, each MAX TNT has a WAN interface to a remote Pipeline unit. The Pipeline is an IP router that supports RIP-v2, and has the IP address 10.6.7.168/24. The route to the Pipeline LAN, as well as any routes the MAX TNT learns about from the remote Pipeline, are AS-external routes (external to the OSPF autonomous system).

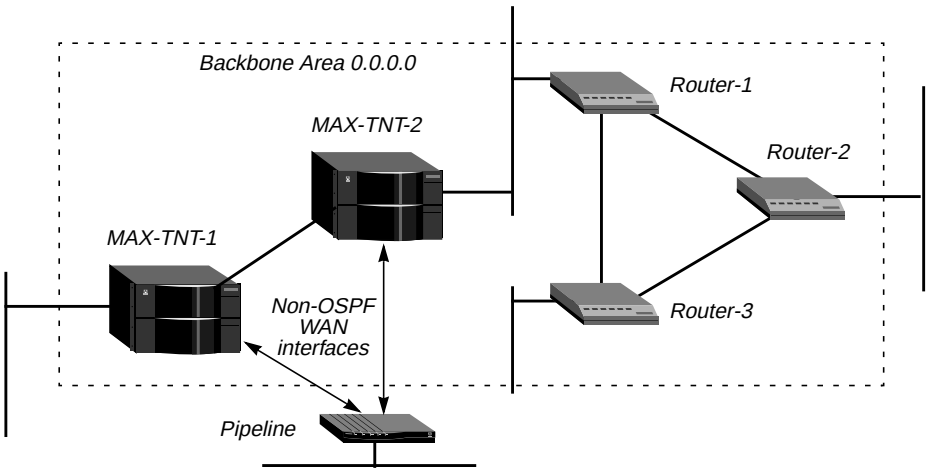


Figure 5-9. An interface that doesn't support OSPF

To enable OSPF to add the RIP-v2 routes to its routing table dynamically, you can configure RIP-v2 normally in the Connection profiles. OSPF will import all RIP routes as Type-2 ASEs. However, in this example, RIP is turned off on the link and ASE information is configured explicitly.

The following example shows the commands entered on either of the MAX TNT systems in Figure 5-9 to configure a link to the Pipeline, and the system's responses:

```
admin> read conn pipeline1
CONNECTION/pipeline1 read

admin> set ip remote = 10.6.7.168/24

admin> set ip rip = routing-off

admin> list ip ospf
active = no
area = 0.0.0.0
area-type = normal
stub-default-cost = 0
hello-interval = 30
dead-interval = 120
priority = 5
authen-type = simple
auth-key = ascend0
cost = 10
down-cost = 1000
ase-type = type-1
ase-tag = c0:00:00:00
transit-delay = 1
retransmit-delay = 5

admin> set active = no

admin> set cost = 240
```

This sets a cost of 240 for the route to the remote Pipeline. Typically, you should reflect the bandwidth of a connection when assigning costs; for example, for a single B-channel connection, the cost would be 24 times greater than a T1 link.

The next set of commands causes the MAX TNT to tag routes learned from RIP and to import them as type-2 LSAs:

```
admin> set ase-type = type-2
admin> set ase-tag = cfff8000
admin> write
CONNECTION/pipeline1 written
```

Example of an NSSA with a type-7 LSA

For background information about NSSAs, see “Hierarchical routing (areas)” on page 5-6. To configure the MAX TNT to route OSPF in an NSSA, all interfaces must be configured with an area-type of NSSA.

Note: To change from a normal area to an NSSA (or vice versa), a reset is required.

In addition, note that type-7 LSAs can only be imported from static route definitions. To configure a type-7 LSA in the MAX TNT, you must specify a static route in an IP-Route profile. Following are the related parameters, shown with sample settings:

```
IP-ROUTE name
   name* = external
   dest-address = 10.4.5.0/22
   gateway-address = 10.4.5.7
   metric = 0
   cost = 1
   preference = 100
   third-party = no
   ase-type = type-1
   ase-tag = c0:00:00:00
   private-route = yes
   active-route = yes
   ase7-adv = N/A
```

Following is an example that configures the MAX TNT to route in an NSSA and import a type-7 LSA that specifies an external route across the WAN link. The example shows the commands entered to configure profiles, and the system’s responses.

1 Configure each IP interface that is running OSPF with an NSSA area type. The MAX TNT does not operate as an ABR, so the area-type as well as the area number must be the same on all interfaces running OSPF:

```
admin> read ip-int {{ 1 c 1 } 0 }
IP-INTERFACE/{ { shelf-1 controller 1 } 0 } read
admin> set ospf area-type = NSSA
admin> write
IP-INTERFACE/{ { shelf-1 controller 1 } 0 } written
```

2 Configure the WAN link that represents an AS-external route. For example:

```
admin> read connection ase-like
CONNECTION/ase-link read
admin> set ip-options remote = 10.4.5.7/22
admin> set ip-options rip = routing-off
```

```
admin> set ip-options ospf active = yes

admin> write
CONNECTION/ase-link written
```

3 Configure a static route to the remote site. For example:

```
admin> new ip-route type7
IP-ROUTE/type7 read

admin> set dest = 10.4.5.0/22

admin> set gateway = 10.4.5.7

admin> set ase7-adv = Advertise

admin> write
IP-ROUTE/type7 written
```

In example route shown above, the ase7-adv parameter specifies that area border routers should convert this type-7 to a type-5 LSA. Note that third-party routing cannot be specified when ASE type-7s are advertised (as specified in RFC 1587).

Advertising pool addresses

When the MAX TNT is operating as an OSPF router and is configured with network summarized pools of addresses for dynamic assignment to hosts, you can specify how to import the pool addresses into OSPF: as external Type-1 routes, external Type-2 routes, or intra-area routes.

For information about specifying pool addresses, see “Configuring address pools for dynamic assignment to dial-in hosts” on page 4-17 of Chapter 4, “IP Router Configuration.”

OSPF information in static routes

When the MAX TNT starts up, it builds the routing table initially using its know static routes, which include those defined in IP-Interface profiles, Connection profiles, and IP-Route profiles. The routes in IP-Route profiles are also passed to the router whenever a route changes. These are the related parameters, shown with sample settings:

```
IP-ROUTE name
   name* = default
   dest-address = 0.0.0.0/0
   gateway-address = 10.2.3.17
   metric =1
   cost =1
   preference = 100
   third-party = no
   ase-type = type-1
   ase-tag = c0:00:00:00
   private-route = yes
   active-route = yes
   ase7-adv
```

The following parameters apply only when OSPF is enabled:

```
IP-ROUTE name
   cost =1
   third-party = no
   ase-type = type-1
   ase-tag = c0:00:00:00
   ase7-adv = N/A
```

For information about the ase-type and ase-tag parameters, see “Handling routes learned from RIP” on page 5-11.

Assigning a cost to a static route

The lower the cost you assign to a route, the more likely the route is to be used to forward data traffic. Typically, you should reflect the bandwidth of a connection when assigning costs; for example, for a single B-channel connection, the cost would be 24 times greater than a T1 link.

The MAX TNT has a default cost of 1 for a connected route (Ethernet) and 10 for a WAN link. If you have two paths to the same destination, the one with the lower cost will be used. Be careful when assigning costs. Incorrect cost metrics can cause delays and congestion on the network.

Following is an example that assigns a cost of 25 to a static route:

```
admin> new ip-route 56klink
IP-ROUTE/56klink read
admin> set dest = 10.1.2.0/24
admin> set gateway = 10.9.8.10
admin> set cost = 25
admin> write
IP-ROUTE/56klink written
```

Specifying a third-party route

OSPF can advertise routes to external destinations on behalf of another gateway (a third party). This is commonly known as advertising a forwarding address. If third-party routing is disabled, the MAX TNT advertises itself as the forwarding address to an external destination. When it is enabled, it advertises the IP address of another gateway.

Depending on the exact topology of the network, it may be possible for other routers to use this type of LSA and route directly to the forwarding address without involving the advertising MAX TNT, increasing the total network throughput. This feature can be used only if all OSPF routers know how to route to the forwarding address. This usually means that the forwarding address is on a local network that has an OSPF router acting as the forwarding router, or that designated router is sending LSAs for that Ethernet to any area that sees the static route's forwarding address LSAs.

Following is an example that configures a route that enables the MAX TNT running OSPF to advertise a third-party router for a the 10.1.2.0 network, where the third-party router’s forwarding address is 10.9.8.10:

```
admin> new ip-route third-party
IP-ROUTE/third-party read
```

```
admin> set dest = 10.1.2.0/24
admin> set gateway = 10.9.8.10
admin> set third-party = yes
admin> write
IP-ROUTE/third-party written
```

Handling type-7 LSAs

The ase7-adv parameter specifies that area border routers should convert this type-7 to a type-5 LSA. Note that third-party routing cannot be specified when ASE type-7s are advertised (as specified in RFC 1587). For details, see “Example of an NSSA with a type-7 LSA” on page 5-16.

Multicast Forwarding

6

This chapter covers the following topics:

Introduction to multicast forwarding

The multicast backbone (MBONE) is a virtual network layered on top of the Internet to support IP multicast routing across point-to-point links. It is used for transmitting audio and video on the Internet in real-time, because multicasting is a much cheaper and faster way to communicate the same information to multiple hosts.

To the MBONE, the MAX TNT looks like a multicast client. It responds as a client to IGMP (Internet Group Membership Protocol) packets it receives from MBONE routers, which may be IGMP version-1 or version-2, including IGMP MTRACE (multicast trace) packets.

To multicast clients on a WAN or Ethernet interface, the MAX TNT looks like a multicast router. Like a router, it sends those clients IGMP queries, receives responses, and forwards multicast traffic. In this implementation, multicast clients are not allowed to source multicast packets—if they do, the MAX TNT discards the packets.

Following are the parameters related to configuring multicast forwarding, shown with their default values:

```
IP-GLOBAL
   multicast-forwarding = no
   mbone-profile = ""
   mbone-lan-interface = { { any-shelf any-slot 0 } 0 }
   multicast-hbeat-addr = 0.0.0.0
   multicast-hbeat-port = 0
   multicast-hbeat-slot-time = 0
   multicast-hbeat-Number-Slot = 0
   multicast-hbeat-Alarm-threshold = 0
   multicast-hbeat-src-addr = 0.0.0.0
   multicast-hbeat-src-addr-mask = 0.0.0.0
   multicast-member-timeout = 360
IP-INTERFACE {{shelf-N slot-N N } N }
   multicast-allowed = no
   multicast-rate-limit = 100
CONNECTION station
   ip-options...
      multicast-allowed = no
      multicast-rate-limit = 100
```

For information about each parameter, see the *MAX TNT Reference Guide*.

Enabling multicast-forwarding for the system

The multicast-forwarding parameter in the IP-Global profile turns on multicast forwarding in the MAX TNT. When you change the multicast-forwarding parameter from No to Yes, the multicast subsystem reads the values in the IP-Global profile and initiates the forwarding function.

Note: If you modify a multicast value in the IP-Global profile, you must set this parameter to No and then set it to Yes again to force a read of the new values.

Specifying a local or WAN MBONE interface

The MBONE interface is where the multicast router resides. If it is across the WAN, the mbone-profile parameter must specify the name of a local Connection profile to that router. If it is on a local Ethernet segment, the mbone-lan-interface parameter must specify the interface address of that segment.

Note: The mbone-profile and mbone-lan-interface parameters are mutually exclusive. You cannot configure the MAX TNT to respond to both a local and WAN multicast router.

Monitoring the multicast heartbeat

Heartbeat monitoring is optional. It is not required for multicast forwarding. When the MAX TNT is running as a multicast forwarder, it is continually receiving multicast traffic. The heartbeat-monitoring feature enables the administrator to monitor possible connectivity problems by continuously polling for this traffic and generating an SNMP alarm trap if there is a traffic breakdown. This is the SNMP alarm trap:

```
Trap type:  TRAP_ENTERPRISE
Code:       TRAP_MULTICAST_TREE_BROKEN (19)
Arguments:
1) Multicast group address being monitored (4 bytes),
2) Source address of last heartbeat packet received (4 bytes)
3) Slot time interval configured in seconds (4 bytes),
4) Number of slots configured (4 bytes).
5) Total number of heartbeat packets received before the unit started
sending SNMP Alarms (4 bytes).
```

To set up heartbeat monitoring, you configure several parameters that define what packets will be monitored, how often and for how long to poll for multicast packets, and the threshold for generating an alarm.

What packets will be monitored

The multicast-hbeat-addr parameter specifies a multicast address. If specified, the MAX TNT listens for packets to and from this group.

The multicast-hbeat-port parameter specifies a UDP port number. If specified, the MAX TNT listens only to packets received on that port.

The multicast-hbeat-src-addr and multicast-hbeat-src-addr-mask parameters specify an IP address and netmask. If specified, the MAX TNT ignores packets from that source for monitoring purposes.

How often and for how long to poll for multicast packets

The multicast-hbeat-slot-time parameter specifies an interval (in seconds). The MAX TNT polls for multicast traffic, waits for this interval, and then polls again.

The multicast-hbeat-Number-Slot parameter specifies how many times to poll before comparing the number of heartbeat packets received to the alarm-threshold.

The threshold for generating an alarm

The multicast-hbeat-Alarm-threshold parameter specifies a number. If the number of monitored packets falls below this number, the SNMP alarm trap is sent.

Enabling multicast forwarding on an interface

Each local or WAN interface that supports multicasting must be configured to allow multicasting (multicast-allowed). When multicast-allowed is set to Yes, the MAX TNT begins handling IGMP requests and responses on the interface. It does not begin forwarding multicast traffic until the rate limit is set.

The multicast-rate-limit specifies the rate at which the MAX TNT accepts multicast packets from its clients. It does not affect the MBONE interface.

Note: By default, the multicast-rate-limit parameter is set to 100. This disables multicast forwarding on the interface. The forwarder handles IGMP packets, but does not accept packets from clients or forward multicast packets from the MBONE router.

To begin forwarding multicast traffic on the interface, you must set the multicast-rate-limit parameter to a number less than 100. For example if you set it to 5, the MAX TNT accepts a packet from multicast clients on the interface every 5 seconds. Any subsequent packets received in that 5-second window are discarded.

For high-bandwidth data, voice, and audio multicast applications, the MAX TNT supports both multicast rate limiting (described immediately above) and prioritized packet dropping. If the MAX TNT is the receiving device under extremely high loads, it drops packets according to a priority ranking, which is determined by the following UDP port ranges:

- Traffic on ports 0–16384 (unclassified traffic) has the lowest priority (50).
- Traffic on ports 16385–32768 (Audio traffic) has the highest priority (70).
- Traffic on ports 32769–49152 (Whiteboard traffic) has medium priority (60).
- Traffic on ports 49153–65536 (Video traffic) has low priority (55).

Specifying a multicast group membership timeout

When the MAX TNT is configured as a multicast forwarder, it forwards polling messages generated by the multicast router and keeps track of active memberships from its client interfaces. In previous releases, if no client responded to the polling messages within six minutes, the MAX TNT stopped forwarding multicast traffic on that interface. In this release, you can configure the timeout value by specifying a value between 60 seconds and 65535 seconds. The factory default is still six minutes. Following is an example that configures the timeout value to 60 seconds:

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

admin> set multicast-member-timeout = 60

admin> write
IP-GLOBAL written
```

Configuring the MAX TNT as a multicast forwarder

This section shows how to configure the MAX TNT as a multicast forwarder receiving multicast packets from a local Ethernet interface, or from a WAN link.

Example of forwarding from an MBONE router on a LAN interface

Figure 6-1 shows a local multicast router on one of the MAX TNT unit's Ethernet interfaces and WAN multicast clients.

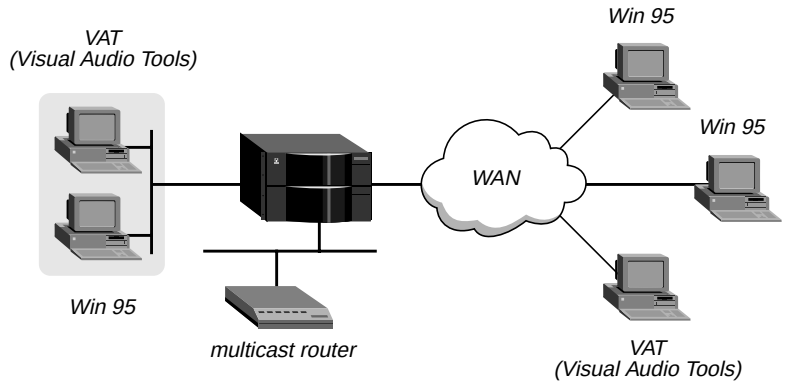


Figure 6-1. Forwarding multicast traffic to dial-in multicast clients

To configure this example setup, first open IP-Global profile and set up multicast forwarding and heartbeat monitoring. (Heartbeat monitoring is an optional feature. You can operate multicast forwarding without it if you prefer.)

Following is an example that specifies the MBONE interface as the second Ethernet port of a card installed in the unit's slot #6, and uses the heartbeat group address of 224.1.1.1:

```
admin> read ip-global
IP-GLOBAL/ read
admin> set multicast-forwarding = yes
admin> set mbone-lan-interface = { {shelf-1 slot-6 2} 0 }
admin> set multicast-hbeat-addr = 224.1.1.1
admin> set multicast-hbeat-port = 16387
admin> set multicast-hbeat-slot-time = 10
admin> set multicast-hbeat-Number-Slot = 10
admin> set multicast-hbeat-Alarm-threshold = 3
admin> write
IP-GLOBAL/ written
```

The next step is to enable multicasting on the MBONE IP interface, as shown in the following example:

```
admin> read ip-interface {{1 6 2} 0}
IP-INTERFACE/{ { shelf-1 slot-6 2 } 0 } read
admin> set multicast-allowed = yes
```

```
admin> write
IP-INTERFACE/{ { shelf-1 slot-6 2 } 0 } written
```

If other local IP interfaces support multicast clients, you should enable multicasting and configure the rate limit in those IP-Interface profiles. (The MBONE interface cannot support clients.)

Finally, enable multicasting on each WAN interface that supports multicast clients. Note that you must also set the rate limit to a number lower than 100 to enable the MAX TNT to forward multicast traffic. For example:

```
admin> read connection vatclient
CONNECTION/vatclient read
admin> set ip multicast-allowed = yes
admin> set ip multicast-rate-limit = 5
admin> write
CONNECTION/vatclient written
```

Example of forwarding from an MBONE router on a WAN link

Figure 6-2 shows a multicast router on a WAN interface with both local and WAN multicast clients.

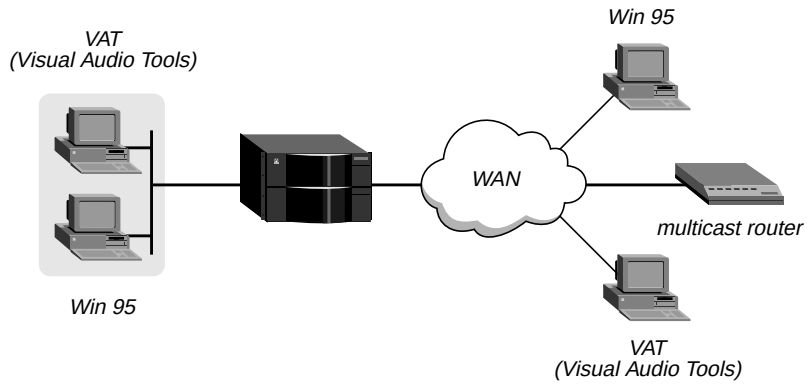


Figure 6-2. Forwarding multicast traffic on both Ethernet and WAN interfaces

To configure this example setup, first open IP-Global profile and set up multicast forwarding. The following example does not use heartbeat monitoring. If you want to configure the MAX TNT for heartbeat monitoring, see the example settings in “Example of forwarding from an MBONE router on a LAN interface” on page 6-5.

Following is an example that specifies the MBONE interface as a WAN link accessed through a Connection profile named Mbone:

```
admin> read ip-global
IP-GLOBAL/ read
admin> set multicast-forwarding = yes
admin> set mbone-profile = Mbone
admin> write
IP-GLOBAL/ written
```

The next step is to enable multicasting on the local IP interface that supports multicast clients. Note that you must also set the rate-limit to a number lower than 100, as shown in the following example:

```
admin> read ip-interface {{1 6 1} 0}
IP-INTERFACE/{ { shelf-1 slot-6 1 } 0 } read

admin> set multicast-allowed = yes

admin> set multicast-rate-limit = 5

admin> write
IP-INTERFACE/{ { shelf-1 slot-6 1 } 0 } written
```

Then, enable multicasting on the MBONE interface, as shown in the next set of commands:

```
admin> read connection Mbone
CONNECTION/Mbone read

admin> set ip multicast-allowed = yes

admin> write
CONNECTION/Mbone written
```

Finally, enable multicasting on each WAN interface that supports multicast clients. Note that you must also set the rate-limit to a number lower than 100. For example:

```
admin> read connection vatclient
CONNECTION/vatclient read

admin> set ip multicast-allowed = yes

admin> set ip multicast-rate-limit = 5

admin> write
CONNECTION/vatclient written
```

Packet and Route Filters

7

This chapter covers the following topics:

Introduction

A packet filter contains rules describing packets and what to do when those packets are encountered. When a packet filter is applied to an interface, the MAX TNT monitors the data stream on that interface and takes a specified action when packet contents match the filter rules. Depending on how the filter is defined, it may apply to inbound or outbound packets, or both. In addition, filter rules are flexible enough to take an action (such as forward or drop) on those packets that match the rules, or all packets *except* those that match the rules.

A route filter contains rules that specify actions on routes in RIP update packets. When a route filter is applied to an IP interface, the MAX TNT monitors RIP packets on that interface and takes a specified action when route matches the filter rules. Depending on how the filter is defined, it may apply to inbound or outbound RIP packets, or both. Route filters are supported only in Filter profiles defined locally in the command-line interface, not in filters defined in RADIUS.

Creating and applying packet filters

This section explains how packet filters work and how to define them in Filter profiles. It also describes how to apply a packet filter to a local or WAN interface.

Basic types of packet filters

The basic types of packet filters are Generic filters or IP filters.

Generic filters examine the byte- or bit-level contents of any packet. They focus on certain bytes or bits in a packet and compare the contents of that location with a value defined in the filter. To use generic filters effectively, you need to know the contents of certain bytes in the packets you wish to filter. Protocol specifications are usually the best source of such information.

IP filters examine higher-level fields specific to IP packets. IP filters focus on known fields in IP packets, such as source or destination address, protocol number, and so forth. They operate on logical information, which is relatively easy to obtain.

Basic applications of packet filters

After you have defined a packet filter, you apply it to an interface to monitor packets crossing that interface. You can apply the filter as one of the following:

- A data filter, to define which packets can or cannot cross the interface
- A call filter, to define which packets can or cannot bring up a connection or reset the idle-timer for an established connection (WAN interfaces only)

Packets can pass through both a data filter and call filter on a WAN interface. If both a data and call filter are applied, the data filter is applied first.

Data filters for dropping or forwarding certain packets

Data filters are commonly used for security, but they can apply to any purpose that requires the MAX TNT to drop or forward only specific packets. For example, you can use data filters to drop packets addressed to particular hosts or to prevent broadcasts from going across the WAN. You can also use data filters to allow users to access only specific devices across the WAN.

When you apply a data filter, its forwarding action (forward or drop) affects the actual data stream by preventing certain packets from reaching the Ethernet from the WAN, or vice versa. Data filters do not affect the idle timer, and a data filter applied to a Connection profile does not affect the answering process.

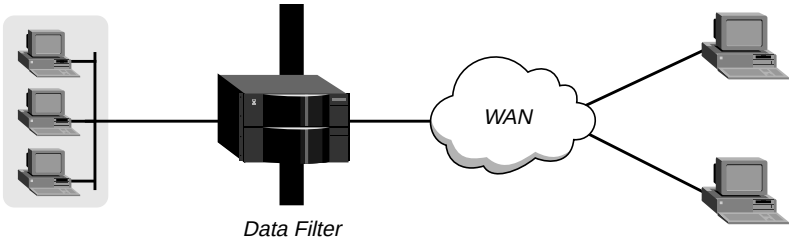


Figure 7-1. Data filters can drop or forward certain packets

Call filters for managing connections

Call filters prevent unnecessary connections and help the MAX TNT distinguish active traffic from “noise.” By default, any traffic to a remote site triggers a call, and any traffic across an active connection resets the connection’s idle timer.

When you apply a call filter, its forwarding action (forward or drop) does not affect which packets are sent across an active connection. The forwarding action of a call filter determines which packets can either initiate a connection or reset a session’s timer. When a session’s idle-timer expires, the session is terminated. The idle-timer is set to 120 seconds by default, so if a connection is inactive for two minutes, the MAX TNT terminates the connection.

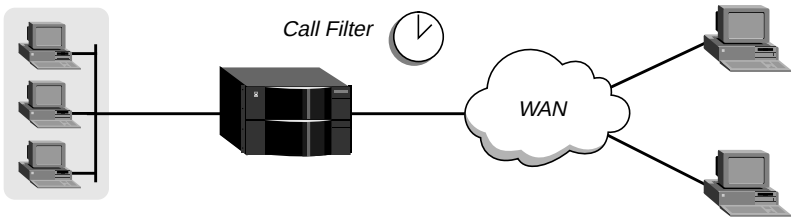


Figure 7-2. Call filters can prevent certain packets from resetting the timer

How packet filters work

A Filter profile can contain up to 12 input and output filter specifications (rules). Each rule has its own forwarding action—forward or drop. A match occurs at the first successful comparison between a filter and the packet being examined. When a comparison succeeds, the filtering process stops and the forward action in that rule is applied to the packet.

If no comparisons succeed, the packet does not match this filter. However, this does not mean that the packet is forwarded. When no filter is in use, the MAX TNT forwards all packets, but

once you apply a filter to an interface, this default is *reversed*. For security purposes, the unit does not automatically forward non-matching packets. It requires a rule that explicitly allows those packets to pass. For an example of an input filter that forwards all packets that did not match a previous rule, see "Example of a filter to prevent IP address spoofing" on page 7-11.

Note: For a call filter to prevent an interface from remaining active unnecessarily, you must define rules for both input and output packets. Otherwise, if only input rules are defined, output packets will keep a connection active, or vice versa.

In a generic filter, all parameter settings in a rule work together to specify a location in a packet and a number to be compared to that location. The Comp-Neq parameter specifies whether a comparison succeeds when the contents of the packet equal or do not equal that number.

In an IP filter, a set of distinct comparisons are made in a defined order. When a comparison fails, the packet is allowed to go on to the next comparison. When a comparison succeeds, the filtering process stops and the forward action in that rule is applied to the packet. The IP filter tests proceed in the following order:

1. Compare source address parameters to the source address of the packet. If they are not equal, the comparison fails.
2. Compare destination address parameters to the destination address in the packet. If they are not equal, the comparison fails.
3. If the protocol parameter is zero (which matches any protocol), the comparison succeeds. If it is non-zero and not equal to the protocol field in the packet, the comparison fails.
4. If the Src-Port-Cmp parameter is not set to none, compare the source port parameter to the source port of the packet. If they do not match as specified in the Src-Port-Cmp parameter, the comparison fails.
5. If the Dst-Port-Cmp parameter is not set to none, compare the destination port parameter to the destination port of the packet. If they do not match as specified in the Dst-Port-Cmp parameter, the comparison fails.
6. If TCP-Estab is Yes and the protocol number is 6, the comparison succeeds.

Working with Filter profiles

Filter profiles contain parameters that set the rules describing packets and what to do when those packets are encountered. Following are the parameters in a Filter profile, shown with their default settings:

```
FILTER filter-name
   filter-name* = filter-name
   input-filters
      input-filters[1]—input-filters[12]
         valid-entry = no
         forward = no
         Type = generic-filter
         gen-filter
            offset = 14
            len = 8
            more = no
            comp-neq = no
            mask = ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff
            value = aa:aa:03:00:00:00:80:f3:00:00:00:00
```

```
            Type = ip-filter
            ip-filter
               protocol = 0
               source-address-mask = 255.255.255.192
               source-address = 192.100.50.128
               dest-address-mask = 0.0.0.0
               dest-address = 0.0.0.0
               Src-Port-Cmp = none
               source-port = none
               Dst-Port-Cmp = none
               dest-port = none
               tcp-estab = no
         output-filters
            output-filters[1]—output-filters[12]
               valid-entry = no
               forward = no
               Type = generic-filter
               gen-filter
                  offset = 14
                  len = 8
                  more = no
                  comp-neq = no
                  mask = ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff
                  value = aa:aa:03:00:00:00:80:f3:00:00:00:00
               Type = ip-filter
               ip-filter
                  protocol = 0
                  source-address-mask = 255.255.255.192
                  source-address = 192.100.50.128
                  dest-address-mask = 0.0.0.0
                  dest-address = 0.0.0.0
                  Src-Port-Cmp = none
                  source-port = none
                  Dst-Port-Cmp = none
                  dest-port = none
                  tcp-estab = no
```

For information about each parameter, see the *MAX TNT Reference Guide*. Table 7-1 shows the top-level elements in a Filter profile. See “How packet filters work” on page 7-3 for related information.

Table 7-1. Basic elements of a Filter profile

Element	Usage
Filter-name	Each filter must be assigned a name so it can be applied by name to an interface. The name you assign becomes the Filter profile’s index.
Input and output-filters	Each filter can contain up to 12 input-filters and 12 output-filters, which are defined individually and applied in order (1–12) to the packet stream. Input-filters are applied to inbound packets. Output-filters are applied to outbound packets.

Table 7-1. Basic elements of a Filter profile (continued)

Element	Usage
Valid-Entry	The Valid-Entry parameter enables or disables the current input or output filter. When it is set to No (the default), that input or output filter is skipped when filtering the data stream. Set it to Yes for each defined filter you intend to use.
Forward	Forward specifies whether the MAX TNT discards or forwards packets that match the filter specification. For a call filter, the forward action is to bring up a connection or reset the timer. When no filters are in use, the MAX TNT forwards all packets by default. When a filter is in use, the default is to discard matching packets (forward = no).
Type	Type can be set to generic-filter or ip-filter. Only the parameters in the corresponding subprofile (gen-filter or ip-filter) are applicable.

Generic filter rules

Generic filters can affect any packet, regardless of its protocol type or header fields. They use the following parameters:

```
gen-filter
   offset = 14
   len = 8
   more = no
   comp-neq = no
   mask = ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff
   value = aa:aa:03:00:00:00:80:f3:00:00:00:00
```

The following subsections provide background information about how these parameters work.

Specifying the offset to the bytes to be examined

The Offset specifies a byte-offset from the start of a frame to the data in the packet to be tested against this filter. For example, with the following filter specification:

```
gen-filter
   offset = 2
   len = 8
   more = no
   comp-neq = no
   mask = 0f:ff:ff:ff:00:00:00:f0:00:00:00:00
   value = 07:fe:45:70:00:00:00:90:00:00:00:00
```

and the following packet contents:

```
2A 31 97 FE 45 70 12 22 33 99 B4 80 75
```

The first two byes in the packet (2A and 31) are ignored due to the two-byte offset.

Specifying the number of bytes to test

The Len parameter specifies the number of bytes to test in a frame, starting at the specified Offset. The MAX TNT compares the contents of those bytes to the value specified in the filter's value parameter. For example, with the following filter specification:

```
gen-filter
   offset = 2
   len = 8
   more = no
   comp-neq = no
   mask = 0f:ff:ff:ff:00:00:00:f0:00:00:00:00
   value = 07:fe:45:70:00:00:00:90:00:00:00:00
```

and the following packet contents:

```
2A 31 97 FE 45 70 12 22 33 99 B4 80 75
```

The filter applies the mask only to the eight bytes following the two-byte offset.

Linking the filter to the next input- or output-filter in sequence

The More parameter specifies whether the MAX TNT includes the next filter condition before determining whether the frame matches the filter. If checked, the current filter condition is linked to the one immediately following it, so the filter can examine multiple non-contiguous bytes within a packet. In effect, this parameter "marries" the current filter to the next one, so that the next filter is applied before the forwarding decision is made. The match occurs only if *both* non-contiguous bytes contain the specified values.

The next filter must be enabled; otherwise, the MAX TNT ignores the filter.

Type of comparison to be performed when matching the packet

The Comp-Neq (compare-not-equals) parameter specifies the type of comparison to make between the specified value and the packet's contents: equal or not equal.

Masking the value before comparison

The Mask is a 12-byte mask to apply to the specified value parameter before comparing it to the packet contents at the specified offset. You can use it to fine-tune exactly which bits you want to compare. It is assumed to the same number of octets as the len parameter.

The MAX TNT applies the mask to the specified value using a logical AND after the mask and value are both translated into binary format. The mask hides the bits that appear behind each binary 0 (zero) in the mask. A mask of all ones (FF:FF:FF:FF:FF:FF:FF:FF) masks no bits, so the full specified value must match the packet contents. For example, with the following filter specification:

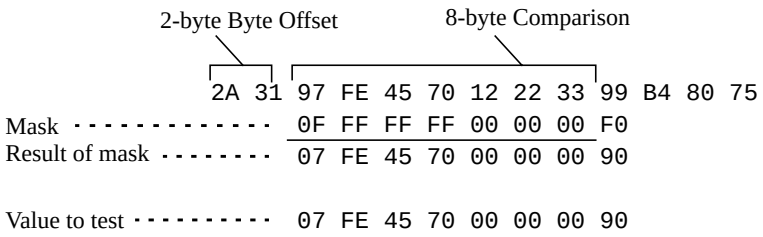
```
gen-filter
   offset = 2
   len = 8
   more = no
   comp-neq = no
```

```
mask = 0f:ff:ff:ff:00:00:00:f0:00:00:00:00
value = 07:fe:45:70:00:00:00:90:00:00:00:00
```

and the following packet contents:

```
2A 31 97 FE 45 70 12 22 33 99 B4 80 75
```

The mask is applied as shown below, resulting in a value that matches the Value.



The packet matches this filter. Because the forward parameter is set to No, the packet will be dropped. The byte comparison works as follows:

- 2A and 31 are ignored due to the two-byte offset.
- 9 in the lower half of the third byte is ignored, because the mask has a 0 in its place. The 7 in the third byte matches the value parameter's 7 in the upper half of that byte.
- F and E in the fourth byte match the value parameter for that byte.
- 4 and 5 in the fifth byte match the value parameter for that byte.
- 7 and 0 in the sixth byte match the value parameter for that byte.
- 12 and 22 and 33 in the seventh, eighth and ninth bytes are ignored because the mask has a 0 in those places.
- 9 in the tenth byte equals the matches the value parameter's 9 in the lower half of that byte. The second 9 in the upper-half of the packet's tenth byte is ignored because the mask has a 0 in its place.

The value to match up in the packet contents

The Value parameter specifies a hexadecimal number to be compared to specific bits contained in packets after the Offset, Length, and Mask calculations have been applied.

IP filter rules

IP filter rules affect only IP and related packets. IP filters use the following parameters:

```
ip-filter
   protocol = 0
   source-address-mask = 255.255.255.192
   source-address = 192.100.50.128
   dest-address-mask = 0.0.0.0
   dest-address = 0.0.0.0
   Src-Port-Cmp = none
   source-port = 0
   Dst-Port-Cmp = none
```

```
dest-port = 0
tcp-estab = no
```

The following subsections provide background information about how these parameters work.

Filtering on the protocol number field in IP packets

A protocol number of zero matches all protocols. If you specify a non-zero number, the MAX TNT compares it to the protocol number field in packets. Common protocols are listed below, but protocol numbers are not limited to this list. For a complete list, see the section on Well-Known Port Numbers in RFC 1700, *Assigned Numbers*, by Reynolds, J. and Postel, J., October 1994.

- 1: ICMP
- 5: STREAM
- 8: EGP
- 6: TCP
- 9: Any private interior gateway protocol (such as Cisco's IGRP)
- 11: Network Voice Protocol
- 17: UDP
- 20: Host Monitoring Protocol
- 22: XNS IDP
- 27: Reliable Data Protocol
- 28: Internet Reliable Transport Protocol
- 29: ISO Transport Protocol Class 4
- 30: Bulk Data Transfer Protocol
- 61: Any Host Internal Protocol
- 89: OSPF

Filtering by source or destination address

To filter a packet based on its source IP address, use the source-address-mask and source-address parameters. To filter on destination address, use the dest-address-mask and dest-address parameters.

If you specify an address mask, it is applied to the corresponding address before comparing it to the source or destination address in a packet. You can use an address mask to mask out the host portion of an address, for example, or the host and subnet portion.

The MAX TNT applies the mask to the address using a logical AND after the mask and address are both translated into binary format. The mask hides the portion of the address that appears behind each binary 0 (zero) in the mask. A mask of all zeros (the default) masks all bits; if the address parameter is all zeros, all addresses are matched. A mask of all ones (255.255.255.255) masks no bits, so the full source or destination address for a single host is matched.

Filtering by port numbers

The source-port and dest-port parameters specify a port number to be compared with the source and destination ports in a packet. Port 25 is reserved for SMTP; that socket is dedicated to receiving mail messages. Port 20 is reserved for FTP data messages, port 21 for FTP control sessions, and port 23 for telnet. A port number of zero matches nothing. To bypass a comparison of port numbers, set the comparison method (Src-Port-Cmp or Dst-Port-Cmp) to None.

The Src-Port-Cmp and Dst-Port-Cmp parameters specify the type of comparison to be made on the corresponding port addresses: none, less (match if the packet's port number is less than the specified one), eql, gtr, or neq (not-equal).

Filtering only established TCP sessions.

Tcp-Estab can be used to restrict the filter to packets in an established TCP session. You can only use it if the Protocol number has been set to 6 (TCP).

Example of a general call filter

The following example shows how to define a call filter. The filter's purpose is to prevent inbound packet from resetting the session-timer. It does not prevent any outbound packets from resetting the timer or placing a call. The example shows the commands entered to define the filter, and the system's responses:

1 Create a Filter profile:

```
admin> new filter out-only
FILTER/out-only read
```

2 List input-filter 1 and activate it. Leave all other parameters set to default values:

```
admin> list input 1
valid-entry = no
forward = no
Type = generic-filter
gen-filter = { 0 0 no no 00:00:00:00:00:00:00:00:00:00:00:00 00:00:00:0+
ip-filter = { 0 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 none 0 none 0 no }
admin> set valid = yes
```

3 List output-filter 1, activate it, and set forward to Yes. Leave all other parameters set to default values:

```
admin> list .. .. output 1
valid-entry = no
forward = no
Type = generic-filter
gen-filter = { 0 0 no no 00:00:00:00:00:00:00:00:00:00:00:00 00:00:00:0+
ip-filter = { 0 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 none 0 none 0 no }
admin> set valid = yes
admin> set forward = yes
```

4 Write the Filter profile:

```
admin> write
FILTER/out-only written
```

Example of a filter to prevent IP address spoofing

IP address spoofing occurs when a remote device illegally acquires a local address to break through a firewall or data filter. In the following example, the filter first defines input filters that drop packets whose source address is on the local IP network or the loopback address (127.0.0.0). In effect, the rules in this filter say: "If you see an inbound packet with one of these source addresses, drop the packet." The third input filter defines every other source address (0.0.0.0) and specifies "Forward everything else to the local network."

The example filter uses a local IP network address of 192.100.50.128, with a subnet mask of 255.255.255.192. These addresses are just examples.

Note: If you apply this filter to the Ethernet interface, the MAX TNT drops IP packets it receives from local LAN and you will not be able to Telnet to the unit.

The following example shows the commands entered to define the filter, and the system's responses:

1 Create a Filter profile:

```
admin> new filter ip-spoof
FILTER/ip-spoof read
```

2 List input-filter 1, activate it, and set the type to ip-filter:

```
admin> list input 1
valid-entry = no
forward = no
Type = generic-filter
gen-filter = { 0 0 no no 00:00:00:00:00:00:00:00:00:00:00:00 00:00:00:+
ip-filter = { 0 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 none 0 none 0 no }

admin> set valid = yes

admin> set type = ip-filter
```

3 List the ip-filter subprofile, and specify the source netmask and address for the local network. Because forward is set to No, if an incoming packet has the local address, the MAX TNT does not forward it onto the Ethernet. Use Set commands to specify the following values:

```
protocol = 0
source-address-mask = 255.255.255.192
source-address = 192.100.50.128
dest-address-mask = 0.0.0.0
dest-address = 0.0.0.0
Src-Port-Cmp = none
source-port = 0
Dst-Port-Cmp = none
dest-port = 0
tcp-estab = no
```

4 List input-filter 2, activate it, and set the type to ip-filter:

```
admin> list .. .. 2
valid-entry = no
forward = no
Type = generic-filter
```

```
gen-filter = { 0 0 no no 00:00:00:00:00:00:00:00:00:00:00:00 00:00:00:0+
ip-filter = { 0 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 none 0 none 0 no }
```

```
admin> set valid = yes
```

```
admin> set type = ip-filter
```

5 List the ip-filter subprofile, and specify the loopback source address. Because forward is set to No, if an incoming packet has this address, the MAX TNT does not forward it onto the Ethernet. Use Set commands to specify the following values:

```
protocol = 0
source-address-mask = 255.0.0.0
source-address = 127.0.0.0
dest-address-mask = 0.0.0.0
dest-address = 0.0.0.0
Src-Port-Cmp = none
source-port = 0
Dst-Port-Cmp = none
dest-port = 0
tcp-estab = no
```

6 List input-filter 3, activate it, set forward to yes, and set the type to ip-filter:

```
admin> list .. .. 3
valid-entry = no
forward = no
Type = generic-filter
gen-filter = { 0 0 no no 00:00:00:00:00:00:00:00:00:00:00:00 00:00:00:0+
ip-filter = { 0 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 none 0 none 0 no }
```

```
admin> set valid = yes
```

```
admin> set forward = yes
```

```
admin> set type = ip-filter
```

7 Leave all default values in the ip-filter subprofile for input-filter 3. Because forward is set to Yes, this specifies that all other (non-local) incoming source addresses are acceptable. Use Set commands to specify the following values:

```
protocol = 0
source-address-mask = 0.0.0.0
source-address = 0.0.0.0
dest-address-mask = 0.0.0.0
dest-address = 0.0.0.0
Src-Port-Cmp = none
source-port = 0
Dst-Port-Cmp = none
dest-port = 0
tcp-estab = no
```

8 List output-filter 1, activate it, set Forward to Yes, and set the type to ip-filter:

```
admin> list .. .. .. output 1
valid-entry = no
forward = no
Type = generic-filter
gen-filter = { 0 0 no no 00:00:00:00:00:00:00:00:00:00:00:00 00:00:00:0+
ip-filter = { 0 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 none 0 none 0 no }
```

```
admin> set valid = yes
```

```
admin> set forward = yes

admin> set type = ip-filter
```

9 List the ip-filter subprofile, and specify the source netmask and address for the local network. (Packets originating on the local network should be forwarded across the WAN.) Use Set commands to specify the following values:

```
protocol = 0
source-address-mask = 255.255.255.192
source-address = 192.100.40.128
dest-address-mask = 0.0.0.0
dest-address = 0.0.0.0
Src-Port-Cmp = none
source-port = 0
Dst-Port-Cmp = none
dest-port = 0
tcp-estab = no
```

10 Write the Filter profile:

```
admin> write
FILTER/ip-spoof written
```

Example of a filter for more complex IP security issues

This section illustrates some of the issues you may need to consider when writing your own IP filters. However, the sample filter presented here does not address the fine points of network security. You may want to use this filter as a starting point and augment it to address your security requirements.

In the following example, the local network supports a Web server and the administrator needs to carry out the following tasks:

- Provide dial-in access to the server's IP address.
- Restrict dial-in traffic to all other hosts on the local network.

However, many local IP hosts need to dial out to the Internet and use IP-based applications such as Telnet or FTP, so their response packets need to be directed appropriately to the originating host. In this example, the Web server's IP address is 192.9.250.5. This filter will be applied in Connection profiles as a data filter.

The following example shows the commands entered to define the filter, and the system's responses:

1 Create a Filter profile:

```
admin> new filter web-safe
FILTER/web-safe read
```

2 List input-filter 1, activate it, set forward to Yes and Type to ip-filter:

```
admin> list input 1
valid-entry = no
forward = no
Type = generic-filter
gen-filter = { 0 0 no no 00:00:00:00:00:00:00:00:00:00:00:00 00:00:00:0+
ip-filter = { 0 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 none 0 none 0 no }
```

```
admin> set valid = yes

admin> set forward = yes

admin> set type = ip-filter
```

3 To allow packets to reach the Web server's address at a destination TCP port which may be used for Telnet or FTP, list the ip-filter subprofile and use Set commands to specify the following values:

```
protocol = 6
source-address-mask = 0.0.0.0
source-address = 0.0.0.0
dest-address-mask = 255.255.255.255
dest-address = 192.9.250.5
Src-Port-Cmp = none
source-port = 0
Dst-Port-Cmp = eql
dest-port = 80
tcp-estab = no
```

4 List input-filter 2, activate it, set forward to Yes type to ip-filter:

```
admin> list .. .. 2
valid-entry = no
forward = no
Type = generic-filter
gen-filter = { 0 0 no no 00:00:00:00:00:00:00:00:00:00:00:00 00:00:00:+
ip-filter = { 0 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 none 0 none 0 no }

admin> set valid = yes

admin> set forward = yes

admin> set type = ip-filter
```

5 To allow inbound TCP packets that are responding to a local user's outbound Telnet request, you can forward TCP packets whose destination port is greater than the source port. (Telnet requests go out on port 23 and responses come back on some random port greater than port 1023.) To define the filter, list the ip-filter subprofile and use Set commands to specify the following values:

```
protocol = 6
source-address-mask = 0.0.0.0
source-address = 0.0.0.0
dest-address-mask = 0.0.0.0
dest-address = 0.0.0.0
Src-Port-Cmp = none
source-port = 0
Dst-Port-Cmp = gtr
dest-port = 1023
tcp-estab = no
```

6 List input-filter 3, activate it, set forward to Yes type to ip-filter:

```
admin> list .. .. 3
valid-entry = no
forward = no
Type = generic-filter
gen-filter = { 0 0 no no 00:00:00:00:00:00:00:00:00:00:00:00 00:00:00:+
ip-filter = { 0 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 none 0 none 0 no }
```

```
admin> set valid = yes

admin> set forward = yes

admin> set type = ip-filter
```

7 To allow inbound RIP updates, you can specify a filter that forwards inbound UDP packets if the destination port is greater than the source port. (For example, suppose a RIP packet goes out as a UDP packet to destination port 520. The response to this request goes to a random destination port greater than 1023.) To create this filter, list the ip-filter subprofile and use Set commands to specify the following values:

```
protocol = 17
source-address-mask = 0.0.0.0
source-address = 0.0.0.0
dest-address-mask = 0.0.0.0
dest-address = 0.0.0.0
Src-Port-Cmp = none
source-port = 0
Dst-Port-Cmp = gtr
dest-port = 1023
tcp-estab = no
```

8 List input-filter 4, activate it, set forward to Yes type to ip-filter:

```
admin> list .. .. 4
valid-entry = no
forward = no
Type = generic-filter
gen-filter = { 0 0 no no 00:00:00:00:00:00:00:00:00:00:00:00 00:00:00:+
ip-filter = { 0 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 none 0 none 0 no }

admin> set valid = yes

admin> set forward = yes

admin> set type = ip-filter
```

9 Leave all default values in the ip-filter subprofile for input-filter 4. This allows unrestricted pings and traceroutes. ICMP does not use ports like TCP and UDP, so a port comparison is unnecessary. Use Set commands to specify the following values:

```
protocol = 0
source-address-mask = 0.0.0.0
source-address = 0.0.0.0
dest-address-mask = 0.0.0.0
dest-address = 0.0.0.0
Src-Port-Cmp = none
source-port = 0
Dst-Port-Cmp = none
dest-port = 0
tcp-estab = no
```

10 Write the Filter profile:

```
admin> write
FILTER/web-safe written
```

Applying a packet filter to an interface

When you apply a packet filter to an interface, it affects the data stream or session management on the interface. Following are the parameters related to applying a packet filter, shown with their default settings:

```
ANSWER-DEFAULTS
   session-info
      call-filter = ""
      data-filter = ""
      filter-persistence = no
CONNECTION station
   session-options
      call-filter = ""
      data-filter = ""
      filter-persistence = no
ETHERNET {shelf-N slot-N N }
   filter-name= ""
```

For information about each parameter, see the *MAX TNT Reference Guide*. For information about call filters and data filters, see “Basic applications of packet filters” on page 7-2.

How the Answer-Defaults profile settings are used

If the MAX TNT uses a local Connection profile for authentication, it does not use packet filters set in the Answer-Defaults session-info subprofile.

If the MAX TNT relies on RADIUS for authentication and the caller’s RADIUS profile applies a data or call filter (or both), those filters are applied to each incoming packet and the MAX TNT does not use filters set in the Answer-Defaults session-info subprofile.

If the MAX TNT relies on RADIUS for authentication, the caller’s RADIUS profile does not apply a data or call filter, and the use-answer-for-all-defaults parameter is set to Yes in the Answer-Default profile, filters set in the Answer-Defaults session-info subprofile are applied to each incoming packet.

How filter persistence affects packet filters

Filter persistence is needed to allow Secure Access Firewalls to persist across connection state changes, but it is not needed for packet filters. If you do set it for a packet filter, the filter persists across connection state changes. For more information about persistence, see Appendix C, “Secure Access Firewalls.”

Applying a packet filter to a WAN interface

You can apply a filter to a WAN interface by specifying its name as a call-filter or data-filter in the Connection session-options subprofile. If you are not sure about data and call filters, see “Basic applications of packet filters” on page 7-2.

When you apply a filter to a WAN interface, it takes effect when the connection is brought up. If both a data filter and call filter are applied, the data filter is applied first. This means that only those packets that pass the data filter reach the call filter.

Following is an example that applies a data filter in a Connection profile:

1 Open a Connection profile and list its session-options subprofile:

```
admin> read conn tlynch
CONNECTION/tlynch read

admin> list session
call-filter = ""
data-filter = ""
filter-persistence = no
idle-timer = 120
ts-idle-mode = no-idle
ts-idle-timer = 120
```

2 Specify the filter's name in the data-filter parameter:

```
admin> set data-filter = ip-spoof
```

3 Write the Connection profile:

```
admin> write
CONNECTION/tlynch written
```

Following is an example that applies a call filter and sets the idle timer to 20 seconds. If no packets get through the call filter for 20 seconds, the connection is torn down. The example shows the command entered, and the system's responses:

1 Open a Connection profile and list its session-options subprofile:

```
admin> read conn bob
CONNECTION/bob read

admin> list session
call-filter = ""
data-filter = ""
filter-persistence = no
idle-timer = 120
ts-idle-mode = no-idle
ts-idle-timer = 120
```

2 Specify the filter's name in the call-filter parameter and reset the idle-timer to 20 seconds:

```
admin> set call-filter = out-only

admin> set idle-timer = 20
```

3 Write the Connection profile:

```
admin> write
CONNECTION/bob written
```

Applying a data filter to a LAN interface

Ethernet interfaces are connected routes, so call filters are not applicable. However, you can apply a data filter that affects which packets are allowed to reach the Ethernet or leave the Ethernet for another interface. A filter applied to an Ethernet interface takes effect immediately. If you change the Filter profile definition, the changes apply as soon as you save the Filter profile.

Note: Use caution when applying a filter to the Ethernet interface. You could inadvertently render the MAX TNT inaccessible from the local LAN.

Following is an example that applies a packet filter to a local network interface:

1 Open the Ethernet profile for the interface and list its contents:

```
admin> dir ether

     8  12/11/1996 15:58:08  { shelf-1 controller 1 }
    16  12/18/1996 16:17:17  { shelf-1 slot-12 1 }
    16  12/18/1996 16:17:17  { shelf-1 slot-12 2 }
    16  12/18/1996 16:17:17  { shelf-1 slot-12 3 }
    16  12/18/1996 16:17:17  { shelf-1 slot-12 4 }

admin> read ether {1 12 1}
ETHERNET/{ shelf-1 slot-12 1 } read

admin> list
interface-address* = { shelf-1 slot-12 1 }
mac-address = 00:c0:7b:69:94:38
ether-if-type = utp
filter-name = ""
```

2 Specify the filter's name in the filter-name parameter:

```
admin> set filter-name = web-safe
```

3 Write the Ethernet profile:

```
admin> write
ETHERNET/{ shelf-1 slot-12 1 } written
```

Creating and applying route filters

Filter profiles also contain parameters that specify route filters, which are applied only to RIP update packets. Route filters contain the same basic elements as a packet filter, as described in "Working with Filter profiles" on page 7-4.

Following are the route-filter parameters in a Filter profile, shown with their default settings:

```
FILTER filter-name
   filter-name* = ""
   input-filters
      input-filters[1]—input-filters[12]
         valid-entry = no
         forward = no
         Type = route-filter
         route-filter
            source-address-mask = 0.0.0.0
            source-address = 0.0.0.0
            route-mask = 0.0.0.0
            route-address = 0.0.0.0
            add-metric = 0
            action = none
   output-filters
      output-filters[1]—output-filters[12]
         valid-entry = no
         forward = no
         Type = route-filter
         route-filter
```

```
source-address-mask = 0.0.0.0
source-address = 0.0.0.0
route-mask = 0.0.0.0
route-address = 0.0.0.0
add-metric = 0
action = none
```

A Filter profile can contain up to 12 input and output filter specifications (rules). For route filters, the Type must be set to route-filter, and the forwarding action has no effect.

Route filter rules

Route filters affect only RIP packets. The source-address-mask and source-address parameters specify the source of the incoming RIP packet (the source of the route), and these parameters have meaning only in input filters. A source address and mask of zero matches any source.

The route-mask and route-address specify the destination of the route. When a route in a RIP packet matches this specification, the MAX TNT takes the specified action.

The action parameter specifies what action to take on a route that matches the source

- none (the default)
- accept (Accept the route by allowing it to affect the routing table.)
- deny (Deny the route by not allowing it to affect the routing table.)
- add (Add the value set in the add-metric parameter to the route metric and accept the route.)

Note: Once you apply a route-filter to an interface, all defined input and output filters are applied in sequence to RIP update packets until a match is found. If no match is found for a route, the default action is to deny the route.

Example of a filter that excludes a route

In the following example, input filters deny the route 90.0.0.0 in inbound RIP packets but allow the rest of the routes specified in the packets. The example shows the commands entered and the system's responses:

```
admin> new filter route-test
FILTER/route-test read

admin> list input 1
valid-entry = no
forward = no
Type = generic-filter
gen-filter = { 0 0 no no 00:00:00:00:00:00:00:00:00:00:00:00 00:00:00:0+
ip-filter = { 0 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 none 0 none 0 no }
route-filter = { 0.0.0.0 0.0.0.0 255.0.0.0 90.0.0.0 0 deny }

admin> set valid = yes

admin> set type = route-filter

admin> list route
source-address-mask = 0.0.0.0
source-address = 0.0.0.0
route-mask = 255.0.0.0
```

```
route-address = 90.0.0.0
add-metric = 0
action = none
admin> set route-mask = 255.0.0.0
admin> set route-address = 90.0.0.0
admin> set action = deny
admin> list .. .. 2
admin> set valid = yes
admin> set type = route-filter
admin> set route action = accept
admin> write
FILTER/route-test written
```

Note: In this example route filter, any route that matches filter 1 is rejected and all other routes are accepted (because they match filter 2).

Example of a filter that configures a route's metric

In the following example, an output filter identifies the route 11.0.0.0 in outbound RIP packets and assigns a high metric to that route. The example shows the commands entered and the system's responses:

```
admin> new filter metrics
FILTER/metrics read
admin> list output 1
valid-entry = no
forward = no
Type = generic-filter
gen-filter = { 0 0 no no 00:00:00:00:00:00:00:00:00:00:00:00 00:00:00:0+
ip-filter = { 0 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 none 0 none 0 no }
route-filter = { 0.0.0.0 0.0.0.0 255.0.0.0 90.0.0.0 0 deny }
admin> set valid = yes
admin> set type = route-filter
admin> list route
source-address-mask = 0.0.0.0
source-address = 0.0.0.0
route-mask = 255.0.0.0
route-address = 90.0.0.0
add-metric = 0
action = none
admin> set route-mask = 255.0.0.0
admin> set route-address = 11.0.0.0
admin> set add-metric = 7
admin> set action = add
admin> write
FILTER/metrics written
```

Applying a route filter to an interface

Route filters examine packets passed across an interface in the MAX TNT. They can be applied to an IP-routing LAN or WAN interface that makes use of RIP. Following are the parameters related to applying a route filter, shown with sample settings:

```
IP-INTERFACE {{shelf-N slot-N N} N}
   route-filter = route-test

CONNECTION station
   ip-options
      route-filter = route-test
```

For information about each parameter, see the *MAX TNT Reference Guide*.

Applying a route filter to a WAN interface

Following is an example that applies a route filter in a Connection profile:

1 Open the Connection profile and list its ip-options subprofile:

```
admin> read conn bdv
CONNECTION/bdv read

admin> list ip-options
ip-routing-enabled = yes
vj-header-prediction = yes
remote-address = 10.1.2.3/24
local-address = 0.0.0.0/0
routing-metric = 1
preference = 60
down-preference = 120
private-route = no
multicast-allowed = no
address-pool = 0
ip-direct = 0.0.0.0
rip = routing-recv-only-v2
route-filter = ""
ospf-options = { no 0.0.0.0 normal 30 120 5 simple ******* 10 1000 type+
multicast-rate-limit = 100
client-dns-primary-addr = 0.0.0.0
client-dns-secondary-addr = 0.0.0.0
client-dns-addr-assign = yes
client-default-gateway = 0.0.0.0/0
```

2 Specify the filter's name in the route-filter parameter:

```
admin> set route-filter = route-test
```

3 Write the Connection profile:

```
admin> write
CONNECTION/bdv written
```

Applying a route filter to a LAN interface

Following is an example that applies a route filter to a local network interface:

1 Open the IP-Interface profile for the interface and list its content:

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

2 Specify the filter's name in the route-filter parameter:

```
admin> set route-filter = route-test
```

3 Write the IP-Interface profile:

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

Authentication Methods

A

This appendix discusses the following topics:

Introduction

Authentication is the first line of defense against unauthorized access to your network. It uses an exchange of information, such as passwords, to verify the identity of a user. The information is usually encrypted at both ends.

This appendix describes the methods of WAN connection authentication supported in the MAX TNT. For information about controlling which SNMP stations and login users can access the MAX TNT itself, and about limiting what a user can do once they have accessed the terminal server or your IP network, see Appendix B, "Authorization Options."

What are your options?

The MAX TNT supports a variety of authentication methods. You can:

- Use calling-line ID (CLID), which may be provided by the telco as part of the call, to verify that the call is being placed from a trusted telephone number.
- Use Dial Number Information Service (DNIS), which may be provided by the telco as part of an ISDN message, to verify the called number.
- Use callback security. After authentication is complete, the MAX TNT can hang up and call back, ensuring that the connection is made only with a trusted number.
- Use expect-send scripts to authenticate logins to the terminal server.
- Specify protocols to use for password authentication of PPP calls. Following are the supported protocols:
  - Password Authentication Protocol (PAP).
  - PAP with encryption (PAP-DES).
  - Challenge Handshake Authentication Protocol (CHAP), which includes encryption.
  - Microsoft's extension of CHAP (MS-CHAP).
- Use token cards to overcome the limitations of static passwords. You can authenticate token cards by means of TOKEN-PAP, TOKEN-CHAP, CACHE-TOKEN.

Password encryption

All of the available authentication methods except PAP and TOKEN-PAP include password encryption. Password encryption protects against passive attacks, in which an unauthorized user monitors information being transmitted and tries to use it later to establish what appears to be a valid session.

Enhanced security with token cards

Token cards protect against both passive attacks and replay attacks, in which an unauthorized user records valid authentication information exchanged between systems and then replays it later to gain entry. Because token cards provide one-time-only passwords, the password changes many times a day, making replay impossible.

Choosing an authentication method

In determining which authentication method to use, you should consider whether the call is between two machines or between a human being and a machine, and then decide how strong the authentication mechanism must be.

For example, if the connection is negotiated between two machines, you should consider whether the other location is trusted, whether that machine protects its own networks against security attacks, and whether it is physically accessible to many users.

If the connection is negotiated with a user who must type in a token or password, you should consider how secure the password is and how frequently you want it to change. Once the user's connection is authenticated, you can use authorization restrictions to prevent the caller from accessing systems or networks you want to protect, as described in Appendix B, "Authorization Options."

Requiring configured profiles for all callers

When the MAX TNT ships from the factory, it is set to not require any authentication. If a caller were to access the unit with its default settings, the MAX TNT would search for a Connection profile to build the session. When it found no configured profile for the caller, it would check the value of the profiles-required parameter in the Answer-Defaults profile. The parameter's default setting would enable the MAX TNT to build a temporary profile for the unknown caller.

Changing the default setting

Ascend recommends that you configure the MAX TNT to require configured profiles for all callers. To do so, set the profiles-required parameter to Yes, as shown in the following example:

```
admin> read answer
ANSWER-DEFAULTS read
admin> set profiles-required = yes
admin> write
ANSWER-DEFAULTS written
```

How the MAX TNT searches for profiles

For IP routing connections, the MAX TNT looks for a profile that matches the client's IP address. It then uses the name and password in that profile to authenticate the session. If a matching IP address is not found (for example, if the connection is assigned an IP address dynamically), the MAX TNT searches for a Connection profile that matches the name and password presented by the caller.

If the MAX TNT does not find a local Connection profile, it looks for a configuration that requires authentication by an External Authentication Server (EAS) such as RADIUS, TACACS, or TACACS+. Large sites often use an EAS to centralize the management of thousands of connections.

Note: When you have configured the MAX TNT to use an EAS, you can specify that it should search that database first, before checking its local profiles. Just set the local-profiles-first parameter to No in the External-Auth profile.

Using call information for authentication

To have the MAX TNT extract CLID or DNIS information the telco includes with each call, and use the information to authenticate the call, you only have to set one parameter in the Answer-Defaults profile and a few parameters in each affected Connection profile. Following are the relevant parameters with examples of their settings:

```
ANSWER-DEFAULTS
   clid-auth-mode = clid-prefer
CONNECTION station
   clid = 555-1212
   calledNumber = 1234
```

First, set the Answer-Defaults profile's clid-auth-mode parameter to specify what the MAX TNT should do with the telco information. The setting you select applied to both CLID and DNIS.

Next, specify the calling-line ID (CLID) or DNIS number (calledNumber) in a Connection profile. The MAX TNT compares the call information to the number in the profile, and can reject the call if the numbers don't match.

If you are using callback, specify the dial number, which the MAX TNT will call back to establish the authenticated session. Also set the callback parameter to Yes, and make sure the answer-originate parameter allows the MAX TNT to both place and receive calls. Following are the related parameters with examples of their settings:

```
CONNECTION station
   dial-number = 1212
   telco-options
      answer-originate = ans-and-orig
      callback = yes
```

Considerations

The MAX TNT matches CLID and DNIS numbers *before* password authentication. Callback, which causes the MAX TNT to hang up on the caller and use the dial number to call back, occurs *after* password authentication.

CLID

You can use CLID authentication only where the call information is available end-to-end and ANI (Automatic Number Identification) applies to the call. In some areas, the WAN provider might not be able to deliver CLIDs, or a caller might keep a CLID private.

CLID authentication occurs before the MAX TNT accepts a call and begins the process of authenticating a password. Typically, people use CLID authentication to protect against the situation where an unauthorized user obtains the name, password, and IP address of an authorized user, and calls in from another location.

Note: In systems using e1-chinese-signaling, the caller-id parameter in the E1 line profile must be set to get-caller-id for CLID authentication to work.

DNIS or called number

An ISDN message presents the called number (which typically is the number dialed by the far end) as part of the call when DNIS is in use. The phone company may present a modified called number for DNIS, in which case the calledNumber parameter in a Connection profile must specify the modified number.

The MAX TNT can use the called number for authentication, but more often it uses the called number to direct the inbound calls to a particular device.

Configuring the MAX TNT to use call information

By default, the MAX TNT is set to ignore CLID and DNIS information, even if the caller presents it. To use the information, set the clid-auth-mode parameter in the Answer-Defaults profile, as shown in the following example:

```
admin> read answer
ANSWER-DEFAULTS read
admin> set clid-auth-mode = clid-prefer
admin> write
ANSWER-DEFAULTS written
```

Ignore is the default setting, which means the MAX TNT doesn't require a matching ID in the call and doesn't use an ID even if the call contains one.

Note: RADIUS profiles have special requirements for CLID authentication. For details, see the *MAX TNT RADIUS Guide*.

Using the CLID information

When you set clid-auth-mode to clid-require, the MAX TNT must receive a CLID from the call, or it refuses the call. When it receives a CLID, it tries to match the CLID to the CLID parameter in a Connection profile or to a RADIUS user profile set up for CLID authentication. If the MAX TNT doesn't receive a CLID or doesn't find a matching profile, it refuses the call.

When you set clid-auth-mode to clid-prefer and the MAX TNT receives a CLID from the call, it tries to match the CLID to the CLID parameter in a Connection profile or to a RADIUS user profile set up for CLID authentication. (See the *MAX TNT RADIUS Guide*.) If the MAX TNT doesn't receive a CLID or doesn't find a matching profile, it uses whatever authentication is configured in the Answer-Defaults profile.

The clid-fallback setting applies only when a RADIUS server authenticates the connection. The MAX TNT must receive a CLID from the call, or it refuses the call. However, it attempts to match the CLID only if the RADIUS server is available. If the MAX TNT doesn't receive a response from the RADIUS server, it uses whatever authentication is configured in the Answer profile.

Using the called number

When you set clid-auth-mode to dnis-require, the MAX TNT must receive a called number from the call, or it refuses the call. When it receives a called number, it tries to match the number to the calledNumber parameter in a Connection profile or to a RADIUS user profile set

up for called-number authentication. If the MAX TNT doesn't receive a called number or doesn't find a matching profile, it refuses the call.

When you set clid-auth-mode to dnis-prefer and the MAX TNT receives a called number from the call, it tries to match the number to the calledNumber parameter in a Connection profile or to a RADIUS user profile set up for called-number authentication. If the MAX TNT doesn't receive a called number or doesn't find a matching profile, it uses whatever authentication is configured in the Answer-Defaults profile.

Specifying the CLID in a Connection profile

After you have set the clid-auth-mode parameter in the Answer-Defaults profile, specify the number to be matched in a Connection profile. For example:

```
admin> read conn tommy
CONNECTION/tommy read
admin> set clid = 555-1212
admin> write
CONNECTION/tommy written
```

Note: In systems using e1-chinese-signaling, the caller-id parameter in the E1 line profile must be set to get-caller-id for CLID authentication to work.

Specifying the called number in a Connection profile

After you have set the clid-auth-mode parameter in the Answer-Defaults profile, specify the number to be matched in a Connection profile. For example:

```
admin> read conn tommy
CONNECTION/tommy read
admin> set calledNumber = 1234
admin> write
CONNECTION/tommy written
```

Using callback for added security

Companies use callback for a variety of reasons, such as savings on phone charges, but the primary use is for security: to ensure that the connection is made with a known phone number. Hanging up and calling back adds a level of certainty that the connection is with a trusted user, especially because the MAX TNT does so immediately after verifying the user's name and password. For the MAX TNT to use callback, it must be able to both receive and initiate calls. Following is an example that configures the answer-originate and callback parameters:

```
admin> read conn tommy
CONNECTION/tommy read
admin> set dial-number = 1212
admin> set telco answer-originate = ans-and-orig
admin> set telco callback = yes
admin> write
CONNECTION/tommy written
```

Authenticating Telnet logins

Once you have set up a basic IP configuration in the MAX TNT system as described in Chapter 4, “IP Router Configuration,” users can Telnet into the MAX TNT command line. The users can initiate a Telnet session to the MAX TNT from a local workstation or from a WAN connection. In both cases, the MAX TNT authenticates the session by means of a User profile, which defines a permission level for the user logging in. For details of User profiles, see the *MAX TNT Reference Guide*.

In addition to the password required by a User profile, you can specify that Telnet requires its own password authentication, which occurs prior to any User profile authentication. Following is the parameter for setting a system-wide Telnet password, with its default value:

```
IP-GLOBAL
   telnet-password = ""
```

By default, the system has the null password for Telnet access, which means the MAX TNT does not prompt users for a password. The following example sets the Telnet password:

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

admin> set telnet-password = secret-password

admin> write
IP-GLOBAL written
```

where *secret-password* can be up to 20 characters. After setting the Telnet password, users are prompted for that password first. If they specify the correct Telnet password, the MAX TNT prompts again for a user name and password to authenticate a User profile. In the following example, a user starts a Telnet session to a MAX TNT unit named “tnt01,” which has a configured Telnet password.

```
% telnet tnt01
<tnt01> Enter password:

Trying 10.1.2.3 ...
Connected to tnt01.abc.com.
Escape character is ‘^]’.
User:
```

After specifying the correct Telnet password, the user is prompted for a user name and password to authenticate a User profile.

Authenticating terminal-server connections

Terminal-server connections are asynchronous calls that are usually initiated by a dial-in user. Depending on the dial-in client software, the call may initiate a login session or an asynchronous Point-to-Point Protocol (PPP) connection.

When the terminal server receives an asynchronous call, it waits briefly to receive a PPP packet. If it times out waiting for the packet, the terminal server sends its Login prompt. When it receives a name and password from the caller, it authenticates them against a Connection profile or by means of an external authentication server and then performs one of the following actions:

- Displays the terminal-server command-line prompt
- Displays a menu of up to four hosts the user can log into
- Immediately logs the user into a designated host
- Proceeds to initiate a PPP or SLIP session with the user

For information about authorizing one of these actions for incoming login sessions, see Appendix B, "Authorization Options."

When the terminal server receives an asynchronous call and immediately detects a PPP packet, it does not send a Login prompt. Instead, it responds with a PPP packet, and Link Control Protocol (LCP) negotiation begins, including negotiation for PAP or CHAP authentication. Establishment of the connection then proceeds as for a regular (synchronous) PPP session. (See "Authenticating PPP connections" on page A-12 for the details of how names and passwords are exchanged in PPP sessions.)

Recommended settings for modem and terminal-adapter calls

When the MAX TNT receives a call from a modem or a V.120 terminal adapter (TA), the call passed to the terminal-server software. Depending on the dial-in client software, the call may initiate a login session or an asynchronous PPP connection. Table A-1 shows some recommended settings for the dial-in client software used with modems or terminal adapters:

Table A-1. Recommended authentication settings for terminal-server calls

Setup	Recommendation
Analog modem with PPP software	The caller's PPP dial-in software immediately begins sending PPP packets, so its configuration should not include an expect-send script. The password in the caller's Connection profile is authenticated during LCP negotiation. See "Authenticating PPP connections" on page A-12.
Analog modem with a communications package	If the dial-in software does not use PPP, it waits for a prompt and then either executes an expect-send script or waits for the user to manually log in. This is an example script: `expect "Login:" send *$username* expect "Password:" send *$password*`
V.120 TA dialing a PPP connection	If the TA is configured to run PPP, it can handle PAP or CHAP authentication as well as other PPP or MP features the TA supports. The TA immediately begins sending PPP packets, so its configuration should not include an expect-send script. The password in the caller's Connection profile is authenticated during LCP negotiation.
V.120 TA with PPP turned off	If the V.120 TA is configured to run without PPP, it does not support PAP or CHAP authentication. It waits for a prompt and then either executes an expect-send script or waits for the user to manually log in. This is an example script: `expect "Login:" send *$username* expect "Password:" send *$password*`

How security mode affects authentication

You may choose to assign the terminal server its own password, to protect the command-line from unauthorized access. If you assign a terminal-server password, you must also set the security-mode parameter to specify how to use it. The security mode determines whether users must supply a password to access the terminal server. Following are the relevant parameters with their default settings:

```
TERMINAL-SERVER
   security-mode = none
   terminal-mode-configuration
      system-password = ""
```

The meaning of the security-mode setting depends partly on whether users log into menu mode or terminal mode. Table A-2 shows your choices:

Table A-2. Security modes in the terminal server

Setting	Effect
None	Users are not prompted for a login name and password when accessing the terminal server.
Partial	Users must supply a name and password to enter the terminal server except when entering menu mode.
Full	Users must always supply a name and password to enter the terminal server.

Following is an example that configures full password security in the terminal-server:

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

admin> set security-mode = full

admin> set terminal system-password = password

admin> write
TERMINAL-SERVER written
```

where *password* is a password up to 15 characters.

Specifying authentication strings

A dial-in user logging into the terminal-server typically receives the following sequence of prompts:

```
** Ascend TNT Terminal Server **

System Password:

Login:
Password:
```

Following are the parameters that define which strings are sent to and expected from a dial-in user during the login process:

```
TERMINAL-SERVER
   terminal-mode-configuration
      system-password = *******
      banner = "** Ascend TNT Terminal Server **"
      login-prompt = "Login: "
      password-prompt = "Password: "
      prompt = "ascend% "
```

The banner parameter defines the first line sent to the dial-in user. Alternatively, RADIUS can set a multi-line banner.

The system-password parameter specifies a password required to gain access to the terminal server. If the password is null or the security-mode parameter is set to None, the MAX TNT does not send the "System Password" prompt. See "How security mode affects authentication" on page A-9.

The login-prompt and password-prompt parameters specify the next two lines sent to the dial-in user. The MAX TNT uses name and password supplied by the user to authenticate a Connection profile or a profile on an external authentication server.

If you change the login and command-line prompt default settings, make sure that the users' expect-send scripts are written to expect the strings you specify.

Following is an example that configures the banner, login prompts, and command-line prompt:

```
admin> read terminal-server
TERMINAL-SERVER read
admin> set terminal banner = "ABC Corp. Terminal Server"
admin> set terminal login-prompt = "Name:"
admin> set terminal password-prompt = "Password:"
admin> set terminal prompt = "abc: "
admin> write
TERMINAL-SERVER written
```

The expect-send script on the other side must expect compatible strings. For example:

```
expect "Name:" send username expect "Password:" send password expect
"ABC Corp. Terminal Server" send "" expect "abc: " send "slip"
```

How immediate mode affects authentication

If the terminal-server is set up for immediate mode, it directs the data stream of a call directly to a host for login using the specified service (TCP, Rlogin, or Telnet). For information about immediate mode, see "Authorizing immediate mode access" on page B-6 of Appendix B, "Authorization Options."

If the incoming call is TCP-clear (unencapsulated) or V.120, the call is authenticated in the terminal-server as usual and then directed to the Telnet host, where the user logs in according to the login sequence on that host. In this case, immediate mode does not affect terminal-server authentication.

However, if the call uses PPP encapsulation, the normal course of events for the MAX TNT is to authenticate the call using PAP or CHAP and then establish an async PPP session using the

router software. To avoid redirection of the call and enable the user to log into the Telnet host instead, you must set the telnet-host-auth parameter to Yes. Following is the related parameter:

```
TERMINAL-SERVER
   immediate-mode-options
      telnet-host-auth = no
```

The telnet-host-auth parameter is related only to asynchronous PPP calls in immediate mode. If it is set to No, the PPP calls fail. If it is set to Yes, the MAX TNT terminal-server processes the calls and directs them to the Telnet host rather than the unit's router software.

When to use the third prompt

Some RADIUS servers require an additional "third" login prompt, defined by the Ascend-Third-Prompt attribute (213). If the call is authenticated by RADIUS and the profile specifies a value for this attribute, you should configure the terminal-server to display the required prompt. If RADIUS expects a third prompt, it always expects it last, after the regular login sequence.

Some ISPs use a terminal server that follows a login sequence different from that used by Ascend, for example, one that includes a menu selection prior to login. If that is the case at your site, you should configure the terminal-server to display the required prompt and specify that it should be displayed first, to mimic the other terminal server and retain compatibility with client software in use by subscribers.

Following are the relevant parameters:

```
TERMINAL-SERVER
   terminal-mode-configuration
      third-login-prompt = ""
      third-prompt-sequence = last
```

The following example shows how to configure these parameters for a RADIUS server that expects a third prompt:

```
admin> read terminal-server
TERMINAL-SERVER read
admin> set terminal third-prompt = Third-Prompt>>
admin> set terminal third-prompt-sequence = last
admin> write
TERMINAL-SERVER written
```

The next example shows how to configure these parameters to mimic another terminal server that expects users to select a service prior to login:

```
admin> read terminal-server
TERMINAL-SERVER read
admin> set terminal third-prompt = Service?
admin> set terminal third-prompt-sequence = first
admin> write
TERMINAL-SERVER written
```

Authenticating PPP connections

During establishment of a PPP data link, the dialing and answering units exchange Link Control Protocol (LCP) packets to establish communications and configure the link. When the link is established, PPP provides for an optional authentication exchange before exchanging Network Control Protocols (NCPs) to set up the link's network-layer protocols.

If a PPP link is configured to require authentication, the units at each end negotiate an authentication protocol. A multilink connection begins with authentication of a base channel. Subsequent channels are authenticated separately when the additional connections are dialed.

PPP authentication in the Answer-Defaults profile

The Answer-Defaults profile specifies whether the MAX TNT rejects incoming PPP calls that do not offer any authentication protocol. You can also use the profile to restrict which authentication protocols the MAX TNT accepts. Following is the relevant parameter with an example setting:

```
ANSWER-DEFAULTS
   ppp-answer
      receive-auth-mode = pap-ppp-auth
```

The receive-auth-mode parameter typically specifies a general setting to support the widest range of authentication protocols. For example:

```
admin> read answer
ANSWER-DEFAULTS read

admin> set ppp receive-auth-mode = any-ppp-auth

admin> write
ANSWER-DEFAULTS written
```

When you specify "any-ppp-auth" as the method of PPP authentication, the MAX TNT accepts incoming PPP calls that support any of the authentication methods, but it drops connections that do not offer any authentication protocols during LCP negotiation.

If you set a specific authentication method, such as PAP or CHAP, the MAX TNT drops connections that do not support that protocol.

If you leave the default "no-ppp-auth" setting, the MAX TNT accepts any incoming PPP call, including those that do not offer any authentication protocols during LCP negotiation.

PPP authentication in Connection profiles

A Connection profile contains the password expected from the caller, and can also specify the authentication protocol and password used to send reciprocal authentication information back to the far end. Following are the related parameters with examples of their settings:

```
CONNECTION station
   ppp-options
      send-auth-mode = any-ppp-auth
      send-password = remote-password
      recv-password = local-password
```

The send-password setting is the password the MAX TNT sends to the far end as part of the initial handshake, and recv-password specifies the password the MAX TNT expects from the far end. For some connections, the send-password might not be required.

The send-auth-mode parameter sets the authentication method the MAX TNT specifies for this PPP connection. The far end of the connection must support the protocol, or the MAX TNT drops the call. The parameter supports the following settings:

- no-ppp-auth (No authentication)
- any-ppp-auth (Any of the supported methods)
- pap-ppp-auth (PAP)
- des-pap-ppp-auth (PAP with DES)
- chap-ppp-auth (CHAP)
- ms-chap-ppp-auth (MS-CHAP)

Note: Additional settings are provided for connections using token cards. For details, see "Token authentication methods for dial-out connections" on page A-18.

If set to any-ppp-auth, the MAX TNT starts with the strongest authentication method (CHAP) and negotiates down to the strongest one that the far end supports.

PAP authentication

Simple PAP is a two-way handshake method of establishing a caller's identity. Used once, during the initial establishment of the data link, it is not a strong authentication method. Passwords are sent as plain text, so they are subject to eavesdroppers using software that monitors information sent across a network.

PAP authentication is typically used only when the dial-in device does not support a stronger authentication method, such as CHAP, or when the remote device requires a plain text password.

Following is an example that configures a connection for PAP authentication:

```
admin> read conn robin
CONNECTION/robin read
admin> set ppp send-auth-mode = pap-ppp-auth
admin> set ppp send-password = remote-password
admin> set ppp recv-password = local-password
admin> write
CONNECTION/robin written
```

PAP with DES

An extension of PAP adds the U.S. Data Encryption Standard (DES) cipher to data transmissions. The caller applies the encryption algorithm to a PPP packet and places the resulting cipher text in the information field of another PPP packet. The receiving end applies the inverse algorithm and interprets the resulting plain text as if it were a PPP packet that had arrived directly on the interface.

Following is an example that configures a connection for PAP with DES:

```
admin> read conn dave
CONNECTION/dave read
admin> set ppp send-auth-mode = des-pap-ppp-auth
admin> set ppp send-password = remote-password
admin> set ppp recv-password = local-password
admin> write
CONNECTION/dave written
```

CHAP authentication

CHAP authentication verifies the caller's identity by using a three-way handshake upon initial link establishment, and possibly repeating the handshake any number of times. The authenticator sends a challenge to the caller, which responds with an MD5 digest calculated from the password. The authenticator then checks the digest against its own calculation of the expected hash value to authenticate the call. A new challenge may be sent at random intervals.

CHAP is a stronger authentication method than PAP, because the password is not send as plain text. In addition, the use of repeated challenges limits the time of exposure to any single attempt to break the encryption code, and the authenticator is in control of how often and when challenges are sent.

Following is an example that configures a connection for CHAP authentication:

```
admin> read conn matt
CONNECTION/matt read
admin> set ppp send-auth-mode = chap-ppp-auth
admin> set ppp send-password = remote-password
admin> set ppp recv-password = local-password
admin> write
CONNECTION/matt written
```

MS-CHAP authentication

Microsoft CHAP (MS-CHAP) is a close derivative of CHAP. However, CHAP is designed to authenticate WAN-aware secure software, and is not widely used to support remote workstations, where an insecure plain text login might be required. MS-CHAP addresses this issue, and also integrates the encryption and hashing algorithms used on Windows networks. Microsoft Windows NT and LAN Manager platforms implement MS-CHAP.

Following is an example that configures a connection for MS-CHAP authentication:

```
admin> read conn ted
CONNECTION/ted read
admin> set ppp send-auth-mode = ms-chap-ppp-auth
admin> set ppp send-password = remote-password
admin> set ppp recv-password = local-password
admin> write
CONNECTION/ted written
```

Authentication using token cards

The MAX TNT supports token-card authentication by using a RADIUS server as the intermediary between the MAX TNT unit answering the call and an External Authentication Server (EAS) such as a Security Dynamics ACE/Server or an Enigma Logic SafeWord server. For the details of configuring the RADIUS server to communicate with the EAS, see the *MAX TNT RADIUS Guide*.

Token cards are hardware devices, typically shaped liked credit-card calculators, with an LCD display that informs users about the current, one-time-only token (password) that will enable access to a secure network. The current token changes many times a day. Token cards keep the changing authentication information continuously up-to-date by maintaining a synchronized clock with an EAS such as an ACE/Server or SafeWord server. Authorized users must have the token card in their possession to gain access to a secure network.

If the EAS is ACE/Server, the user has a SecurID token card that displays a randomly generated access code, which changes every 60 seconds.

If the EAS is SafeWord, the user can have one of the following types of token cards:

- ActivCard
- CryptoCard
- DES Gold
- DES Silver
- SafeWord SofToken
- SafeWord MultiSync
- DigiPass
- SecureNet Key
- WatchWord

The MAX TNT supports the use of token cards only through RADIUS. The RADIUS server must be configured to interact with the EAS modules, which typically run on the same physical system as the RADIUS server.

Note: When simple RADIUS authentication is in use, the RADIUS server itself acts as the EAS. When token-card authentication is in use, the RADIUS server passes the authentication request on to an ACE/Server or SafeWord server, and that system is referred to as the EAS. This does not affect the MAX TNT External-Auth profile configuration, which must still specify RADIUS as the external server.

Authenticating dial-in connections by means of tokens

Figure A-1 shows a dial-in connection to the MAX TNT. The remote user must use a token card to gain access to the secure network.

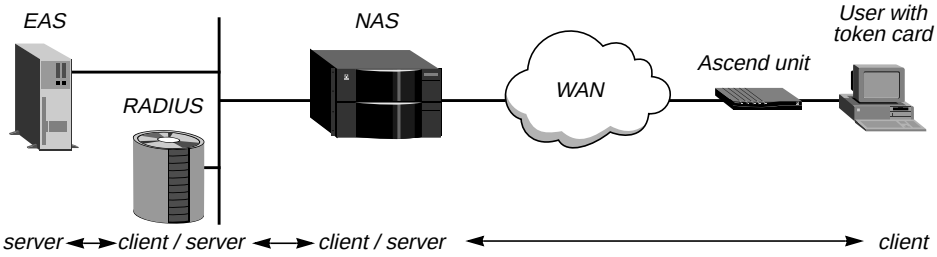


Figure A-1. Token card authentication for dial-in connections

A user with a token card initiates a connection to the MAX TNT, usually from terminal-server password mode in another Ascend unit. By dialing up the MAX TNT (the Network Access Server, or NAS), the user becomes a client of the NAS.

The MAX TNT sends an Access-Request packet to the RADIUS server to authenticate the incoming call, becoming a client of the RADIUS server. (For details of Access-Request packets, see the *MAX TNT RADIUS Guide*.)

The RADIUS server forwards the connection request to the EAS (ACE/Server or SafeWord server), becoming a client of the EAS.

The EAS sends an Access-Challenge packet back through the RADIUS server and the MAX TNT to the user dialing in. The user sees the challenge message, obtains the current password from his or her token card, and enters that password in response to the challenge message. The password travels back through the NAS and the RADIUS server to the EAS.

The EAS sends a response to the RADIUS server, specifying whether the user has entered the proper user name and password. If the user enters an incorrect password, the EAS returns another challenge and the user can again try again. After three incorrect attempts, the MAX TNT terminates the call.

Finally, the RADIUS server sends an authentication response to the MAX TNT. If authentication is unsuccessful, the MAX TNT receives an Access-Reject packet. If authentication is successful, the MAX TNT receives an Access-Accept packet containing a list of attribute-value pairs from the user profile in the RADIUS server's database. The MAX TNT uses the attribute-value pairs to create the connection.

Configuring the MAX TNT as the NAS

To configure the MAX TNT to function as the NAS as shown in Figure A-1, you must set up the Answer-Defaults profile to allow the appropriate authentication method. For example, you might set the receive-auth-mode parameter to any-ppp-auth, as described in "PPP authentication in the Answer-Defaults profile" on page A-12.

You must also set up the External-Auth profile to authenticate the connections via RADIUS. For details, see the *MAX TNT RADIUS Guide*.

Using the MAX TNT to dial out to a secure network

Figure A-1 shows a dial-out connection from the MAX TNT. The local user must use a token card to gain access to the remote secure network.

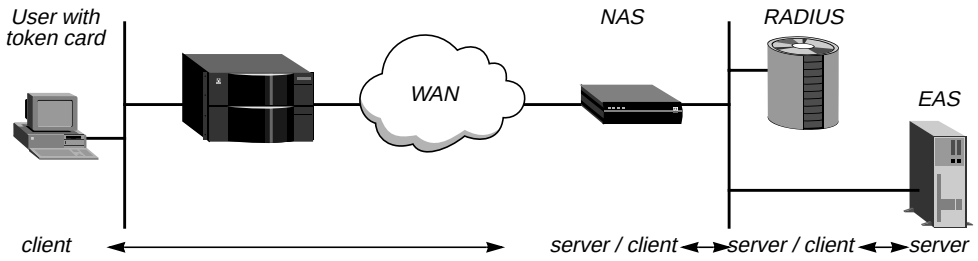


Figure A-2. Token card authentication for dial-out connections

A local user with a token card initiates a connection by logging into the MAX TNT terminal server, invoking password mode, and dialing out to the remote NAS. These actions require that the user have login privileges to the MAX TNT unit (a User profile), and the MAX TNT must have a Connection profile configured for a connection to the remote NAS.

After logging into the MAX TNT and invoking the terminal server, a user can enable password mode and respond to token challenges as follows:

1 At the terminal server prompt, enter the Set Password command:

```
ascend% set password
```

The following message appears:

```
Entering Password Mode...
```

The prompt changes to the following:

```
[^C to exit] Password Mode>
```

2 Bring up the connection in the usual way: by invoking a program that requires a connection to a host on the remote secure network, or by using one of the MAX TNT digital modems to dial the remote NAS.

While the connection is being negotiated, the remote NAS returns a challenge prompt that looks like this:

```
From: hostname
0-Challenge: challenge

Enter next password:
```

3 At the challenge prompt, enter the token obtained from the token card.

A user has 60 seconds to enter the token correctly, and thus establish the connection to the secure network.

If the token is not entered within 60 seconds, the login attempt times out. If the token is entered incorrectly, the challenge prompt is displayed again up to three times. The hostname displayed in the first line of the challenge is the name of the NAS called (such as a remote MAX unit). It is optional on some systems.

If the send-auth-mode parameter in the Connection profile is configured incorrectly, no challenge prompt appears, or the user sees an error message such as the following:

```
From: hostname
Received unexpected PAP Challenge!... check PPP Auth Mode
```

4 To return to normal terminal server operations, the user presses Ctrl-C at the Password Mode prompt.

Token authentication methods for dial-out connections

The Connection profile to the remote NAS must allow dialout if that is the method used to bring up the connection. In addition, the profile must specify an authentication method that handles token challenges. The calling unit requests the specified authentication method, but the RADIUS daemon and user profile accessed by the answering NAS determine which method is actually used. See the *MAX TNT RADIUS Guide* for related information.

The following authentication methods enable the MAX TNT to add bandwidth to the connection on demand, without prompting the caller to supply the most recent token:

- token-pap-ppp-auth
- token-chap-ppp-auth
- cache-token-ppp-auth

Token PAP

Token PAP (token-pap-ppp-auth) is an extension of PAP authentication. The token supplied by a user is sent in the clear (via PAP), but because the password is one-time-only, the security risk is usually not serious.

Following is an example that configures a connection for token PAP authentication:

```
admin> read conn don
CONNECTION/don read
admin> set ppp send-auth-mode = token-pap-ppp-auth
admin> set ppp send-password = placeholder
admin> write
CONNECTION/don written
```

where ***placeholder*** is a password sent to the remote NAS, but not used in negotiating the session. The remote NAS expects a name and password for PAP authentication, so the Connection profile supplies one. However, the RADIUS user profile for this connection uses a special password, such as Password = ACE (to indicate ACE token authentication), and the password sent by the MAX TNT is ignored.

A password challenge is returned to the user, and the token sent in response is used to authenticate the base channel of the connection.

Note: Token PAP is appropriate for single-channel dial-out calls. It is not practical for multi-channel calls, because any time that bandwidth requirements cause another channel to come up, the user is challenged for another token.

Token CHAP

Token CHAP (token-chap-ppp-auth) uses token PAP to authenticate the base channel of the call (as described in the preceding section), and authenticates additional channels by means of CHAP. The RADIUS profile at the far end must be specify appropriate attributes for token CHAP, or token PAP is used instead. (For details, see the *MAX TNT RADIUS Guide*.)

In addition to the PPP authentication parameters (see “PPP authentication in the Answer-Defaults profile” on page A-12), the following parameter can also apply to token CHAP authentication:

```
CONNECTION station
   mpp-options
      aux-send-password
```

Following is an example that configures a connection for token CHAP authentication:

```
admin> read conn bob
CONNECTION/bob read
admin> set ppp send-auth-mode = token-chap-ppp-auth
admin> set ppp send-password = placeholder
admin> set mpp aux-send-password = password-2
admin> write
CONNECTION/bob written
```

where ***placeholder*** is a password sent to the remote NAS, but not used in negotiating the session. The remote NAS expects a name and password for CHAP authentication, so the Connection profile supplies one. However, the RADIUS user profile for this connection uses a special password, such as Password = ACE (to indicate ACE token authentication), and the password sent by the MAX TNT is ignored.

A password challenge is returned to the user, and the token the user sends is used to authenticate the base channel of the call. If the connection requires additional bandwidth, the MAX TNT uses CHAP to send a digest of the auxiliary password specified in the aux-send-password parameter.

Cache token

Cache token (cache-token-ppp-auth) uses CHAP to transmit the initial token, and caches the token for reuse when channels are added to the call, or when a new call is made. The RADIUS profile at the far end must be set up with appropriate attributes that specify how long the token will be cached. (For details, see the *MAX TNT RADIUS Guide*.)

Following is an example that configures a connection for cache token authentication:

```
admin> read conn holly
CONNECTION/holly read
admin> set ppp send-auth-mode = cache-token-ppp-auth
admin> set ppp send-password = placeholder
admin> write
CONNECTION/holly written
```

where ***placeholder*** is a password sent to the remote NAS, but not used in negotiating the session. The remote NAS expects a name and password for CHAP authentication, so the Connection profile supplies one. However, the RADIUS user profile for this connection uses a special password, such as Password = ACE (to indicate ACE token authentication), and the password sent by the MAX TNT is ignored.

The user is prompted for a token to authenticate the base channel of the call via CHAP. If the RADIUS server has been configured correctly, it caches that encrypted token for the specified period, or for the amount of idle time specified for the connection. When channels are added to the call, or when a new call is made, the RADIUS server uses the cached password to authenticate them.

Authorization Options

B

This appendix discusses the following topics:

Introduction

Authorization defines what a user may do once he or she has access to your local area network. Authorization occurs *after* authentication has been completed.

In the MAX TNT, you configure authorization in the following profiles:

- The Terminal-Server profile, to restrict access to the terminal-server software.
- The SNMP profile, to restrict access to the system by means of SNMP manager utilities.
- IP-Global and Connection profiles, to restrict access to certain DNS systems.

For information about using packet filters or firewalls to define which packets may cross your network, see Chapter 7, "Packet and Route Filters."

Terminal-server authorization

The MAX TNT terminal server handles incoming calls initiated by means of a modem or terminal-adapter (TA). These calls are usually initiated by a dial-in user, so authorization is an important part of the setup. For details of authenticating terminal-server logins, see Appendix A, "Authentication Methods."

For the MAX TNT to answer modem or TA calls, the terminal-server software must be enabled. Following is the related parameter:

```
TERMINAL-SERVER
   enabled = yes
```

The following commands enable the terminal-server:

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

admin> set enabled = yes

admin> write
TERMINAL-SERVER written
```

Most sites do not allow dial-in users to access the MAX TNT terminal-server administrative commands, such as Traceroute or Ping. In most cases, the terminal server is used as a stepping stone toward access to one or more network hosts. There are several ways to configure the Terminal-Server profile to enable this type of access:

- Terminal mode, to enable users to invoke Telnet, Rlogin, PPP, or Serial Line IP (SLIP) sessions.
- Immediate mode, to immediately initiate a Telnet, Rlogin, or TCP connection to a designated IP address.
- Menu mode, to display a menu of up to four hosts, from which the user can select one to invoke a Telnet session.

Authorizing terminal-mode access

Typically, administrators set up terminal-mode to negotiate a user-to-host session as part of the dial-in expect-send script. Instead of providing only the login and password needed to authenticate a Connection profile, the script also includes the terminal-server prompt and a

command, such as PPP, SLIP, Telnet, or Rlogin. In this way, the session to a host is invoked as part of the login process, so the user never actually sees the command-line prompt. Alternatively, you can provide access to the command-line and restrict which commands are accessible.

Password-protecting the command line

For details of setting up a password that is always required to access the terminal-server command line, see “How security mode affects authentication” on page A-9 of Appendix A, “Authentication Methods.”

Authorizing network commands

By default, the Terminal-Server profile disables the use of the Ping and Traceroute commands as a security measure, because these commands authorize users to gain information about the network. Following are the parameters related to restricting the use of the Ping and Traceroute commands, with examples of their settings:

```
TERMINAL-SERVER
   terminal-mode-configuration
      ping = no
      traceroute = no
```

You can set the parameters to Yes if you want to enable the use of the Ping and Traceroute commands.

Authorizing interactive logins

By default, the Terminal-Server profile disables the use of the TCP, Rlogin, and Telnet commands, because those commands are provided in immediate mode in a more secure fashion. Following are the parameters related to restricting the use of these commands, shown with their default settings:

```
TERMINAL-SERVER
   terminal-mode-configuration
      tcp = no
      rlogin = no
      telnet-options
         telnet = no
```

The following commands enable the use of the Telnet command from the terminal-server prompt:

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

admin> list terminal telnet
telnet = no
telnet-mode = ascii
auto-telnet = no
local-echo = no

admin> set telnet = yes

admin> write
TERMINAL-SERVER written
```

Setting Telnet session defaults

The following parameters affect Telnet session defaults:

```
TERMINAL-SERVER
   terminal-mode-configuration
      terminal-type = vt100
      clear-call = no
      buffer-chars = yes
      telnet-options
         telnet = yes
         telnet-mode = ascii
         auto-telnet = no
         local-echo = no
```

Users can modify some of the default values an a per-session basis when they invoke the Telnet command. Following is an example that configures some of the session parameters:

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

admin> set clear-call = yes

admin> set telnet auto-telnet = yes

admin> set telnet local-echo = yes

admin> write
TERMINAL-SERVER written
```

The terminal-type specifies a terminal type for the Telnet session, such as the vt100. Clear-call specifies whether when the user terminates a Telnet session, the connection is terminated as well. Buffer-chars determines whether the terminal server buffers input characters for 100 milliseconds before forwarding them to the host, or sends the characters as received.

In the telnet-options subprofile, telnet-mode specifies whether binary, ascii, or transparent mode is the default for Telnet sessions. Auto-telnet instructs the terminal server to interpret unknown command strings as the name of a host for a Telnet session. Local-echo sets a global default for echoing characters locally, which users can change within an individual Telnet session.

Authorizing PPP sessions

Typically, PPP sessions are initiated by dial-in software, such as Netscape Navigator, Microsoft Explorer, or Windows 95 (which has a resident TCP/IP stack). The terminal-server software initially handles a call if it is made by means of a modem or TA. However, after the terminal server detects a PPP packets from the caller, it passes the call on to the router, where it is handled as a regular PPP connection without entering a terminal-server interface.

If the user's dial-in software does not support PPP, the user can still initiate a PPP session from within the terminal-server software. To do so, a user could log into the terminal server in terminal mode and use the PPP command, or include the PPP command in an expect-send script; for example:

```
expect "Login:" send $username expect "Password:" send $password expect
"ascend% " send "ppp"
```

Following are the parameters related to authorizing PPP sessions initiated from the terminal-server software, with their default settings:

```
TERMINAL-SERVER
   ppp-mode-configuration
      ppp = no
      delay = 5
      direct = no
      info = session-ppp
```

For example, the following commands enable PPP sessions, and specify that the MAX TNT should start PPP negotiation immediately the PPP command is invoked:

```
admin> read terminal-server
TERMINAL-SERVER read
admin> set ppp ppp = yes
admin> set ppp direct = yes
admin> write
TERMINAL-SERVER written
```

You can use the delay parameter to instruct the terminal server to transition to packet-mode processing after the specified number of seconds. By setting the info parameter, you can specify that no message is displayed, or choose between `PPP Mode` and `PPP Session`.

Authorizing SLIP sessions

Some applications require SLIP rather than PPP. The MAX TNT does not support a direct SLIP dial-in, because SLIP doesn't support authentication. However, if slip-mode is enabled in the terminal server, users can initiate a SLIP session and then run an application such as FTP in that session. To initiate SLIP, the user must invoke a session from within the terminal-server software. To do so, a user could log into the terminal server in terminal mode and use the SLIP command, or include the SLIP command in an expect-send script; for example:

```
expect "Login:" send $username expect "Password:" send $password expect
"ascend% " send "slip"
```

Following are the parameters related to authorizing SLIP sessions initiated from the terminal-server software, with their default settings:

```
TERMINAL-SERVER
   slip-mode-configuration
      slip = no
      slip-bootp = no
      info = basic-slip
```

For example, the following commands authorize SLIP sessions and specify that the terminal server will respond to BootP in SLIP sessions:

```
admin> read term
TERMINAL-SERVER read
admin> set slip slip = yes
admin> set slip slip-bootp = yes
admin> write
TERMINAL-SERVER written
```

The slip-bootp parameter enables the terminal server to respond to BootP within SLIP sessions. If it is enabled, an interactive user who initiates a SLIP session can get an IP address from the designated IP address pool via BootP. If it is disabled, the terminal server does not run BootP; instead, the system prompts the user to accept an IP address at the start of the SLIP

session. By setting the info parameter, you can specify that a default startup message will be displayed, or an “advanced” message that includes a netmask and IP gateway address.

Authorizing immediate mode access

In immediate mode, the terminal-server does not display the command-line prompt or a menu of hosts, instead it directs dial-in users immediately to a designated host using the specified service: TCP, Rlogin, or Telnet. When it uses Telnet to initiate the connection to the host, you can configure the terminal-server to pass the call to the host before authenticating it. In that case, the Telnet host is responsible for authentication. See Appendix A, “Authentication Methods.”

Following are the parameters required to set up immediate mode, with examples of their settings:

```
TERMINAL-SERVER
   immediate-mode-options
      service = telnet
      telnet-host-auth = no
      host = 10.2.3.4
      port = 514
```

The following example shows how to bypass local authentication and require that the Telnet host handle it instead:

```
admin> read terminal-server
TERMINAL-SERVER read
admin> set immediate service = telnet
admin> set immediate telnet-host-auth = yes
admin> set immediate host = 10.2.3.4
admin> write
TERMINAL-SERVER written
```

If the service parameter is set to none, immediate mode is disabled. The other choices for establishing an immediate host connection for dial-in users are Telnet, Raw-TCP, or Rlogin.

The host parameter specifies the hostname or address to which users will be connected in terminal server immediate mode. You can also specify a TCP port number to use for the connections.

Note: For incoming asynchronous PPP calls, you must set the telnet-host-auth parameter to Yes to prevent redirection of the call to the MAX TNT router software. For details, see “How immediate mode affects authentication” on page A-10 of Appendix A, “Authentication Methods.”

Authorizing menu mode access

In menu mode, the terminal-server does not display the command-line prompt. Instead, it displays a menu of hosts from which a user can select to initiate a Telnet login. If you define the menu locally in the menu-mode-options subprofile, it can display up to four host descriptions. If you define the menu in RADIUS, by means of the Ascend-Host-Info attribute, it can display up to ten host descriptions. For details, see the *MAX TNT RADIUS Guide*.

For details of setting up a password that is required to access the terminal-server when menu mode is in use, or when users toggle from menu-mode to terminal-mode, see “How security mode affects authentication” on page A-9 of Appendix A, “Authentication Methods.”

Following are the parameters that enable you to describe up to four hosts that will be accessible to users in menu mode:

```
TERMINAL-SERVER
   menu-mode-options
      start-with-menus = no
      toggle-screen = no
      remote-configuration = no
      text-1 = ""
      host-1 = ""
      text-2 = ""
      host-2 = ""
      text-3 = ""
      host-3 = ""
      text-4 = ""
      host-4 = ""
```

Setting start-with-menus to Yes means the terminal server brings up the menu upon initial login. If the toggle-screen parameter is set to Yes, users can press 0 (the zero key) in the menu to toggle to the terminal-server command line. You can configure menu-mode to obtain the menu from RADIUS by setting the remote-configuration parameter to Yes. The Text and Host parameters expect a text description and an IP address of up to four hosts, respectively. The MAX TNT uses only the specified IP addresses to access the hosts.

The following example configures the menu shown in Figure B-1, and specifies that the menu should be displayed upon initial login:

```
admin> read terminal
TERMINAL-SERVER read
admin> set menu start-with-menus = yes
admin> set menu text-1 = administration
admin> set menu text-2 = engineering
admin> set menu text-3 = marketing
admin> set menu text-4 = techpubs
admin> set menu host-1 = 10.2.3.4
admin> set menu host-2 = 10.2.3.57
admin> set menu host-3 = 10.2.3.121
admin> set menu host-4 = 10.2.3.224
admin> write
TERMINAL-SERVER written
```

With this configuration, the MAX TNT authenticates the user’s login name and password, and then displays a text-based menu such as the one shown in Figure B-1:

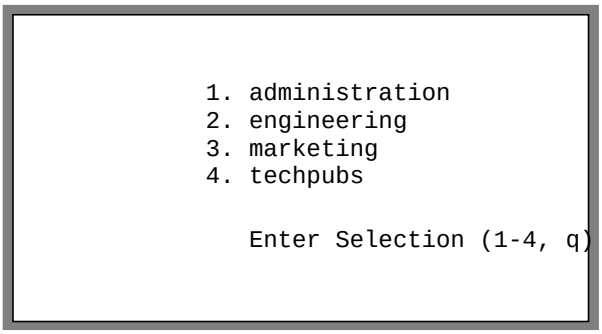


Figure B-1. Menu mode

Users can Telnet to the specified host by pressing 1, 2, 3, or 4, or can quit the menu by pressing Q. Quitting the menu terminates the connection. If the toggle-screen parameter is set to Yes, users can press 0 to exit menu mode and enter the terminal-server command line.

Authorizing access to specific DNS servers

Domain Name Service (DNS) is a TCP/IP service for centralized management of address resolution. Service providers may maintain multiple DNS servers, each one dedicated to a particular client or location. In that case, it may be important for security reasons to ensure that connections are always directed to the correct DNS service. With per-connection DNS access, a service provider can direct specific users to the DNS server appropriate to their service or location. For information about configuring local DNS servers and options, see Chapter 4, “IP Router Configuration.”

What is client DNS?

Client DNS enables the MAX TNT to direct incoming connections to DNS servers belonging to a particular location or customer, and to prevent those users from accessing the local DNS servers. The addresses configured for client DNS servers are presented to WAN connections during IPCP negotiation.

Client DNS has two levels: a global configuration that applies to all PPP connections (defined here in the IP-Global profile), and a connection-specific configuration that applies only to the WAN connection defined in the Connection profile. The client DNS addresses configured in the IP-Global profile are used only if the caller’s configured profile specifies no client servers.

Following are parameters related to configuring client DNS, shown with default values:

```
IP-GLOBAL
   client-primary-dns-server = 0.0.0.0
   client-secondary-dns-server = 0.0.0.0
   allow-as-client-dns-info = True
CONNECTION station
   ip-options
      client-dns-primary-addr = 0.0.0.0
      client-dns-secondary-addr = 0.0.0.0
      client-dns-addr-assign = yes
```

A connection can use one of the following DNS servers:

- The server specified by the dns-primary-server or dns-secondary-server parameter in the IP-Global profile. This server typically provides information about local hosts.

  You can also specify access to this server if no client DNS servers are available.
- The server specified by the client-primary-dns-server or client-secondary-dns-parameter in the IP-Global profile. This server will be used by all PPP connections.
- The server specified by the client-dns-primary-addr or client-dns-primary-addr parameter in a Connection profile.This server will be used only profiles that specify it.

The MAX TNT uses the global addresses only if a Connection profile does not specify any, or if the client-dns-addr-assign has been set to No. You can use the client-dns-addr-assign parameter to turn off client DNS for this connection without deleting its configuration.

Configuring client DNS servers at the system level

Following is an example that configures client DNS servers at the system level:

```
admin> read ip-global
IP-GLOBAL read
admin> set client-dns-pri = 8.22.17.56
admin> set client-dns-sec = 8.22.17.107
admin> write
IP-GLOBAL written
```

The secondary server is accessed only if the primary one is inaccessible. If both client DNS servers in the IP-Global profile are not accessible and the caller's configured profile does not specify a connection-specific client DNS server, the MAX TNT may allow the client to access the local DNS servers, depending on the setting of the allow-as-client-dns-info parameter. Following is an example that allows clients to access local DNS servers when client DNS servers are not found:

```
admin> read ip-global
IP-GLOBAL read
admin> set allow-as-client-dns-info = True
admin> write
IP-GLOBAL written
```

Setting connection-specific DNS parameters

Following is an example that sets up connection-specific DNS parameters:

```
admin> read connection cherry
CONNECTION/cherry read
admin> set ip-options client-dns-primary-addr = 10.2.3.4
admin> set ip-options client-dns-secondary-addr = 10.2.3.56
admin> set ip-options client-dns-addr-assign = yes
admin> write
CONNECTION/cherry written
```

The secondary server is accessed only if the primary one is inaccessible.

Authorizing SNMP stations to access the unit

The MAX TNT supports SNMP on a TCP/IP network. An SNMP management station that uses the Ascend Enterprise MIB can query the MAX TNT, set parameters, sound alarms when certain conditions appear in the MAX TNT, and perform other management tasks.

An SNMP manager must be running on a host on the local IP network, and the MAX TNT must be able to find that host, either via static route or RIP. In addition to these restrictions, the MAX TNT has its own SNMP password security (community strings), which you should set up to protect the MAX TNT from being reconfigured from an unauthorized SNMP station.

Overview of SNMP security

The SNMP profile contains SNMP-readable information related to the unit itself and its SNMP security. There are two levels of security:

- community strings, which must be known by a community of SNMP managers to access the box, and
- address security, which excludes SNMP access unless it is initiated from a specified IP address.

Following are the related parameters:

```
SNMP
   enabled = no
   read-community = public
   read-write-community = write
   enforce-address-security = no
   read-access-hosts = [ 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 ]
   write-access-hosts = [ 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 ]
   contact = ""
   location = ""
```

Enabling SNMP in the MAX TNT

If you leave the enabled parameter in the SNMP profile set to No (the default), no SNMP utilities can access the MAX TNT. To enable SNMP on a unit:

```
admin> read SNMP
SNMP read
admin> set enabled = yes
admin> write
SNMP written
```

Note: The contact and location fields are SNMP readable and settable, and should indicate the person to contact about this unit, and its location.

Setting community strings

SNMP community strings set the administrative authorization policy for executing SNMP Set and Get commands from a management station. When the management station interacts with

the MAX TNT, it must provide the proper community string to gain read access, and provide a separate communicate string to gain write access to the system's configuration.

Default communities are as follows:

- The community string "public" is assigned by default for read access
- The community string "write" is assigned by default for read-write access.

Following is an example that assigns the string "secret" as the string required for and SNMP management station to gain read-write access to the MAX TNT:

```
admin> read SNMP
SNMP read
admin> set read-write-community = secret
admin> write
SNMP written
```

You can specify up to 32 characters in an SNMP community string.

Setting up and enforcing address security

If the enforce-address-security parameter is set to No (its default value), any SNMP manager that presents the right community name will be allowed access. If it is set to Yes, the MAX TNT checks the source IP address of the SNMP manager and allows access only to those IP addresses listed in the read-access-host and write-access-host arrays. Each array can include up to five host addresses.

Following is an example that enforces address security and specifies a trusted address for both read and write access:

```
admin> read snmp
SNMP read
admin> list
enabled = no
read-community = public
read-write-community = write
enforce-address-security = no
read-access-hosts = [ 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 ]
write-access-hosts = [ 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 ]
contact = ""
location = ""
admin> set enforce-address-security = yes
admin> set read-access 1 = 10.2.3.4
admin> set write-access 2 = 10.2.56.123
admin> write
SNMP written
```

Secure Access Firewalls

C

This appendix covers the following topics:

Introduction to Secure Access firewalls

Secure Access is a software option available for the MAX TNT. To purchase Secure Access, contact your Ascend distributor.

Firewalls are similar to packet filters, but they are more complex than filters and typically change dynamically in response to characteristics of packets that pass through them. In general, a firewall can be designed to notice the passage of a packet with some specific bit patterns, and in response to that packet, invoke rules that cause other rules to be created dynamically.

Uploading firewalls

The *Secure Access User's Guide* explains how to use the Secure Access Manager (SAM) application to create a firewall and load firewalls. This section provides some background information about using the SAM to log into the MAX TNT and upload a firewall.

Permissions requirements

The SAM has several command-line and permissions requirements that must be met to successfully load a firewall. When loading to the MAX TNT, the SAM prompts for a user name and password, and uses the specified name and password to select a User profile. Or, if the MAX TNT unit allows access without a password, the SAM uses the default User profile. The User profile must meet the following requirements:

- The prompt must be terminated by a greater-than sign followed by a space ("...> ").
- The profile must enable the allow-system permission.
- The profile must enable the allow-update permission.

When the SAM has logged into the MAX TNT, it prints "Using CLI" in its status window.

Loading the firewall

When you load a firewall, the MAX TNT creates a Firewall profile and assigns the profile the name of the firewall. The MAX TNT does not allow the following characters in names:

- brackets ([])
- braces ({})
- space characters

If the firewall name contains one of these characters, the character is replaced with an underscore in the name of the Firewall profile. A Firewall profile contains the following parameters:

```
FIREWALL name
   name* = my-firewall
   version = 2
   data = [ ACAfiwgAAAAAAAADE2RmZDTiz0zOLeDkBAAFTVl4DAAAAA== ]
```

You apply the firewall to an interface by specifying this name, so the name must be unique across both Filter and Firewall profiles. If you create a duplicate name, either by uploading a firewall or changing the name in a Filter or Firewall profile, an error message is displayed.

The version and data parameters are intended to be set only by the SAM. The version parameter specifies the firewall version; if you change its value in the Firewall profile, one of the following messages will probably be displayed:

```
error: Base 64 decode failed
error: Firewall does not load properly (corrupted?)
```

The data parameter contains information about the firewall definition. If you list the data parameter separately, it is displayed as a sparse array; for example:

```
admin> list data
data[0] = ACAfiwgAAAAAAAADE2RmZDTiz0zOLeDkBAAFTVl4DAAA
data[33] = AA==
data[66] =
...
```

In the SAM File Save As dialog box, there is a type labeled as follows:

```
TNT Profile Files (*.prf)
```

This type represents a set of commands that can be directly piped into a MAX TNT command-line session to store the firewall manually.

Diagnostic commands

The MAX TNT has a FWALLversion diagnostic-level command for displaying the firewall versions supported by the current system software. For example:

```
admin> FWALLversion
1 2
```

The output shows all firewall versions supported in the current code. The version numbers are separated by spaces. The SAM uses this information to ensure that firewalls you uploaded are supported.

The MAX TNT also has a FWALLdblog diagnostic-level command for displaying firewall messages. For example:

```
admin> FWALLdblog
```

By default, the SAM causes a message to be generated for all packets blocked by a firewall. Firewall messages are sent to the logging mechanism configured in the Log profile, such as syslog or the console. For details, see the *MAX TNT Reference Guide*.

Applying a firewall to an interface

For a firewall to take effect, you must apply it to a LAN or WAN interface in the MAX TNT. This section explains how.

How the Answer-Defaults profile settings are used

Following are the Answer-Defaults settings related to firewalls, shown with their default settings:

```
ANSWER-DEFAULTS
   session-info
      call-filter = ""
      data-filter = ""
      filter-persistence = no
```

If the MAX TNT uses a local Connection profile for authentication, it does not use firewalls set in the Answer-Defaults session-info subprofile.

If the MAX TNT relies on RADIUS for authentication and the caller's RADIUS profile applies a filter or firewall (or both), the filter or firewall specified in the user profile is applied to each incoming packet and the MAX TNT does not use those set in the Answer-Defaults session-info subprofile.

If the MAX TNT relies on RADIUS for authentication, the caller's RADIUS profile does not apply a filter or firewall, and the use-answer-for-all-defaults parameter is set to Yes in the Answer-Default profile, filters or firewalls set in the Answer-Defaults session-info subprofile are applied to each incoming packet.

Filter persistence for firewalls

Before Secure Access was supported, the MAX TNT simply constructed a filter on a WAN interface when the connection was established and destroyed the filter when the connection was brought down, even if the connection just timed out momentarily. This works fine for static packet filters, but does not accommodate a firewall. Filter persistence is needed to allow firewalls to persist across connection state changes, but it is not needed for filters. If you do set it for a static packet filter, the filter persists across connection state changes.

Note: Although you can apply a Secure Access firewall as a call filter, filter-persistence does not apply to call-filter firewalls at this release. If the call filter you require is relatively simple, you may want to implement it as a regular packet filter for performance reasons.

Applying a firewall to a WAN interface

Following are the parameters related to applying a firewall to a WAN interface, shown with their default settings:

```
CONNECTION station
   session-options
      call-filter = ""
      data-filter = ""
      filter-persistence = no
```

Following is an example that applies a firewall to a WAN interface:

1 Open the Connection profile and list its session-options subprofile:

```
admin> read conn jchu
CONNECTION/jchu read

admin> list session
call-filter = ""
data-filter = ""
filter-persistence = no
idle-timer = 120
ts-idle-mode = no-idle
ts-idle-timer = 120
```

2 Specify the Firewall profile name in the data-filter parameter:

```
admin> set data-filter = my-firewall
```

3 Write the Connection profile:

```
admin> write
CONNECTION/jchu written
```

See "Filter persistence for firewalls" on page C-4 for information about applying a firewall as a call filter.

Applying a firewall to a LAN interface

A firewall on an Ethernet interface affects which packets are allowed to reach the Ethernet or leave the Ethernet for another interface. A firewall applied to the Ethernet interface takes effect immediately. If you change the Firewall profile definition, the changes apply as soon as you save the Firewall profile. Following is the parameter related to applying a firewall to a LAN interface, shown with its default settings:

```
ETHERNET {shelf-N slot-N N }
   filter-name= ""
```

Note: Use caution when applying a firewall to the Ethernet interface. You could inadvertently render the MAX TNT inaccessible from the local LAN.

Following is an example that applies a firewall to a local network interface:

1 Open the Ethernet profile for the interface and list its contents:

```
admin> dir ether

     8  12/11/1996 15:58:08  { shelf-1 controller 1 }
    16  12/18/1996 16:17:17  { shelf-1 slot-12 1 }
    16  12/18/1996 16:17:17  { shelf-1 slot-12 2 }
    16  12/18/1996 16:17:17  { shelf-1 slot-12 3 }
    16  12/18/1996 16:17:17  { shelf-1 slot-12 4 }

admin> read ether {1 12 1}
ETHERNET/{ shelf-1 slot-12 1 } read

admin> list
interface-address* = { shelf-1 slot-12 1 }
mac-address = 00:c0:7b:69:94:38
ether-if-type = utp
filter-name = ""
```

2 Specify the Firewall profile name in the filter-name parameter:

```
admin> set filter-name = my-firewall
```

3 Write the Ethernet profile:

```
admin> write
ETHERNET/{ shelf-1 slot-12 1 } written
```

Index

A

B

C

D

E

F

G

H

I

L

M

N

O

P

Q

R

S

T

U

V

W

Z