

Chapter 19

Transaction Packet Assembler Disassembler (TPAD)

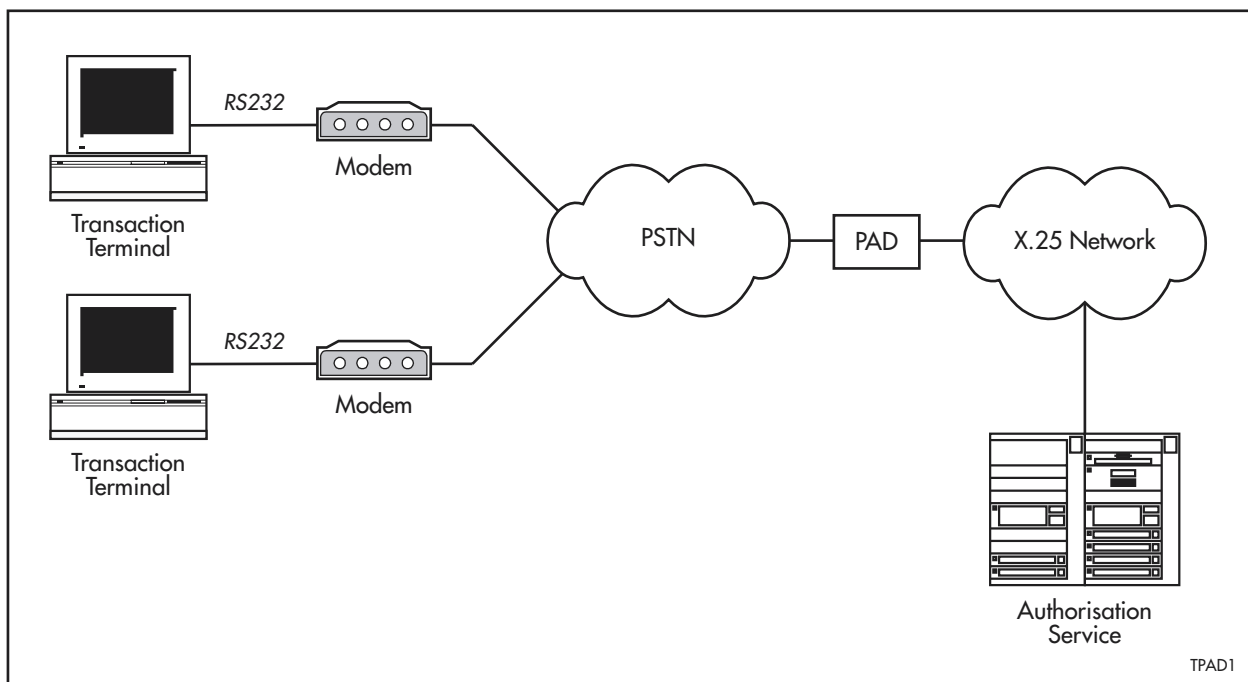
Introduction	19-2
The TPAD Protocol	19-4
TPAD on the Switch	19-5
TPAD Chip and PIN Authentication	19-7
The Hayes Standard AT Command Set on the Switch	19-8
TPAD Fast Disconnect	19-9
Configuration Examples	19-10
Configuring X.25 Interfaces	19-10
Creating TPAD Instances	19-12
Troubleshooting TPAD	19-14
Command Reference	19-17
add tpad	19-17
create tpad	19-18
delete tpad	19-20
destroy tpad	19-20
disable tpad	19-21
disable tpad debug	19-21
enable tpad	19-22
enable tpad debug	19-22
reset tpad	19-23
set tpad	19-24
show tpad	19-26
show tpad connections	19-29
show tpad counter	19-31

Introduction

The *Transaction Packet Assembler/Disassembler Protocol* (TPAD) provides a method for exchanging credit card transaction information between a transaction terminal (a swipe card machine or PC based emulation) or back office server and a credit card authorisation service. TPAD is defined by APACS (*Association for Payment Clearing Services*) Standards 30 and 40.

In a typical retail environment, transaction terminals are connected via the local PSTN to an X.25 PAD (*Packet Assembler/Disassembler*) that encapsulates the transaction data in X.25 packets and makes an X.25 connection to the authorisation service ([Figure 19-1 on page 19-2](#)). Each transaction terminal requires a separate modem and analog PSTN line, and response times are typically 25–30 seconds. Ethernet-based systems are not supported.

Figure 19-1: Traditional retail environment with asynchronous TPAD connections.

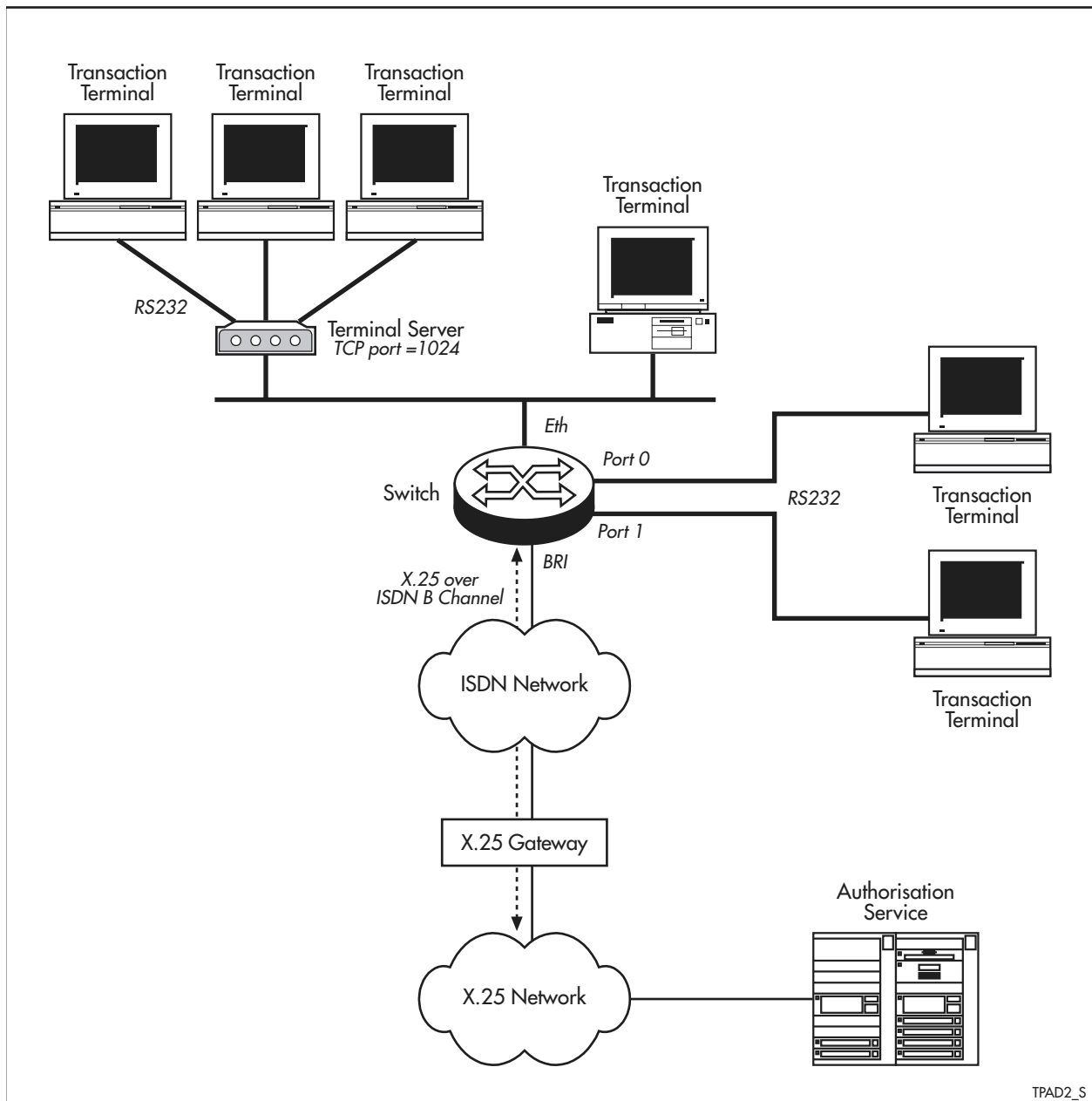


The switch's implementation of TPAD enhances the authorisation process in two key areas ([Figure 19-2 on page 19-3](#)):

- The PAD process is incorporated within the switch, and the switch connects to the X.25 network using ISDN. A single Basic Rate ISDN connection with two 64 kbps B channels can support a virtually unlimited number of transaction terminals, and response times are reduced to 5–7 seconds. On a switch with voice ports, one B channel can be reserved for credit card authorisations and the other for telephone and facsimile services.

- TPAD in the switch accepts connections from transaction terminals via both asynchronous ports and TCP connections to a specified TCP port. TPAD data received via the TCP port is extracted from the TCP packet and processed as if it was received via an asynchronous port. This enables the switch to support an unlimited number of both asynchronous and Ethernet-based transaction terminals. Asynchronous terminals can be connected directly to the asynchronous ports on the switch or to a terminal server that can be configured to make Telnet connections to a specified IP address and TCP port, providing a scalable solution.

Figure 19-2: Switch-based TPAD in a retail environment.



The TPAD Protocol

The TPAD protocol, defined by APACS 30 and APACS 40 standards, enables credit card authorisation transactions between a transaction terminal in a retail environment and an authorisation service. The protocol is an ASCII protocol designed for serial communications devices. Each transaction is a sequence of message exchanges, and each message is simple string of ASCII characters. In a traditional retail environment, transaction terminals use modems to connect via the local PSTN to an X.25 PAD. The PAD encapsulates the TPAD messages in X.25 packets and forwards them over the X.25 network to the authorisation service. A typical transaction proceeds in five distinct phases:

1. Connection of the transaction terminal to the TPAD.

The transaction terminal establishes a physical connection to the TPAD by raising carrier to the TPAD or dialling the TPAD modem, depending on whether the terminal is connected directly or via a modem.

2. Connection via the TPAD to the authorisation host.

Once the physical connection from the transaction terminal to the TPAD has been established, the transaction terminal makes a call to the authorisation service by transmitting a *Call Request* of the form:

Anua-number<CR>

or

Anua-number:nui-string<CR>

where *nua-number* and *nui-string* are the authorisation host's NUA (*Network User Address*) and NUI (*Network User Information*), respectively, and <CR> is the Carriage Return character (ASCII 13).

When the TPAD receives a valid *Call Request*, it sends a Line Feed character (ASCII 10) to the terminal as acknowledgement and then transmits an X.25 *Call Request* to the authorisation service. When the TPAD receives an X.25 Call Accept from the X.25 network it transmits an Enquiry character (ASCII 5) to the terminal to confirm that the call has been connected.

3. Data transfer.

Credit card authorisation transactions are now exchanged between the terminal and authorisation service via the TPAD. Transaction requests and responses have the same format:

<STX>*transaction-data*<ETX> [*lrc*]

where <STX> is the Start of Text character (ASCII 2), *transaction-data* is the data, <ETX> is the End of Text character (ASCII 3) and [*lrc*] is an optional *Longitudinal Redundancy Check* (LRC) character for error checking.

When the TPAD receives a valid transaction from the terminal, it transmits an Acknowledgement character (ASCII 6) back to the terminal to acknowledge receipt and then forwards the data to the authorisation service as an X.25 packet. The authorisation service transmits a transaction response as an X.25 packet that the TPAD forwards to the terminal as a TPAD message. When the terminal receives a valid transaction response, it transmits the Acknowledgement character (ASCII 6) back to the TPAD to acknowledge receipt.

4. Call clearing.

The transaction terminal clears the call by transmitting the sequence

<DLE><EOT>

to the TPAD, where <DLE> is the Data Link Escape character (ASCII 16) and <EOT> is the End of Transmission character (ASCII 4). The TPAD transmits an *X.25 Clear Request* to the authorisation service.

5. Disconnecting the transaction terminal.

The physical connection between the TPAD and the transaction terminal is cleared by the TPAD lowering the DTR and RTS signals to the terminal or modem.

TPAD on the Switch

The implementation of TPAD on the switch complies with APACS 30 and APACS 40, and also provides two significant enhancements. TPAD connects directly to the X.25 network using ISDN to significantly reduce response times. Connections are accepted from transaction terminals via both asynchronous ports and TCP connections to a nominated TCP port. Serial transaction terminals can be connected to terminal servers that support Telnet connections to specific IP addresses and TCP ports, providing a scalable solution for future growth in the retail environment.

The connection to the X.25 network can be made over a synchronous interface, the D channel of a Basic Rate ISDN service (if the ISDN service supports this feature), or the B channels of a Basic Rate ISDN service. For synchronous connections, this requires configuring LAPB on the synchronous interface, defining a set of X.25 call parameters and creating an X.25 interface over LAPB:

```
create lapb=lapb-interface over=synn dod=ON [options...]
add x25t cpar=call-index nui=nui [options...]

create x25t=x25-interface over=lapbn defcpar=call-index
dteaddress=dteaddress [options...]
```

For X.25 over the D channel of a Basic Rate ISDN service, this requires configuring LAPD, defining a set of X.25 call parameters and creating an X.25 interface over LAPD:

```
set lapd=interface mode=automatic
add x25t cpar=call-index nui=nui [options...]

create x25t=x25-interface over=lapdn defcpar=call-index
dteaddress=dteaddress [options...]
```

For X.25 over the B channel of a Basic Rate ISDN service, this requires creating an ISDN call definition, configuring LAPB to use the ISDN call as a physical interface, defining a set of X.25 call parameters, and creating an X.25 interface over LAPB:

```
set lapd=interface mode=automatic

add isdn call=name number=number precedence=out
    callingnumber=number [options...]

create lapb=lapb-interface over=isdn-call dod=on [options...]

add x25t cpar=call-index nui=nui [options...]

create x25t=x25-interface over=lapbn defcpar=call-index
    dteaddress=dteaddress [options...]
```

A TPAD instance is created or destroyed by using the commands:

```
create tpad=name over=x25-interface
    [callretry=number-retries] [checksum={on|off|yes|no}]
    [fastdisconnect={on|off}] [isdncall=isdncall-name]
    [maxconnections=connections] [maxtransfers=transfers]
    [nocarrierresponse={on|off}] [silentretry={on|off}]
    [source={async|tcp|all}] [tcpport=port]
    [transfertype={multithread|multivc|normal|transparent}]

destroy tpad=name
```

where *name* is a manager-assigned name for the TPAD instance, and *x25-interface* is the X.25 interface used by the TPAD instance to communicate with the authorisation service.

The **source** parameter determines the sources of local connections to the TPAD. If **async** or **all** is specified, connections are allowed from terminals connected to asynchronous ports on the switch.

The asynchronous ports must be assigned to the TPAD instance by using the command:

```
add tpad=name asyn=port-number
```

Asynchronous ports that are assigned to a TPAD instance cannot be used by other TPAD instances or other facilities such as Asynchronous Call Control or LPD.

Ports can be removed from the TPAD instance by using the command:

```
delete tpad=name asyn=port-number
```

If **tcp** or **all** is specified, connections to the specified TCP port are allowed. The **tcpport** parameter is required and must specify the TCP port to be used. TPAD data received via a TCP port is extracted from the TCP packet and then processed as if it were received directly via an asynchronous port.

An existing TPAD instance can be modified by using the command:

```
set tpad=name [callretry=number-retries]
    [checksum={on|off|yes|no}] [isdncall=isdncall-name]
    [fastdisconnect={on|off}] [maxconnections=connections]
    [maxtransfers=transfers] [nocarrierresponse={on|off}]
    [silentretry={on|off}] [tcpport=port]
    [transfertype={multithread|multivc|normal|transparent}]
```

A TPAD instance is disabled by default when it is created. A TPAD instance can be enabled or disabled by using the commands:

```
enable tpad=name

disable tpad=name
```

A TPAD instance that is temporarily disabled does not accept local connections or process TPAD messages.

A TPAD instance can be reset, clearing all active calls and setting all counters to zero (0) by using the command:

```
reset tpad=name
```

In addition to the extensive debugging facilities provided by the ISDN and X.25 software on the switch, TPAD-specific debugging can be enabled or disabled on a per-instance basis by using the commands:

```
enable tpad=name debug
```

```
disable tpad=name debug
```

The status and configuration of the TPAD server and all TPAD instances can be monitored by using the command:

```
show tpad[=name]
```

Activity on each TPAD instance and local port can be monitored by using the commands:

```
show tpad[=name] counter
```

```
show tpad[=name] connections
```

TPAD Chip and PIN Authentication

The switch implementation of the TPAD standard also supports the “chip on card” verification method, which means that a chip is inserted into EFT-POS cards and it is the chip that authenticates the user’s PIN for the account, without having to contact the bank’s authorisation service.

The switch plays a small part in the transaction. It is not involved in the PIN authentication between the till and the chip on the card, but it does verify the size of the transaction packets sent to the authorisation service. When the chip on a card authenticates the PIN, the packets carrying transaction data via the switch to the authorisation service are longer. The TPAD instance uses an X.25 interface to communicate with the authorisation service. To provide faster transaction speeds for these longer packets, change the **maxdata** parameter for the X.25 call from the default value of 128 bytes to 512 using the [add x25t cpar command on page 14-44](#) or the [set x25t cpar command on page 14-72 of Chapter 14, X.25](#).

The Hayes Standard AT Command Set on the Switch

A partial list of the Hayes Standard AT Command Set, or AT Command Set, is supported on the switch. AT Commands are used to control switch operations.

While the switch is in command mode, the transaction terminal may communicate directly with the switch using AT commands, which the switch executes. The switch enters command mode when:

- It powers up.
- A connection is terminated.
- It executes a command other than a dial command.
- It receives the escape sequence, which consists of three consecutive “+++” characters, while the switch is in data mode.

The format for entering AT Commands from the transaction terminal or server is *ATXn*, where:

- *AT* is the prefix that initiates each command to the switch.
- *X* is the AT command.
- *n* is the specific value for the command.

The AT commands supported by TPAD are shown in [Table 19-1 on page 19-8](#).

Table 19-1: AT Commands supported by TPAD

Command	Description
AT	<i>Attention</i> : this command checks that the switch is in control mode and that it accepts Hayes AT commands, and returns OK.
+++	<i>Escape Sequence</i> : this command is the escape sequence that allows the switch to exit data mode and enter command mode.
ATDn	<i>Dial</i> : this command instructs the switch to set the ISDN number of the call to which the TPAD, X.25 and LAPB are attached. The dial string <i>n</i> can be up to 30 characters long, and specifies the telephone number. The switch returns CONNECT.
ATH0 or ATH	<i>Hook Control (also called “Hang-up”)</i> : this command instructs the switch to terminate the ISDN call specified in the ISDNCALL parameter. This disconnects the ISDN call for the TPAD instance that the ATH or ATH0 command is received on if the TPAD instance has the FASTDISCONNECT parameter set to ON. This command usually follows the escape sequence.
ATV1	<i>Result Code Format</i> : This command instructs the switch to return results codes in word (English) format. This is the default result code format.

TPAD result codes are shown in [Table 19-2](#).

Table 19-2: TPAD Result Codes

Result code	Description
OK	Acknowledges the execution of a command line.
CONNECT	Confirms that ISDN number has been specified with ATDn.
NO CARRIER	Specifies that the switch has dropped the ISDN call, or that the call has failed to connect.
ERROR	Invalid command.

TPAD Fast Disconnect

The Fast Disconnect feature enables service switching. Retailers can use service switching if they use their transaction terminal or server to authorise transactions with more than one bank, or accept more than one type of credit card, and frequently need to terminate and establish new connections. If Fast Disconnect functionality is enabled on the switch, the till user only has to wait for a timeout on the till, rather than a timeout on the till and the switch. Therefore, the till user can process more transactions with different banks in a given time period.

The AT command ATH0 (or shorter version, ATH) is used to tell the switch to drop the ISDN call that the switch is using to call the authorisation service. All commands issued by the switch are acknowledged with a result code.

To enable fast disconnect functionality, use the command:

```
create tpad=name over=x25-interface fastdisconnect=on
      isdncall=isdn-call-name [other-options...]
```

To modify the fast disconnect configuration, use the command:

```
set tpad=name fastdisconnect={on|off}
   [isdncall=isdn-call-name] [other-options...]
```

- *name* is a character string 1 to 15 characters long. Valid characters are any printable characters.
- *x25-interface* is the number of an X.25 DTE logical interface from 0 to 7.
- *isdn-call-name* is the name of the ISDN call.

To display fast disconnect configuration information, use the commands:

```
show tpad[=name]
show tpad[=name] counter
```

Configuration Examples

The following examples illustrate how to configure TPAD on the switch. The basic steps are:

1. [Configuring X.25 Interfaces](#) that TPAD uses.
2. [Creating TPAD Instances](#) to use the X.25 interfaces.

Configuring X.25 Interfaces

TPAD uses an X.25 interface to communicate with the authentication service. Three different X.25 configurations are supported:

1. X.25 over a synchronous interface. This may be appropriate for situations where the X.25 service provider supports multiple SVCs.
2. X.25 over the ISDN D channel. This scenario is appropriate where the level of TPAD traffic is low (the D channel has a bandwidth of only 16 Kbps) and the B channels are required for data or voice traffic. Note, however, that X.25 over the D channel is not supported by all ISDN service providers.
3. X.25 over the ISDN B channel. This scenario is appropriate where the level of TPAD traffic is too high to be supported by the D channel, or where the ISDN service provider does not support X.25 over the D channel. B channels can be reserved for voice and data traffic to ensure there are always channels available for transactions as well as telephony services.

The following examples illustrate how to configure an X.25 interface, `x25t0`, using each of these options.

To create an X.25 interface over a synchronous interface

1. **Configure LAPB over the synchronous interface.**

Create LAPB interface 0 over synchronous interface 0 and enable dial-on-demand.

```
create lapb=0 over=syn0
```

2. **Create a set of call parameters for the X.25 interface.**

A call parameter set specifies characteristics of the calls made by the X.25 interface. In particular, if the transaction terminals are not capable of specifying the X.25 NUI in the TPAD call setup message, the NUI can be specified here and TPAD automatically inserts it in the X.25 Call Request. The **maxdata** parameter is optional, but setting it to 512 or more speeds up transactions when chip and PIN authentication is used:

```
add x25t cpar=1 nui=bt_host_1 maxdata=512
```

3. **Create the X.25 interface.**

Create an X.25 interface over the LAPB interface and assign the call parameter set. Other X.25 parameters may also need to be set; in particular the switch's DTE address and the channels to be used for calls may need to be specified to match those provided by the X.25 network:

```
create x25t=0 over=lapb0 defcpar=1 dteaddress=9876543  
ltc=1024 htc=1056 npvc=0
```

To create an X.25 interface over the ISDN D channel

1. Configure LAPD.

A LAPD interface is automatically created for each ISDN interface on the switch. LAPD interfaces are numbered starting from zero (0) and incrementing for each additional ISDN interface discovered during startup. For this example, the switch has a single Basic Rate ISDN interface. Set the LAPD mode to **automatic**:

2. Create a set of call parameters for the X.25 interface.

A call parameter set specifies characteristics of the calls made by the X.25 interface. In particular, if the transaction terminals are not capable of specifying the X.25 NUI in the TPAD call setup message, the NUI can be specified here and TPAD automatically inserts it in the X.25 Call Request. The **maxdata** parameter is optional, but setting it to 512 or more speeds up transactions when chip and PIN authentication is used:

```
add x25t cpar=1 nui=bt_host_1 maxdata=512
```

3. Create the X.25 interface.

Create an X.25 interface over the LAPD interface and assign the call parameter set. Other X.25 parameters may also need to be set; in particular the channels to be used for calls may need to be specified to match those provided by the X.25 network:

```
create x25t=0 over=lapd0 defcpar=1 dteaddress=9876543  
ltc=1024 htc=1056 npv=0
```

To create an X.25 interface over the ISDN B channel

1. Configure LAPD.

A LAPD interface is automatically created for each ISDN interface on the switch. LAPD interfaces are numbered starting from zero (0) and incrementing for each additional ISDN interface discovered during startup. For this example, the switch has a single Basic Rate ISDN interface. Set the LAPD mode to **automatic**:

```
set lapd=0 mode=automatic
```

2. Create an ISDN call definition.

The X.25 interface uses an ISDN call as its physical interface. When an X.25 call is made, and ISDN call is made over one of the B channels on the Basic Rate ISDN interface to carry the X.25 call. The important parameters are the calling number, called number, and precedence:

```
add isdn call=btb number=5552000 precedence=out  
callingnumber=5551000
```

3. Configure LAPB over the ISDN call.

Create LAPB interface 0 over the ISDN call and enable dial-on-demand.

```
create lapb=0 over=isdn-btb dod=on
```

4. Create a set of call parameters for the X.25 interface.

A call parameter set specifies characteristics of the calls made by the X.25 interface. In particular, if the transaction terminals are not capable of specifying the X.25 NUI in the TPAD call setup message, the NUI can be specified here and TPAD automatically inserts it in the X.25 Call Request. The **maxdata** parameter is optional, but setting it to 512 or more speeds up transactions when chip and PIN authentication is used:

```
add x25t cpar=1 nui=bt_host_1 maxdata=512
```

5. Create the X.25 interface.

Create an X.25 interface over the LAPB interface and assign the call parameter set. Other X.25 parameters may also need to be set; in particular the channels to be used for calls may need to be specified to match those provided by the X.25 network:

```
create x25t=0 over=lapb0 defcpar=1 dteaddress=9876543
lrc=1024 htc=1056 npvc=0
```

Creating TPAD Instances

Once an X.25 interface has been created, a TPAD instance must be created and configured to use the X.25 interface to communicate with the authentication service. The following examples assume that X.25 interface `x25t0` has been created as shown in [“Configuring X.25 Interfaces” on page 19-10](#). The first example illustrates the steps required to configure a basic TPAD system that accepts connections from transaction terminals via asynchronous ports on the switch. The second example extends the first example by adding support for TCP connections.

To create a TPAD instance that accepts asynchronous connections

1. Enable TPAD.

To enable the TPAD server on the switch, use the command:

```
enable tpad
```

2. Create a TPAD instance.

Create a TPAD instance named “yetanotherbank” to use the X.25 interface `x25t0`. Enable LRC checksums, set the idle timer so that X.25 calls are held open for 30 seconds pending new local connections, and limit the number of simultaneous connections to two:

```
create tpad=yetanotherbank over=x25T0 checksum=on
maxconn=2
```

3. Add asynchronous ports.

Local connections are allowed on ports 0 and 1. Limit connections to asynchronous ports and assign ports 0 and 1:

```
set tpad=yetanotherbank source=async
add tpad=yetanotherbank asyn=0
add tpad=yetanotherbank asyn=1
```

4. Enable the TPAD instance and check the configuration.

To enable the TPAD instance, use the command:

```
enable tpad=yetanotherbank
```

To check the configuration of the TPAD instance, use the command:

```
show tpad=yetanotherbank
```

To monitor activity, use the commands:

```
show tpad=yetanotherbank counter
```

```
show tpad=yetanotherbank connections
```

To add support for TCP connections

1. Enable IP and assign an IP address to the Ethernet interface.

The IP routing module must be enabled to accept TCP/IP connections. TCP connections are made from devices on the local LAN so the switch's Ethernet interface must be assigned an IP address:

```
enable ip
```

```
add ip interface=vlan1 ipadd=192.168.11.5
```

2. Enable TCP connections and set the TCP port.

To enable connections from both the currently assigned asynchronous ports and TCP connections to TCP port 1025, use the command:

```
set tpad=yetanotherbank source=all tcpport=1025
```

3. Adjust connection parameters.

Now that connections are accepted via TCP, there can potentially be a significant increase in the number of local connections and hence X.25 calls. Increase the maximum number of simultaneous connections and transactions supported:

```
set tpad=yetanotherbank transfertype=multithread
```

```
set tpad=yetanotherbank maxconn=8 maxtrans=64
```

4. Configure the transaction terminals.

Both Ethernet-capable transaction terminals and serial transaction terminals connected to terminal servers are now supported. The Ethernet capable transaction terminal or terminal server must be configured to make Telnet or TCP connections to the IP address 192.168.11.5 and TCP port 1025. Refer to the documentation for the transaction terminal or terminal server for instructions.

5. Check the configuration.

To check the configuration of the TPAD instance, use the command:

```
show tpad=yetanotherbank
```

To monitor activity, use the commands:

```
show tpad=yetanotherbank counter
```

```
show tpad=yetanotherbank connections
```

Troubleshooting TPAD

X.25 and/or TPAD configuration errors can lead to strange results. The following are some typical configuration errors and the remedial action required to correct the problem.

Missing or Invalid X.25 NUI

Figure 19-3 is a sample output from the **show log** command on page 61-36 of Chapter 61, **Logging Facility** and **show tpad counter** command on page 19-31 commands showing evidence (in bold) of a missing or invalid X.25 NUI in the call packet.

Figure 19-3: Diagnosing a missing or invalid X.25 NUI in a TPAD configuration.

Date/Time	S	Mod	Type	SType	Message

13 16:21:49	3	LOG			LAPD, ALL TEIs on BRI0 removed by network.
13 16:21:50	3	ICC	CALL	UP	ISDN call ACTIVE, direction OUT, channel B134.
13 16:21:59	3	TLNT	AUTH	OK	Telnet connection accepted from 172.16.2.43 (TTY 18)
13 16:22:03	3	USER	USER	LON	manager login on TTY18
13 16:22:07	3	TPAD	TPAD	CONN	Port 36373791 Connected to Tpad 0
13 16:22:07	3	TPAD			TCP port connect 36373791 IP 172.16.2.43
13 16:22:22	3	TPAD	CALL		Port 36373791 MultiThread Call Issued NUA - 13
13 16:22:22	4	TPAD	TPAD	DOWN	Port 36373791 Call Cleared by Network
13 16:22:28	3	TPAD	TPAD	DISC	Port 36373791 Disconnected from Tpad 0

Name	ASYN/TCP		OutData		InCallConfs
	Checksum	Errors	InData	OutCalls	NetworkClears

btca	1199		0	0	1
		0	0	1	0

The distinctive feature is the instantaneous X.25 clear generated from the network (recorded in the log) and incrementing of the *NetworkClears* counter. The X.25 NUI must be specified by the transaction terminal or set in the X25T CPAR parameters.

To resolve the problem, set the X.25 NUI correctly either in the transaction terminal or in the X25T CPAR on the switch. To set the CPAR parameters in the switch, use a command such as:

```
add x25t cpar=call-index nui=nui
```

where *call-index* is the index of the call parameter definition associated with the X.25 interface on the switch, and *nui* is the NUI to use.

Invalid X.25 Logical Channel Number

The following figure is a sample output from the [show log command on page 61-36 of Chapter 61, Logging Facility](#), the [show tpad counter command on page 19-31](#), and [show tpad connections command on page 19-29](#) that show evidence (in bold) of an invalid X.25 logical channel number in the call packet.

Figure 19-4: Diagnosing an invalid X.25 logical channel number in a TPAD configuration.

Date/Time	S	Mod	Type	SType	Message

13 16:24:17	3	LOG			LAPD, ALL TEIs on BRI0 removed by network.
13 16:24:18	3	ICC	CALL	UP	ISDN call ACTIVE, direction OUT, channel B148.
13 16:24:24	3	TLNT	AUTH	OK	Telnet connection accepted from 172.16.2.43 (TTY 18)
13 16:24:27	3	USER	USER	LON	manager login on TTY18
13 16:24:34	3	TPAD	TPAD	CONN	Port 36373793 Connected to Tpad 0
13 16:24:34	3	TPAD			TCP port connect 36373793 IP 172.16.2.43
13 16:24:39	3	TPAD	CALL		Port 36373793 MultiThread Call Issued NUA - 1
13 16:34:02	3	TPAD	TPAD	DOWN	Port 36373793 Call Cleared by User
13 16:35:02	3	TPAD	TPAD	DISC	Port 36373793 Disconnected from Tpad 0

Name	ASYN/TCP		OutData	InCallConfs	NetworkClears
	Checksum Errors		InData	OutCalls	UserClears

btca	1199		0	0	0
	0		0	1	0

Name	ASYN/TCP	Trans. Type	State	InProgress	

btca	1199	Multithread	Connecting	000	

The distinctive feature is the lack of response from the X.25 network. The TPAD counters indicate a call being issued (*OutCalls*) but not answered (*InCallConfs*). The TPAD *State* also shows “Connecting” until the call is cleared by the user equipment. To correct this problem ensure that the X.25 logical channel numbers specified on the X25T module are correct.

Invalid TPAD Checksum

The following figure is a sample output from the [show log command on page 61-36 of Chapter 61, Logging Facility](#), and [show tpad connections command on page 19-29](#) that show evidence (in bold) of an invalid checksum.

Figure 19-5: Diagnosing an invalid X.25 logical channel number in a TPAD configuration

Date/Time	S	Mod	Type	S	Type	Message

13 16:25:57	3	LOG				LAPD, ALL TEIs on BRI0 removed by network.
13 16:25:58	3	ICC	CALL	UP		ISDN call ACTIVE, direction OUT, channel B141.
13 16:26:03	3	TLNT	AUTH	OK		Telnet connection accepted from 172.16.2.43 (TTY 18)
13 16:26:10	3	USER	USER	LON		manager login on TTY18
13 16:26:19	3	TPAD	TPAD	CONN		Port 36373795 Connected to Tpad 0
13 16:26:19	3	TPAD				TCP port connect 36373795 IP 172.16.2.43
13 16:26:32	3	TPAD	CALL			Port 36373795 MultiThread Call Issued NUA - 1
13 16:26:32	3	TPAD	TPAD	ACT		Port 36373795 Call Confirm
13 16:26:41	4	TPAD				Port 36373795 BAD Data Checksum
13 16:26:41	5	TPAD				Port 36373795 BAD Data - NAK Sent
13 16:26:45	3	TPAD	TPAD	DOWN		Port 36373795 Call Cleared by User
13 16:26:46	3	TPAD	TPAD	DISC		Port 36373795 Disconnected from Tpad 0

Name	ASYN/TCP			OutData	InCallConfs	NetworkClears
	Checksum Errors			InData	OutCalls	UserClears

btca	1205			0	1	0
	1			0	1	0

Name	ASYN/TCP	OutData	InCallConfs	NetworkClears
	Checksum Errors	InData	OutCalls	UserClears

btca	1205	0	1	0
	1	0	1	0

The distinctive feature is the “Bad Data Checksum” entry in the log and an increment in the *Checksum Errors* counter. This problem can be corrected by toggling the checksum setting in TPAD by using the commands:

```
set tpad=name checksum=off
set tpad=name checksum=on
```

where *name* is the manager-assigned name of the TPAD instance.

Command Reference

This section describes the commands available on the switch to enable, configure, control and monitor TPAD. TPAD requires that X.25 be correctly configured and enabled. See [Chapter 14, X.25](#) for a detailed description of the commands required to configure and enable X.25. If X.25 connections are to be made over ISDN then ISDN call definitions must be created. See [Chapter 12, Integrated Services Digital Network \(ISDN\)](#) for a detailed description of the commands required to create ISDN call definitions. If transactions are to be accepted via TCP connections, then TCP/IP must be correctly configured and enabled. See [Chapter 23, Internet Protocol \(IP\)](#) for a detailed description of the commands required to configure and enable TCP/IP.

The shortest valid command is denoted by capital letters in the Syntax section. See “[Conventions](#)” on page lxvi of [About this Software Reference](#) in the front of this manual for details of the conventions used to describe command syntax. See [Appendix A, Messages](#) for a complete list of messages and their meanings.

add tpad

Syntax `ADD TPAD=name ASYn=port-number`

where:

- *name* is a character string 1 to 15 characters long. Valid characters are any printable characters.
- *port-number* is the number of one of the switch's asynchronous ports. Ports are numbered sequentially starting with port 0.

Description This command adds an asynchronous port to the specified TPAD instance.

The **tpad** parameter specifies the name of the TPAD instance. The specified TPAD instance must already exist.

The **asyn** parameter specifies the asynchronous port to be added to the TPAD instance. The port must not be assigned to another TPAD instance or any other modules (such as Asynchronous Call Control). If the TPD instance is used exclusively for TCP connections, then a port does not need to be added.

Examples To add asynchronous ports 2 and 3 to the TPAD instance “yetanotherbank”, use the commands:

```
add tpad=yetanotherbank asy=2
add tpad=yetanotherbank asy=3
```

Related Commands [create tpad](#)
[delete tpad](#)
[show tpad](#)

create tpad

Syntax `CREate TPAD=name OVER=x25-interface`
`[CALLretry=number-retries] [CHECKSum={ON|OFF|YES|NO}]`
`[FASTdisconnect={ON|OFF}] [ISDNcall=isdn-call-name]`
`[MAXconnections=connections] [MAXTransfers=transfers]`
`[NOCARRIERresponse={ON|OFF}] [SILENtretry={ON|OFF}]`
`[Source={ASYNc|TCP|ALL}] [TCPport=port]`
`[Transfertype={MULTIThread|MULTIVc|NORMal|TRANSPARENT}]`

where:

- *name* is a character string 1 to 15 characters long. Valid characters are any printable characters.
- *x25-interface* is the number of an X.25 DTE logical interface from 0 to 7.
- *number-retries* is a decimal number from 1 to 10.
- *isdn-call-name* is the name of the ISDN call.
- *connections* is a decimal number from 1 to 8.
- *transfers* is a decimal number from 0 to 64.
- *port* is the number of one of the switch's asynchronous ports. Ports are numbered sequentially starting with port 0.

Description This command creates a named TPAD instance over the specified X.25 circuit.

The **tpad** parameter specifies the name of the TPAD instance. The specified TPAD instance must already exist. The TPAD instance is disabled by default when it is created, and must be explicitly enabled by using the [enable tpad command on page 19-22](#).

The **over** parameter specifies the X25T interface to be used by the TPAD. The X25T interface must already exist.

The **callretry** parameter specifies the number of retries the TPAD module issues on the LAPB call before aborting the session. If the **callretry** parameter is set to 0, the TPAD module does not retry a call. The default is 3.

The **checksum** parameter specifies whether the LRC checksum is checked and generated on TPAD transaction data. The default is **on**.

The **isdncall** parameter specifies the name of the ISDN call on which the TPAD is operating. The **isdn** parameter is needed when the fast-disconnect (ATH) functionality is required. If no ISDN call is supplied and the **fastdisconnect** parameter is supplied, an error results.

The **fastdisconnect** parameter specifies whether the TPAD instance accepts the ATH command to drop the ISDN call. If the **ISDNCALL** parameter is not supplied, the command fails. The default is **off**.

If the **fastdisconnect** attribute is set in the TPAD instance, then a terminal that issues an ATH or ATH0 command deactivates the specified ISDN call. Deactivation occurs even when other terminals or traffic are using this ISDN call.

The **maxconnections** parameter specifies the maximum number of simultaneous connections allowed via TCP or asynchronous connections. The default is **1**.

The **maxtransfers** parameter specifies the maximum number of simultaneous transaction transfers allowed on the TPAD instance, and is valid if **transfertype** is set to **multithread** or **multivc**. If **transfertype** is set to **normal**, then the limit is automatically enforced at the level set by **maxconnections** since each connection can perform only a single transaction followed by a subsequent disconnection.

The **nocarrierresponse** parameter specifies whether the TPAD instance responds to the terminal with NOCARRIER when the call comes down or fails to connect for any reason. The default is **off**.

The **silentretry** parameter specifies whether the TPAD instance retries the call without notifying the till about its actions. This means that when TPAD retries the call, the no-call failure sequence (<EOT> or "NO CARRIER") is returned until the last retry fails. The number of times TPAD retries is specified with the **callretry** parameter.

The **source** parameter specifies the type(s) of local connections accepted by the TPAD. If **async** is specified, the TPAD accepts connections via local asynchronous ports. If **tcp** is specified, the TPAD accepts connections via the specified TCP port using Telnet or TCP. If **all** is specified, the TPAD accepts connections via local asynchronous ports and the specified TCP port using Telnet or TCP. The default is **async**.

The **tcpport** parameter specifies the TCP port on which the TPAD accepts incoming TCP connections. The default is **0**, which forces the user to set the parameter to enable TCP connections.

The **transfertype** parameter specifies the types of transactions allowed by the TPAD. If **multithread** is specified, multiple transaction requests are sent down the same X.25 virtual circuit. The X.25 virtual circuit is not released until all requested transactions have received clear requests. This option copes with situations where X.25 connections are made and cleared for individual transactions. If **multivc** is specified, multiple transaction requests are sent down different X.25 virtual circuits. The X.25 virtual circuit is not released until all requested transactions have received corresponding clear requests. If **normal** is specified, an X.25 connection is made and cleared for every transaction, and only one transaction is allowed per TPAD port. If **transparent** is specified, all asynchronous data is passed to the X.25 network without validation. Data packets must be delimited with a Carriage Return character (ASCII 13). The default is **normal**.

Examples To create the TPAD instance "yetanotherbank" that accepts connections via asynchronous ports 2 and 3, and TCP connections via port 10000, use the command:

```
cre tpad=yetanotherbank so=all tcp=10000
add tpad=yetanotherbank asyn=2
add tpad=yetanotherbank asyn=3
```

Related Commands

- [add tpad](#)
- [destroy tpad](#)
- [enable tpad](#)
- [reset tpad](#)
- [set tpad](#)
- [show tpad](#)

delete tpad

Syntax `DELEte TPAD=name ASYn=port-number`

where:

- *name* is a character string 1 to 15 characters long. Valid characters are any printable characters.
- *port-number* is the number of one of the switch's asynchronous ports. Ports are numbered sequentially starting with port 0.

Description This command deletes the specified asynchronous port from the specified TPAD instance.

The **tpad** parameter specifies the name of the TPAD instance. The specified TPAD instance must already exist.

The **asyn** parameter specifies the asynchronous port to be removed from the TPAD instance. The specified port must already be assigned to the TPAD instance and must not currently be in use. The TPAD instance may be left without any local communications if the **source** parameter is set to **async** and all ports are deleted.

Examples To remove asynchronous port 2 from the TPAD instance “yetanotherbank”, use the commands:

```
del tpad=yetanotherbank asy=2
```

Related Commands [add tpad](#)
[create tpad](#)
[show tpad](#)

destroy tpad

Syntax `DESTroy TPAD=name`

where *name* is a character string 1 to 15 characters long. Valid characters are any printable characters.

Description This command destroys an existing TPAD instance. The **tpad** parameter specifies the name of the TPAD instance. The specified TPAD instance must already exist, and must not have any active X.25 connections.

Examples To destroy the TPAD instance “onebank”, use the command:

```
dest tpad=onebank
```

Related Commands [create tpad](#)
[disable tpad](#)
[reset tpad](#)
[set tpad](#)
[show tpad](#)

disable tpad

Syntax `DISable TPAD[=name]`

where *name* is a character string 1 to 15 characters long. Valid characters are any printable characters.

Description This command disables TPAD or the specified TPAD instance. If a TPAD instance is specified, the TPAD instance must already exist. All TPAD connection requests are ignored when received while TPAD or the TPAD instance is disabled. TPAD instances are disabled by default when they are created.

Examples To disable the TPAD instance “onebank”, use the command:

```
dis tpad=onebank
```

To disable all TPAD instances, use the command:

```
dis tpad
```

Related Commands [delete tpad](#)
[disable tpad debug](#)
[enable tpad](#)
[enable tpad debug](#)
[reset tpad](#)
[show tpad](#)

disable tpad debug

Syntax `DISable TPAD=name DEBug`

where *name* is a character string 1 to 15 characters long. Valid characters are any printable characters.

Description This command disables the display of debug information for the specified TPAD instance. The specified TPAD instance must already exist. Debugging information is sent to the terminal or Telnet session from which the command was entered.

Examples To disable the display of debugging information for the TPAD instance “BTCA”, use the command:

```
dis tpad=btca deb
```

Related Commands [enable tpad debug](#)
[reset tpad](#)
[show tpad](#)

enable tpad

Syntax ENABle TPAD [=name]

where *name* is a character string 1 to 15 characters long. Valid characters are any printable characters.

Description This command enables TPAD or the specified TPAD instance. If a TPAD instance is specified, the TPAD instance must already exist. Connection requests are now accepted. TPAD instances are disabled by default when created, and must be explicitly enabled by using this command.

Examples To enable the TPAD instance “onebank”, use the command:

```
ena tpad=onebank
```

To enable all TPAD instances, use the command:

```
ena tpad
```

Related Commands [create tpad](#)
[disable tpad](#)
[reset tpad](#)
[set tpad](#)
[show tpad](#)

enable tpad debug

Syntax ENABle TPAD [=name] DEBug

where *name* is a character string 1 to 15 characters long. Valid characters are any printable characters.

Description This command enables the display of debug information for the specified TPAD instance. The specified TPAD instance must already exist. Debugging information is sent to the terminal or Telnet session from which the command was entered.

Examples To enable the display of debugging information for the TPAD instance “BTCA”, use the command:

```
ena tpad=btca deb
```

Related Commands [disable tpad](#)
[disable tpad debug](#)
[enable tpad](#)
[reset tpad](#)
[set tpad](#)
[show tpad](#)

reset tpad

Syntax `RESET TPAD=name`

where *name* is a character string 1 to 15 characters long. Valid characters are any printable characters.

Description This command clears all active calls and resets counter values to zero on the specified TPAD instance. The specified TPAD instance must already exist.

Examples To reset the counters for TPAD instance “CCAuth”, use the command:

```
reset tpad=ccauth
```

Related Commands [disable tpad](#)
[enable tpad](#)
[enable tpad debug](#)
[set tpad](#)
[show tpad](#)

set tpad

Syntax SET TPAD=*name* [CALLretry=*number-retries*]
 [CHECKSum={ON|OFF|YES|NO}] [ISDncall=*isdn-call-name*]
 [FASTdisconnect={ON|OFF}] [MAXConnections=*connections*]
 [MAXTransfers=*transfers*] [NOCARRIERresponse={ON|OFF}]
 [SILEntretry={ON|OFF}] [TCPport=*port*]
 [TRAnsferType={MULTIThread|MULTIVc|NORMAl|TRANSParent}]

where:

- *name* is a character string 1 to 15 characters long. Valid characters are any printable characters.
- *isdn-call-name* is the name of the ISDN Call.
- *connections* is a decimal number from 1 to 8.
- *transfers* is a decimal number from 0 to 64.
- *number-retries* is a decimal number from 1 to 10.
- *port* is the number of one of the switch's asynchronous ports. Ports are numbered sequentially starting with port 0.

Description This command modifies the configuration of an existing TPAD instance.

The **tpad** parameter specifies the name of the TPAD instance. The specified TPAD instance must already exist.

The **callretry** parameter specifies the number of retries the TPAD module issues on the LAPB call before aborting the session. If the **callretry** parameter is set to 0, the TPAD module does not retry a call. The default is 3.

The **checksum** parameter specifies whether the LRC checksum is checked and generated on TPAD transaction data. The default is **on**.

The **isdncall** parameter specifies the name of the ISDN call that TPAD is operating on. The **isdn** parameter is needed when the fast-disconnect (ATH) functionality is required. If no ISDN call is supplied and the **fastdisconnect** parameter is supplied, an error results.

The **fastdisconnect** parameter specifies whether the TPAD instance accepts the ATH command to drop the ISDN call. If the **isdncall** parameter is not supplied, the command fails. The default is **off**.

If the **fastdisconnect** attribute is set in the TPAD instance then a terminal that issues an ATH or ATH0 command deactivates the specified ISDN call. Deactivation occurs even if other terminals or traffic are using this ISDN call.

The **maxconnections** parameter specifies the maximum number of simultaneous connections allowed via TCP or asynchronous connections. The default is 1.

The **maxtransfers** parameter specifies the maximum number of simultaneous transaction transfers allowed on the TPAD instance, and is valid if **transfertype** is set to **multithread** or **multivc**. If **transfertype** is set to **normal**, then the limit is automatically enforced at the level set by **maxconnections** since each connection can perform only a single transaction followed by a subsequent disconnection.

The **nocarrierresponse** parameter specifies whether the TPAD instance responds to the terminal with NOCARRIER if the call comes down or fails to connect for any reason. The default is **off**.

The **silentretry** parameter specifies whether the TPAD instance retries the call without notifying the till about its actions. This means that when TPAD retries the call, the no-call failure sequence (<EOT> or "NO CARRIER") is returned until the last retry fails. The number of times TPAD retries is specified with the **callretry** parameter.

The **tcpport** parameter specifies the TCP port on which the TPAD accepts incoming TCP connections. The default is **0**, which forces the user to set the parameter to enable TCP connections.

The **transfertype** parameter specifies the types of transactions allowed by the TPAD. If **multithread** is specified, multiple transaction requests are sent down the same X.25 virtual circuit. The X.25 virtual circuit is not released until all requested transactions have received corresponding clear requests. This option copes with situations where X.25 connections are made and cleared for individual transactions. If **multivc** is specified, multiple transaction requests are sent down different X.25 virtual circuits. The X.25 virtual circuit is not released until all requested transactions have received corresponding clear requests. If **normal** is specified, an X.25 connection is made and cleared for every transaction, and only one transaction is allowed per TPAD port. If **transparent** is specified, all asynchronous data is passed to the X.25 network without validation. Data packets must be delimited with a Carriage Return character (ASCII 13). The default is **normal**.

Examples To set the TPAD instance "twinbank" to retry a call 2 times and reply with "no carrier" if the LAPB call is dropped or fails to connect, use the command:

```
set tpad=twinbank call=2 noc=on
```

Related Commands

- [create tpad](#)
- [destroy tpad](#)
- [show tpad](#)

show tpad

Syntax SHow TPAD [=name]

where *name* is a character string 1 to 15 characters long. Valid characters are any printable characters.

Description This command displays the configuration of TPAD instances. If a TPAD instance is specified, configuration information for the specified TPAD instance is displayed. If a TPAD instance is not specified, configuration information for all TPAD instances is displayed ([Figure 19-6 on page 19-26](#), [Table 19-3 on page 19-27](#)).

Figure 19-6: Example output from the **show tpad** command

```
TPAD Configuration
Status ..... ENABLED
List of Instances:
-----
Onebank:
  Status ..... ENABLED
  Debug ..... NO
  CheckSum ..... ON
  Transfer Type ..... Normal
  Over ..... x25t0
  Max-Transfers ..... 8
  Max-Connections ..... 2
  Source ..... ALL
  Number retries on Call ..... 3
  TCP port ..... 10000
  No Carriers Response ..... ON
  Fast Disconnect (ATH) ..... OFF

Anotherbank:
  Status ..... ENABLED
  Debug ..... NO
  CheckSum ..... ON
  Transfer Type ..... Normal
  Over ..... x25t0
  Max-Transfers ..... 8
  Max-Connections ..... 2
  Source ..... Async
  Number retries on Call ..... 3
  TCP port ..... -
  No Carriers Response ..... ON
  Fast Disconnect (ATH) ..... ON
  ISDN Call ..... Anotherb
-----
```

Table 19-3: Parameters in the output of the **show tpad** command

Parameter	Meaning
TPAD Module Status	Whether the TPAD server is enabled or disabled.
Name	The name of a TPAD instance.
Status	Whether the TPAD instance is enabled or disabled.
Debug	Whether debugging is enabled for the TPAD instance.
Checksum	Whether LRC checksums are checked and generated.
Transfer Type	The transaction types allowed by this TPAD instance: No Transfer Multithread Multivc Normal Transparent
Over	The X.25 interface used by the TPAD instance to communicate with the authorisation service.
Max-Transfer	The maximum number of simultaneous transaction transfers allowed by the TPAD instance.
Max-Connections	The maximum number of simultaneous connections via asynchronous ports or TCP allowed by the TPAD instance.
Source	The types of local connections accepted by the TPAD instance; either None, Async, TCP, or All.
Number retries on call	Shows the number of attempts the TPAD instance makes trying to establish a call. Note that "0" means the TPAD instance does not attempt to retry a call.
TCP Port	The TCP port on which the TPAD instance accepts TCP connections, or "-" if TCP connections are not accepted. The TCP Port parameter is shown when a TCP port is configured.
No Carriers Response	Whether the nocarrier parameter of the TPAD instance is on or off.
Fast Disconnect (ATH)	Whether the TPAD instance is set to hang up its ISDN call when it receives an ATH or ATH0 command from the transaction terminal.
ISDN Call	The name of the ISDN call that the TPAD instance hangs up if it receives an ATH or ATH0 command. This is displayed if fastdisconnect is set to on .

Examples To display configuration information for the TPAD instance "twinbank", use the command:

```
sh tpad=twinbank
```

To display configuration information for all TPAD instances, use the command:

```
sh tpad
```

Related Commands

- [add tpad](#)
- [create tpad](#)
- [disable tpad](#)
- [disable tpad debug](#)
- [enable tpad](#)

enable tpad debug
set tpad
show tpad counter
show tpad connections

show tpad connections

Syntax `SHoW TPAD[=name] CONnnections`

where *name* is a character string 1 to 15 characters long. Valid characters are any printable characters.

Description This command displays information about the ports used for local connections by TPAD instances. If a TPAD instance is specified, information about the connections used by the specified TPAD instance is displayed. If a TPAD instance is not specified, information about the connections used by all TPAD instances is displayed (Figure 19-7 on page 19-29, Table 19-4 on page 19-29).

Figure 19-7: Example output from the **show tpad connection** command

Name	ASYN/TCP	Trans. Type	State	InProgress
Onebank	async-3	Multithread	Online	1
	1025	Multithread	Online	5
lloyds	async-2	Multithread	Connecting	0

Table 19-4: Parameters in the output of the **show tpad connection** command

Parameter	Meaning
Name	The name of a TPAD instance.
ASYN/TCP	An asynchronous port or TCP port on which the TPAD instance is accepting local connections.
Trans. Type	The transaction types allowed by this TPAD instance on this port: No Transfer Multithread Multivc Normal Transparent
State	The current state the port: Idle Dialing Connecting Suspend Clearing Online
InProgress	The number of transactions currently in progress for the port.

Examples To display port information for the TPAD instance “lloyds”, use the command:

```
sh tpad=lloyds con
```

To display port information for all TPAD instances, use the command:

```
sh tpad con
```

Related Commands

- [add tpad](#)
- [create tpad](#)
- [disable tpad](#)
- [disable tpad debug](#)
- [enable tpad](#)
- [enable tpad debug](#)
- [set tpad](#)
- [show tpad](#)
- [show tpad counter](#)

show tpad counter

Syntax SHow TPAD[=*name*] COUnter

where *name* is a character string 1 to 15 characters long. Valid characters are any printable characters.

Description This command displays counters for TPAD instances. If a TPAD instance is specified, counters for the specified TPAD instance are displayed. If a TPAD instance is not specified counters for all TPAD instances are displayed (Figure 19-8 on page 19-31, Table 19-5 on page 19-31).

Figure 19-8: Example output from the **show tpad counter** command

```
List of interfaces:
-----
Onebank:
  Asyn-1:
    Out-Data ..... 1
    In-Data ..... 1
    Network Clears ..... 1
    User Clears ..... 0
    Checksum Errors ..... 0
    Out-Calls ..... 1
    In-Call Confirms ..... 1
    Call Retries ..... 0
    Hang-up Commands ..... 3
    No Carrier returns ..... 3

  TcpPort-1025:
    Out-Data ..... 1
    In-Data ..... 1
    Network Clears ..... 3
    User Clears ..... 4
    Checksum Errors ..... 0
    Out-Calls ..... 1
    In-Call Confirms ..... 1
    Call Retries ..... 7
    Hang-up Commands ..... 3
    No Carrier returns ..... 4
-----
```

Table 19-5: Parameters in the output of the **show tpad counter** command

Parameter	Meaning
Name	The name of a TPAD instance.
Port/TCP	An asynchronous port or TCP port on which the TPAD instance is accepting local connections.
Out-Data	The number of TPAD data messages transmitted to the transaction terminal via the associated port.
In-Data	The number of TPAD data messages received from the transaction terminal via the associated port.
Network Clears	The number of calls initiated by transaction terminals on the associated port that were cleared by the X.25 network.

Table 19-5: Parameters in the output of the **show tpad counter** command (cont.)

Parameter	Meaning
User Clears	The number of calls cleared by the transaction terminal on the associated port.
Checksum Errors	The number of TPAD data messages received containing invalid checksums (no checksum when there should have been a checksum, or a checksum that did not match the computed value).
Out-Calls	The number of calls to the authorisation service initiated by transaction terminals on the associated port.
In-Call Confirms	The number of X.25 call confirmation messages received from the X.25 network for TPAD calls initiated by transaction terminals on the associated port.
Call Retries	The number of calls retried by the TPAD module.
Hang-up Commands	The number of modem ATH commands received on the interface.
No Carrier returns	The number of NOCARRIER messages returned to the terminal if an ISDN call is dropped or if the call fails to connect.

Examples To display the counters for the TPAD instance “twinbank”, use the command:

```
sh tpad=twinbank cou
```

To display the counters for all TPAD instances, use the command:

```
sh tpad cou
```

Related Commands

- add tpad
- create tpad
- disable tpad
- disable tpad debug
- enable tpad
- enable tpad debug
- set tpad
- show tpad
- show tpad connections