FarLinX X25 Gateway

Installation
and
User Manual

Manual Revision 1.55
For use with: FarLinX X.25 Gateway version 2.7.x

FarSite Communications Ltd
info@farsite.com
www.farsite.com

© Copyright FarSite Communications Ltd. 2007-2019

Copyright

The copyright to this manual and the software described herein is owned by FarSite Communications Limited; it may not be translated or modified without prior written permission from FarSite Communications Limited.

Disclaimer

Whilst every effort is made to ensure accuracy within this manual, FarSite Communications Limited cannot be held responsible for errors or omissions and reserves the right to revise this document without notice.

Trademarks

All trademarks and registered trademarks are acknowledged.

1 INTRODUCTION 8
1.1 XOT and X.25 Switching Operation 8
1.2 TCP-to-XOT Operation 8
1.3 TCP-to-X.25 Operation 8
1.4 POS Operation 9
1.5 X.29 Host PAD 9
1.6 Triple-X PAD 9
1.7 Maximum Session Limitation 9
1.8 Back Panel Connections 9
2 TCP-X.25 MESSAGE TRANSFER OVERVIEW 11
2.1 Character Stream 11
2.2 Message Packetisation 11
2.2.1 ETX-termination 11
2.2.2 CR-termination 11
2.3 Message Header Conversion 12
2.3.1 Cisco Record Boundary Preservation (RBP) 12
2.3.2 ISO Transport and RFC-1006 12
2.3.3 RFC-1086 12
2.3.4 OFTP 13
2.3.5 X.25 Data Switching over TCP-X.25 13
2.3.6 Other header formats 13
2.4 POS Extensions 13
2.4.1 ISO 8583 13
2.4.2 APACS Message Options 14
2.4.3 CTL-Online 14
2.4.4 Portuguese Asynchronous POS Communication protocol (SIBS) 15
2.4.5 PDHFSGW Message Type 15
3 INITIAL CONFIGURATION 16
3.1 Connecting to the Gateway 16
3.2 Changing the Password 16
3.3 JAVA Settings 17
3.4 Changing the Gateway's IP Address 19
3.5 System Date and Time 19
3.6 Log Level 19
3.7 IP Security (IPSEC) 20
3.7.1 IPSEC Summary Page 20
3.7.2 IPSEC Add/Edit entry dialog 21
3.7.3 Example configuration for three Gateways 21
4 X.25 LINE CONFIGURATION 22
4.1 Create an X.25 Line 22
4.2 Configure X.25 Line 24
4.2.1 General 24
4.2.2 Layer1 Settings 24
4.2.2.1 Layer1 Settings for Synchronous Serial Lines 24
4.2.2.2 Layer 1 Settings for E1 interfaces 25
4.2.3 Layer2 Settings 26
4.2.4 Layer3 Settings 28
4.2.5 Advanced Settings 30
4.3 X.25 Line Operation 31
4.3.1 Delete 31
4.3.2 Rename 31
4.3.3 Restart 31
4.3.4 Start 31
4.3.5 Stop 32
4.3.6 Attach to Port … 32
4.3.7 Clone … 32
4.3.8 Monitor Start/Stop 32
5 PAD 33
5.1 Overview 33
5.2 Configure Telnet PAD 33

5.3 Configure AUX Port 34
5.4 Auto Call 35
5.4.1 Autocall Enable 35
5.4.2 Destination NUA 35
5.4.3 Destination Line 35
5.4.4 Destination IP Address or Destination Host 35
5.4.5 Source NUA 35
5.4.6 Call User Data 35
5.5 Default Values for PAD Parameters 36
5.6 X.28 Command Reference 36
5.6.1 CLR – Clear a Call in Progress 37
5.6.2 INT – Generate a Call Interrupt Packet 37
5.6.3 PROF 37
5.6.4 PAR? – Display Numeric Terminal Parameters 37
5.6.5 RESET – Generate a Reset Packet 37
5.6.6 SET – Set Numeric Terminal Parameters 38
5.6.7 STAT – Examine Call Status 38
5.6.8 Initiate a Call 38
5.6.9 Terminating a telnet-based PAD session 38
5.7 X.3 PAD Parameters 38

6 XOT & X.25 SWITCHING CONFIGURATION 40
6.1 SVC Routing Table 40
6.1.1 Precise Routes 40
6.1.2 Wild-card Routes: 42
6.1.3 Default Route 42
6.1.4 Calling Address Translation 42
6.1.5 Called Address Translation 44
6.1.6 Closed User Group (CUG) 44
6.1.7 Forwarding Size 45
6.1.8 TCP Keep Alives 45
6.2 PVC Routing Table 46
6.3 XOT-to-XOT Routing 48

7 TCP-X.25 CONFIGURATION 49
7.1 X.25-to-TCP 49
7.1.1 General Settings 50
7.1.1.1 Name 50
7.1.1.2 Message Type 50
7.1.1.3 Local X.25 NUA / Source Channel 51
7.1.1.4 X.25 Line 51
7.1.1.5 Destination Host Name / IP Address 51
7.1.1.6 Destination TCP Port Number 51
7.1.1.7 Load balancing 52
7.1.2 Advanced Settings 52
7.1.2.1 Backup Target Host settings 52
7.1.2.2 Parity conversion 53
7.1.2.3 Generic TCP Message Header conversion 53
7.1.2.4 Alternative Message Termination Character 54
7.1.2.5 Construct TCP/IP Addressing From X.25 CUD 54
7.1.2.6 Forwarding Size 55
7.1.2.7 Insert Calling NUA in data header 55
7.1.2.8 Use control bits in data header 55
7.1.2.9 TCP Keep Alive 55
7.1.3 X.29 Host 56
7.2 TCP-to-X.25 57
7.2.1 General Settings 58
7.2.1.1 Name 58
7.2.1.2 Message Type 58
7.2.1.3 Local TCP Port Number 59
7.2.1.4 Local X.25 NUA 59
7.2.1.5 Destination X.25 NUA / Channel 59
7.2.1.6 X.25 Line 59
7.2.1.7 X.25 Call User Data 60
7.2.1.8 X.25 Facilities 60
7.2.1.9 Enable X.25 Backup Line 60

7.2.2 Advanced Settings 61
7.2.2.1 Parity conversion 61
7.2.2.2 PAD Routing – X.25 Call Routing Method 61
7.2.2.3 Generic TCP Message Header conversion 62
7.2.2.4 Alternative Message Termination Character 62
7.2.2.5 Forwarding Size 62
7.2.2.6 Use control bits in data header 63
7.2.2.7 TCP Keep Alive 63
7.3 RFC-1086 Settings 63
7.3.1 RFC-1086 Overview 64
7.3.2 Radius Authentication 65
7.3.3 RFC-1086 Configuration 65
7.3.3.1 Connection Retry 65
7.3.3.2 Authentication 66
7.3.3.3 Radius Configuration 66
7.4 Configuration of POS Extensions 67
7.5 TCP-to-XOT 67
7.6 Timeouts 68
7.6.1 Idle timeout 69
7.6.2 Hold timeout 69
7.6.3 TCP call 69
8 DEVELOPER REFERENCE 70
8.1 XDRPD (Extended Dynamic Routing with Packetised Data) 70
8.1.1 Message Format 70
8.1.1.1 Insert Calling NUA in data header 71
8.1.1.2 Use control bits in data header 71
8.1.2 SVC TCP-to-X.25 Operation 72
8.1.3 SVC Connection Closure initiated by TCP/IP host 73
8.1.4 PVC TCP-to-X.25 Operation 74
8.1.5 PVC X.25-to-TCP Operation 74
8.1.6 PVC Connection Closure initiated by TCP/IP host 75
8.1.7 Connection Closure initiated by Gateway 75
8.1.8 Data Transfer 75
8.1.9 Reset Packet Transfer 75
8.1.10 Sample Applications 75
8.2 DRPD (Dynamic Routing with Packetised Data) 76
8.2.1 Message Format 76
8.2.2 TCP-to-X.25 Operation 76
8.2.3 X.25-to-TCP Operation 78
8.2.4 Data Transfer 78
8.2.5 Sample Applications 78
8.3 PAD and Call Address Block Routing 78
8.3.1 Call Address Block Routing – no PID 79
8.3.2 Call Address Block Routing – X.29 PID 79
8.3.3 PAD Routing 80
8.4 Cisco Record Boundary Preservation (RBP) protocol 81
8.5 X.25 Data Switching over TCP 81
8.5.1 Packet Format over TCP 82
8.5.2 Connection Request (CR) packet 82
8.5.3 Connection Confirm (CC) packet 82
8.5.4 Disconnection Request (DR) packet 82
8.5.5 Data (DT) packet 82
8.5.6 Message Elements 83
8.5.7 Examples 83
8.6 PDHFSGW Message Type 83
8.7 Generic Header Conversion 84
8.7.1 Conv hdr 2-bin 84
8.7.2 Conv hdr 2-ascii 84
8.7.3 Conv hdr 4-bin 85
8.7.4 Conv hdr 4-ascii 85
8.7.5 Custom Message Headers 85
9 STATISTICS 86
9.1 FarLinX Summary Page Statistics 86

9.1.1 Configured Ports 87
9.2 Port Statistics 87
9.2.1 Port Statistics – X.25 88
9.2.2 Port Statistics – Errors 90
9.3 Sessions statistics 91
9.4 XOT statistics 92

10 SNMP TRAPS/ALARMS 94

11 LOGS 95
11.1 Log Configuration – General 95
11.2 Event Log 95
11.3 Transaction Log 96
11.4 Log Configuration – Detailed 96
11.4.1 X25-XOT Event Log Level 97
11.4.2 TCP-X25 Event Log Level 97
11.4.3 Debug Level 97
11.4.4 Data Trace Level 97
11.5 Logging to a syslog server 97
11.5.1 Structure of Messages Sent to syslog 98

12 X.25 MONITOR 99
12.1 Installing and Configuring the FarSync Line Monitor 99
12.2 Configure the Gateway Settings 99
12.3 Start/Stop Monitor 100
12.4 Monitoring the X.25 lines using Wireshark 101

13 HTTPS SUPPORT 103
13.1 Private key 103
13.2 Generate certificates 104
13.3 Upload certificates 104
13.4 Certificate Backup 106

14 MAINTENANCE 107
14.1 Configuration Backup and Restore 107
14.1.1 Backup 107
14.1.2 Restore 107
14.1.3 Import 107
14.2 System Update 108
14.3 Firmware Upgrade 108
14.3.1 Upgrade 108
14.3.2 Switch Back 109
14.4 Restart and Shutdown 110
14.5 System Status 110
14.6 Factory Default 110
14.7 Support Information 111

15 LIVE STANDBY GATEWAY 113
15.1 Hardware setup 113
15.2 Configuration 113
15.3 Shared IP Transfer 115

16 X.25 CAUSE AND DIAGNOSTIC CODES 116
16.1 Gateway Initiated Clearing and Resetting Reasons 116
16.1.1 DCE X.25 Line Clearing Reasons 116
16.1.2 DTE/1984 X.25 Line Clearing Reasons 116
16.1.3 DTE/1980 X.25 Line Clearing Reasons 116
16.1.4 XOT (X.25 over TCP/IP) Specific Clearing Reasons 116
16.1.5 XOT (X.25 over TCP/IP) Specific Resetting Reasons 117
16.2 X.25 Diagnostic Code Explanations 117
16.2.1 Standard X.25 Diagnostic Codes 117
16.2.2 Non-Standard but Common X.25 Diagnostic Codes 118
16.3 X.25 Cause Code Explanations 119
16.3.1 Clearing Causes 119
16.3.2 Reset Causes 119

16.3.2.1 Reset-Specific Diagnostic Codes 120
16.3.3 Restarting Causes 120

1 INTRODUCTION

This manual describes the installation, configuration and operation of the FarLinX X.25 Gateway. If you have not already done so, use the Software Identification Code provided with your Gateway to download the additional FarLinX Gateway Resources from the download area at www.farsite.com/download. Contact your reseller if you have difficulty locating this code. The FarLinX Gateway Resources download includes additional documentation, support applications, third party software and sample applications for use with the FarLinX Gateway

The FarLinX X.25 Gateway supports X.25 packet switching between X.25 and XOT (X.25 over TCP/IP), routing between TCP and XOT and data translation between X.25/XOT and TCP connections. Optionally, there can be special data translations for Point of Sale protocols.

The FarLinX X.25 Gateway can act as an access node for inter-connecting X.25 equipment across a TCP/IP network, such as the Internet. It does so using the standard XOT (X.25 over TCP/IP) protocol, as defined in RFC-1613 and so can interface to other XOT gateways, switches or hosts from other manufacturers. The FarLinX X.25 Gateway can be used to switch X.25 traffic between X.25 and XOT links. Both Switched and Permanent Virtual Circuits (SVCs and PVCs) are supported.

The FarLinX X.25 Gateway can operate as a general purpose means for connecting TCP/IP clients to X.25 hosts and also X.25 terminal equipment to TCP/IP servers. A wide variety of interconnection and data translation options are available.

The FarLinX X.25 Gateway supports PAD connections to X.25 or XOT from Async connected terminals that require a “Triple-X” PAD (X.3, X.28, X.29) interface and also from devices over the LAN that can use Telnet to connect to the Gateway’s virtual PAD.

POS (Point of Sale) support can be optionally included. This is an extension of the general purpose TCP to X.25 translation with special support for popular POS protocols. The FarLinX X.25 Gateway can inter-connect existing legacy terminals attached to an X.25 network to acquirer host systems with TCP/IP interfaces and can also interconnect TCP/IP POS terminals with legacy X.25 host systems.

1.1 XOT and X.25 Switching Operation

The Gateway provides X.25 switching capability over both XOT and physical X.25 links. An X.25 client can, via the Gateway, make a connection to a server on the remote side of an IP network, or to another server connection via a local X.25 link. The remote server can be a native XOT host, or it can be an X.25 host connected to an X.25 link attached to a remote XOT Gateway, or attached to a Cisco router, for example.

When the Gateway receives a Call Request packet, it examines its routing table, using the X.25 Called DTE Address to determine the destination for the connection; the Call is then routed out either via XOT or the local X.25 link.
Data transfer then takes place over the established virtual circuit. If either DTE clears the call, the Gateway will clear the virtual circuit to the peer.

The Gateway supports a variety of routing functions, such as matching wild cards and setting Closed User Groups. It also provides other special functions such as address translation.

1.2 TCP-to-XOT Operation

The FarLinX X.25 Gateway can support TCP routing to XOT and can thus be a convenient point on a TCP/IP network to handle the translation of TCP/IP connections to XOT where the remote host requires data in XOT format, or even possibly X.25 format where there is a further Gateway performing the XOT-to-X.25 translation at another point on the network.

1.3 TCP-to-X.25 Operation

The Gateway provides connection-oriented session establishment and data packetization services to allow applications sending data over TCP/IP to interface to X.25 connected hosts. The Gateway provides routing facilities to select the appropriate host from those available.

The Gateway listens on a number of TCP ports – one port per configured X.25 host. When a client establishes a TCP connection to the Gateway, the Gateway sets up an X.25 virtual circuit to the appropriate host.

Similarly, the Gateway can route incoming X.25 calls using the Called DTE Address (and Sub-address as appropriate) to determine the target TCP/IP host name/IP address and port number.

Once the end-to-end connection is complete, the data transfer takes place. There are a number of options as to how the data transfer/packetisation is handled.

At the end of the data transfer/s, if the terminal disconnects, the Gateway will clear the X.25 virtual circuit. Alternatively, if the host clears the X.25 virtual circuit first, the Gateway will disconnect the TCP connection to the terminal.

1.4 POS Operation

The POS (Point of Sales) extensions to the Gateway expand the general purpose TCP-X.25 operation providing specialised connection establishment and data packetisation protocol handling services for APACS, ISO 8583 and other POS protocols to allow POS terminals operating over TCP/IP to interface to X.25 connected hosts.

1.5 X.29 Host PAD

The FarLinX X.25 Gateway can be configured to act as an X.29 host PAD for remote terminals or applications connecting over X.25 and expecting a host that supports X.29. The host PAD sends X.29 PAD configuration commands over X.25 to the connecting terminals. The host PAD profile is configurable.

1.6 Triple-X PAD

Triple-X PAD (X.28, X.3 and X.29) is supported by the Gateway and is accessible over TCP/IP via Telnet or the Gateway's async port. The connecting terminal is presented with a Triple-X PAD service which can be used to make X.25 connections over the physical X.25 line(s) or via XOT connections to remote hosts. The initial PAD profiles both for the async port and for Telnet access, are fully configurable.

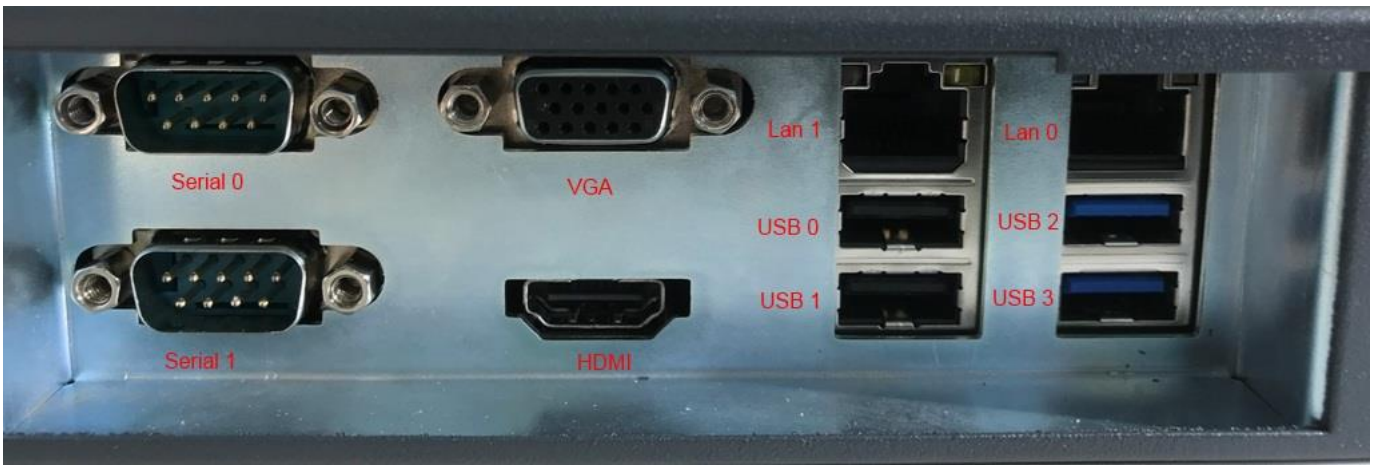
1.7 Maximum Session Limitation

The FarLinX X25 Gateway can support 4000 simultaneous sessions using an X.25 packet size of up to 512 bytes. On older Gateways fitted with only 512MB DRAM, the maximum number of sessions is 3000. Use the Support Info page to determine the size of DRAM fitted to your Gateway. The maximum number of simultaneous sessions will be lower for packet sizes over 512 bytes.

1.8 Back Panel Connections

Recent models of the FarLinX Gateway have two Ethernet ports and two serial ports, but in this release of the firmware only one of the Ethernet ports and one of the serial ports is available for use.

The following diagram shows the external connections available on the back panel.



Please connect the Ethernet cable to the port “Lan 0” and the serial port cable, if used, to the port “Serial 0”.

All other connections should only be used under instruction from the FarSite Support group.

2 TCP-X.25 MESSAGE TRANSFER OVERVIEW

Because X.25 is a packet based protocol and TCP is a stream based protocol, there can be issues when interconnecting the two network types. The FarLinX X.25 Gateway is designed to overcome these issues. Because the X.25 protocol deals with natively delineated complete packets, X.25 based applications are typically very sensitive to the receipt of complete messages. TCP applications on the other hand expect to receive a data stream with the delineation of message boundaries signalled through additional in-band structuring of the data.

There are three general types of data transfer supported by the Gateway, some with specific sub-modes:

- Character Stream
- Message Packetisation
- Message Header Conversion

In addition, the POS Extensions option can involve both Message Packetisation and Message Header Conversion – these are each described in their own sub-section below.

As well as turnkey message transfer modes, the Gateway also supports flexible modes of operation targeted at developers – these are described in detail in Section 8.

2.1 Character Stream

By default, the TCP to X.25 operation is configured for Character Stream operation – the Gateway simply forwards the blocks of data received in a TCP data stream on to the X.25 interface as they are received. In this mode, the **Forwarding Size** is configurable, allowing the TCP-based data to be forwarded to the X.25 as and when required.

Character Stream operation can work reliably for typical transaction applications, even those where the X.25 Host treats the received X.25 data unit as the entire message, provided that the TCP-connected client sends each transaction request in a single TCP transmission. This is because transaction applications are, by their very nature, half-duplex, with each request requiring a response – there is thus no scope for two messages to get concatenated together. The only potential problem would be if the message were to be split into separate TCP packets – this should never happen if the entire message is sent in one go, provided the message size is less than the TCP MTU (maximum transmission unit).

Character Stream operation is not suitable for applications when more than one message is sent, one after the other without waiting for a response, or when messages are longer than the TCP MTU – except when the X.25 application itself ignores the X.25 data boundaries.

2.2 Message Packetisation

In Message Packetisation modes, the Gateway does not perform any conversion as such; it attempts to recognise the end of each message within the TCP data stream, forwarding them over X.25 within a complete X.25 data packet sequence, thus preventing any unwanted concatenation or fragmentation of the messages and guaranteeing the synchronisation of the messages with the X.25 data unit boundaries. There are sub-modes for Message Packetisation as described in the following sections.

2.2.1 ETX-termination

This mode would be used when all messages are terminated by the ETX character – IA5 (0, 3).

2.2.2 CR-termination

This mode would be used when all messages are terminated by the Carriage Return character – IA5 (0, 13).

In this mode it is also possible to configure an alternative message termination character – refer to Section 7.2.2.4 for more details.

2.3 Message Header Conversion

Some messages formats use a header to describe the X.25 message payload over TCP connections, but the message header is not required within the X.25 data stream.

When performing conversion, the Gateway interprets the header from the received TCP payload – this header will contain the length of the message in order to delimit the end of the message, allowing the Gateway to recognise the start of the next message. The Gateway then removes the message header before forwarding the message body to the X.25 connection. In the other direction, the Gateway takes the X.25 message and constructs and inserts the appropriate message header before forwarding the entire message (header plus body) to the TCP connection.

Note that the maximum message size that can be supported for header conversion is 4096 bytes.

2.3.1 Cisco Record Boundary Preservation (RBP)

The Gateway supports the Cisco Record Boundary Preservation format – this is a header of 6 bytes in size, as follows:

Bytes	Usage	Contents
0 & 1	Version number	0xd7 0x4a
2 & 3	Payload Length	Length in Network Byte order
4	More Data flag	equivalent to X.25 M-bit
5	reserved	0x00

The use of the More Data flag allows messages of any size to be transferred (as they can be divided into conveniently sized fragments).

Cisco RBP thus allows any type of X.25 data payload to be carried over TCP/IP, but of course it does require that the equipment at the other end of the TCP connection can also support the Cisco RBP format.

2.3.2 ISO Transport and RFC-1006

The Gateway is capable of transferring ISO Transport Class 0 protocol data units, using the RFC-1006 encapsulation on the TCP connection.

The RFC-1006 header is removed before forwarding the data units over the X.25 virtual circuit, so the X.25 data payload is simply the Transport Protocol Data Unit (i.e. standard for Transport Class 0 over X.25).

This allows the Gateway to support, for example, FTAM applications without native X.25 support and to interface to servers such as Siemens EWSD switches.

Note that the Gateway can support transport classes other than class 0, but RFC-1006 is specified as being class 0 only (there being little point in using one of the other classes over TCP seeing as a TCP connection is equivalent to a Class 4 Transport connection).

2.3.3 RFC-1086

The FarLinX implements a subset of RFC-1086 whereby only one function of the Registration procedure is supported, i.e. only TCP-to-X.25 sessions can be established.

In addition, an optional Identification Frame may precede the Registration phase from which a username and a password can be extracted for Radius Authentication.

When the RCF-1086 Identification/Registration process is complete, the FarLinX Gateway will process the subsequent packets as RFC-1006.

2.3.4 OFTP

The Odette File Transfer Protocol (OFTP) is a protocol used for EDI (Electronic Data Interchange) between two communications business partners. It comes from the Odette-Organisation (Organization for data exchange by teletransmission in Europe).

When the OFTP message type is selected, the Gateway assumes that the TCP payload contains a header of 4 bytes in size, which contains the length of the message (including the header size). The Gateway will remove this header before forwarding the message over the X.25 virtual circuit and will add the header to messages received from the X.25 interface before forwarding it over the TCP connection. The 4 byte header is formatted as follows:

Bytes	Usage	Contents
0	Version number	0x10
1,2 & 3	Payload Length	Length in Network Byte order

2.3.5 X.25 Data Switching over TCP-X.25

The Gateway can switch X.25 virtual circuits over an IP network via TCP connections and then switch back to native X.25 at the remote side of the IP network. This does of course rely on there being a FarLinX Gateway at both sides of the IP network.

The only configuration required is the selection of X.25 Data Switching as the required message type, the IP address of the remote Gateway and the Port Number of the X.25 Data Switching service on the remote Gateway. For more details, see Section 8.5.

2.3.6 Other header formats

When converting generalised header formats, the Gateway can support the following:

- 2 or 4 byte headers
- Length field 2 or 4 bytes in size within header
- Length field offset within header (if length field smaller than header)
- Length field encoded in binary or ASCII
- Length includes or excludes the size of the header itself.

The Gateway provides the means of configuring these aspects of the header independently, thus allowing the Gateway to support the conversion of a wide range of different generic message headers.

It is also possible to select that the header is carried over between TCP and X.25 data payloads.

For more details, see Section 8.7

2.4 POS Extensions

These modes of message transfer require that the POS extension to the Gateway has been purchased.

2.4.1 ISO 8583

When the ISO 8583 message type is selected, the Gateway assumes that the TCP payload contains a header of 2 bytes in size, which contains the length of the message (including the header size). The Gateway will remove this header before forwarding the message over the X.25 virtual circuit and will add the header to messages received from the X.25 interface before forwarding it over the TCP connection.

There is an alternative header format for ISO 8583 supported by the Gateway, with a header of 4 bytes, plus a variant of the CTL-Online format (see below) but without parity conversion.

If the format of the message in the TCP payload is exactly the same as in the X.25 payload, ISO 8583 messages can be routed by selecting the character stream message type instead.

2.4.2 APACS Message Options

The Gateway supports APACS 30, APACS 40 and other varieties of APACS which conform to the same basic framing. In these cases the APACS messages are carried within the TCP payload without any additional headers (APACS messages can also be used with systems which employ headers, such as CTL-Online – this is covered in Section 2.4.3 below).

APACS connections over X.25 can operate in two different ways (irrespective of whether they are APACS 30 or APACS 40, etc).

TPAD-HOST Mode
APACS messages are sent without a LRC in PAD-HOST mode and no APACS control frames (ENQ, ACK, NAK, DLE EOT) are required.

T/T-TPAD (or HOST-T/T) Mode
APACS messages are sent complete with the LRC. APACS control frames (ENQ, ACK, NAK) are used to regulate the flow and termination of the APACS message, each message being acknowledged with an ACK frame. Completion of a transaction is indicated by DLE EOT.

The Gateway is capable of converting from one of these two formats into the other – this is necessary when inter-connecting terminals which implement one format with a host which implements the other.

TPAD Conversion Mode
The Gateway converts from the T/T-TPAD mode on the terminal side to the TPAD-Host mode on the Host side when TPAD Conversion is configured.

When the connection is established, the Gateway sends an ENQ to the terminal. When the terminal transmits the transaction request message, the Gateway removes the LRC, forwards the request to the host, which then responds with an ACK. When the transaction response is received from the Host, the Gateway calculates the LRC and forwards the response to the terminal. The ACK from the terminal is discarded and when the terminal sends the DLE EOT, the Gateway clears the connection to both the terminal and the host.

HPAD Conversion Mode
The Gateway can convert from the TPAD-Host mode on the terminal side to the T/T-TPAD mode on the Host side when HPAD Conversion is configured.

When the connection is established, the Gateway discards the ENQ from the host. When it receives the transaction request from the terminal, it calculates the LRC and forwards the request plus LRC to the Host. It discards the ACK from the Host and then when it receives the Response, it checks and removes the LRC and forwards the response to the terminal. It then sends an ACK to the Host. When the terminal clears the connection, the Gateway sends DLE EOT to the host.

If the Gateway receives a NAK from the Host, or if the LRC check fails, it clears the connection; the Terminal then has to re-try the transaction.
HGEPOS
In addition to the ENQ/ACK control characters and the STX-Data-ETX-LRC format used by APACS, HGEPOS also has an encrypted data format – SOH-Length-STX-Data-EOT-LRC.

2.4.3 CTL-Online

There are 2 CTL-Online message formats – they are typically used for APACS, but could be equally applicable for other message protocols involving 7-bit data.

The TCP payload contains a header, the format of which is described below and depends on which of the 2 formats is being used. The Gateway will remove this header before forwarding the message over the X.25 virtual circuit and will add the header to messages received from the X.25 interface before forwarding it over the TCP connection.

The Gateway employs parity conversion, using 7-bit Even parity. The parity bit is removed from each byte received in X.25 data units and added to data transmitted over the X.25 virtual circuit.

APACS CTL-Online Hdr-3
The TCP payload contains a header of 3 bytes in size – the first byte contains 0xff and the remaining 2-bytes contain the length of the message (excluding the header size).

APACS CTL-Online Hdr-2
The TCP payload contains a header of 2 bytes in size, which contains the length of the message (excluding the header size).

2.4.4 Portuguese Asynchronous POS Communication protocol (SIBS)

SIBS messages are supported in conjunction with PAD routing – i.e. the TCP client application supplies the call parameters.

2.4.5 PDHFSGW Message Type

This message type is used to allow connections from PDH Clients to X.25-attached ATMs. It is a form of dynamic routing, similar to XDRPD/DRPD, in that it allows the client to specify the called X.25 DTE address. For more details, see Section 8.6.

3 INITIAL CONFIGURATION

3.1 Connecting to the Gateway

The Gateway's configuration can be viewed and updated using a browser over a TCP/IP network. This requires a configuring PC running a web browser that is able to run an embedded Java Applet. The Java runtime should be version 8, as applets are no longer supported in later version of the runtime. As of the time of writing, the only browser capable of doing this is Internet Explorer version 11. You need to install a 32-bit version of the Java Runtime, we recommend version 8 Update 221. A copy of the Java Runtime and Internet Explorer 11 can be found in the FarLinX Gateway Resources download.

Note that although the Gateway supports IPv6 addressing for both configuration and operation, the Gateway must initially be configured using IPv4 addressing.

When the Gateway is switched on for the first time, the **IP address** of the Gateway defaults to having the IPv4 address **10.0.0.1** with a subnet mask of 255.0.0.0. To enable your computer to connect to a new Gateway, please configure your client PC's IP address to an address in the range 10.x.x.x, for example 10.0.0.2. Depending on your LAN addressing this may require you to make a temporary direct cable connection to the Gateway.

Once the Gateway has powered up, connect it to the local area network. Note: If you connect the Gateway with your configuring PC directly, a LAN crossover cable may be needed.

Use the browser to navigate to http://10.0.0.1/
A login dialog will then be presented. Enter the default user credentials:

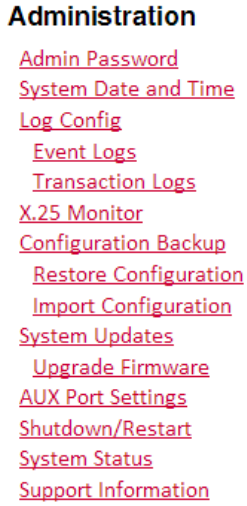
username: **admin**
password: **farlinx**

In order to help with initial management/configuration of the Gateway, ICMP Echo (Ping) Response is enabled by default. Once the Gateway is up and running you may wish to disable the ICMP Ping option for security reasons. This can be configured via the LAN configuration page.

When you have successfully logged in, there are a number of configuration changes that you should make immediately. These are detailed in the rest of this section.

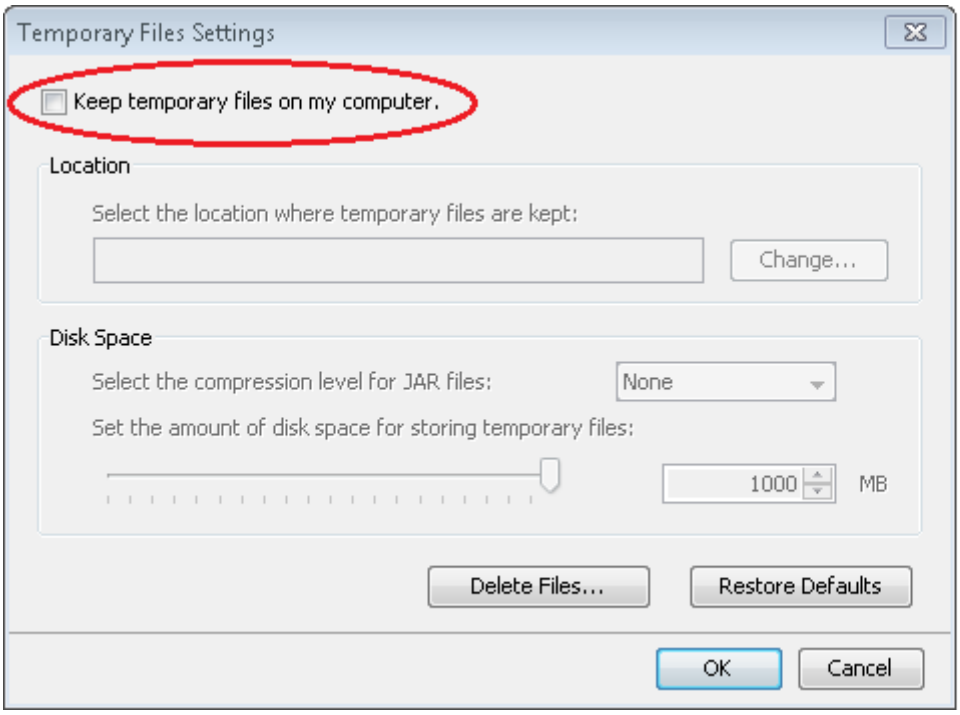
3.2 Changing the Password

To secure your Gateway, you should change the default Administrator password. Click Admin Password under **Administration** on the navigation bar. Enter the new password in the form provided. Then click the 'Save' button to apply the new password.



3.3 JAVA Settings

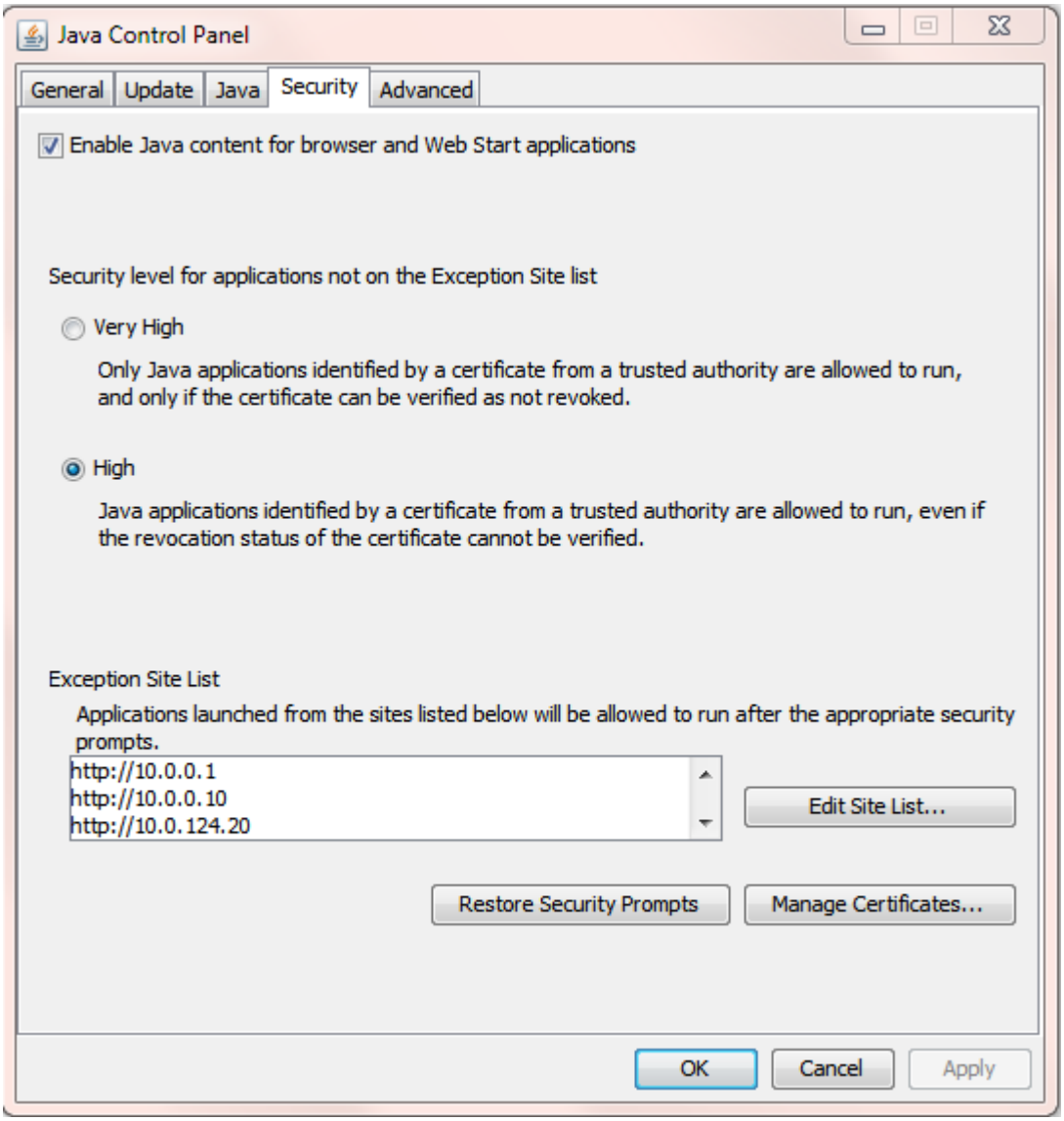
The Gateway uses a Java applet for its X.25 line configuration and management support. If you are using a Windows PC to configure the Gateway, the default JAVA runtime settings may not be suitable for the X.25/Gateway Management configuration. The Java Runtime Settings can be found in the Control Panel. Under JAVA→General →Temporary internet files Settings, it is very important that the option **"Keep temporary files on my computer"** is **unchecked**. This is an example dialog:



Select the Security tab of the Java Control Panel. Please ensure that the "Enable Java content for browser and Web Start applications" is ticked.

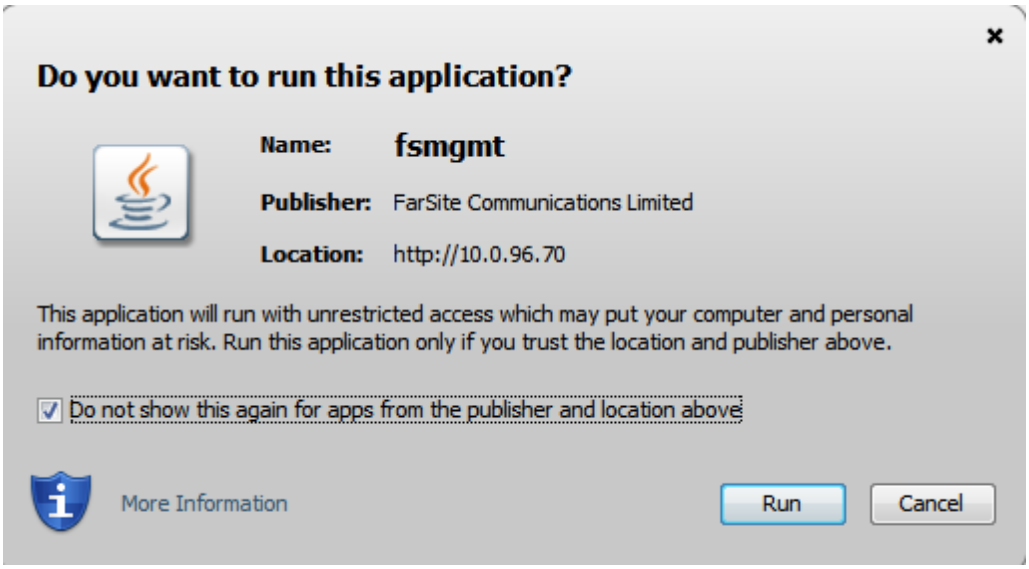
Set the Security Level to the lowest available setting, in this case it is "High".

And then add an entry under the "Exception Site List" for each IP address that may be used for configuring the Gateway. See the example below:

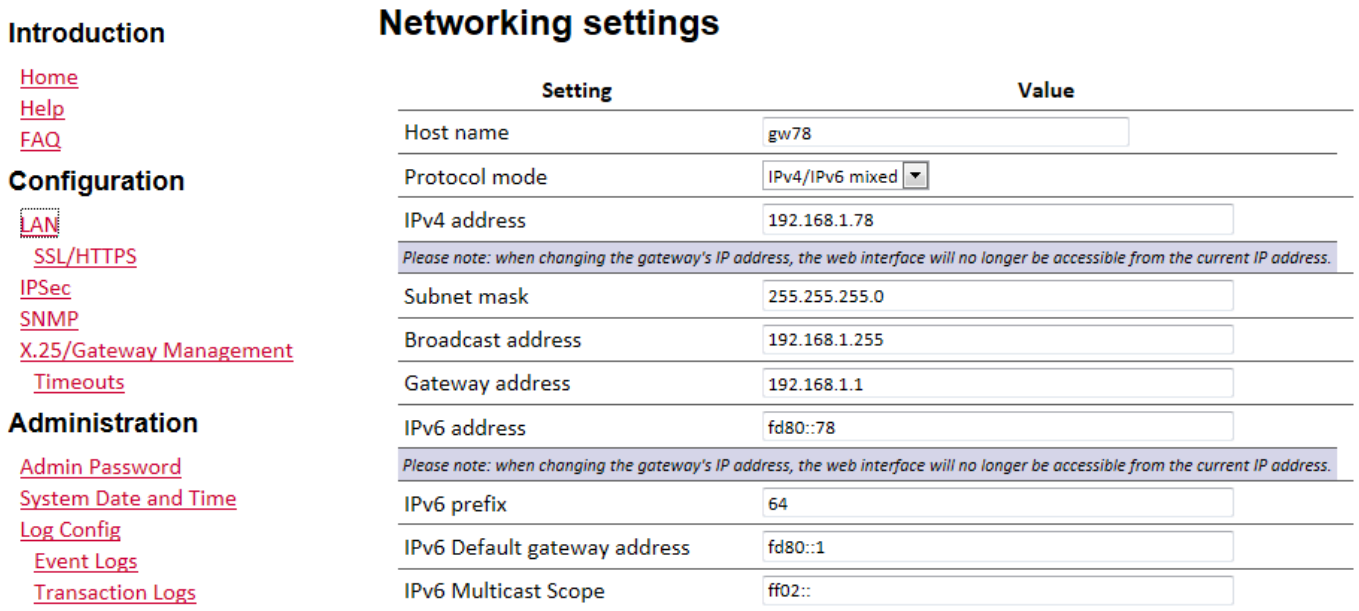


Press OK when done.

Finally, when starting the applet for the first time you will encounter a number of security prompts. Please tick the "Do not show this again" option and click "Run".



3.4 Changing the Gateway's IP Address



To change the IP address and other network settings used by the Gateway to access your local network, click LAN under **Configuration** on the navigation bar. All the network settings for the Gateway's LAN IP network interface are available in the form provided to be configured as required. Set these up as required for your network (e.g. IP address, Subnet mask etc.).

If you require the Gateway to accessible from IPv6 clients, or to use IPv6-based routes, then select the Protocol mode "IPv4/IPv6 mixed" or "IPv6 only" and configure the IPv6 address/prefix for the Gateway to use. Then click 'Save' to save and apply your modified configuration. To configure the Gateway using the IPv6 address, the format in the address bar is http://[IPv6 address].

Note: You will now need to direct your browser to select the IP address that you have just assigned to the Gateway to continue configuration. For example if you configured an IP address of 192.168.1.100 then use http://192.168.1.100 to access the configuration web interface.

3.5 System Date and Time

The Gateway will produce log files at start of each day and names the files based on the current date. The system date and time can be set automatically using SNTP, but if the Gateway is not connected to the internet, this may not be possible. In this case it is strongly recommended to set the correct system time as part of the initial configuration of the Gateway, otherwise it will cause confusion in the management of log files.

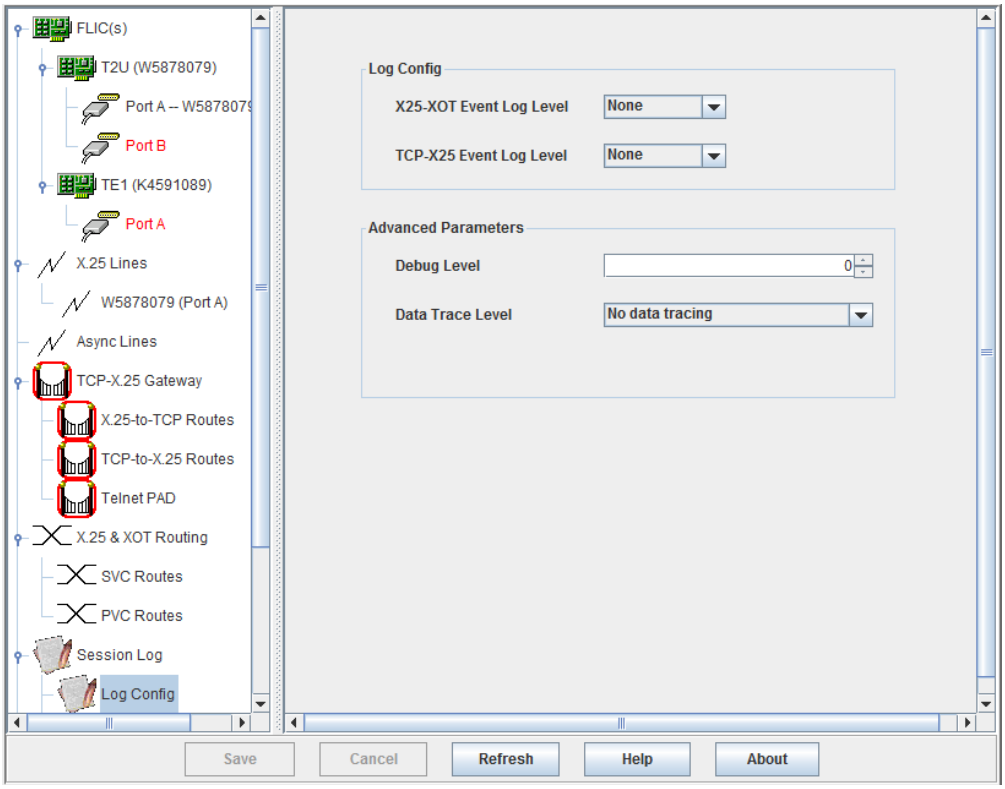
Click System Date and Time under **Administration** and input the current date and time (or preferably setup the SNTP (Simple Network Time Protocol) Configuration to enable the Gateway to use an SNTP server that is accessible from the attached LAN – ensure that your LAN settings are set up correctly e.g. subnet mask, gateway address etc., in order to access the remote SNTP server) – then click Save.

Because log files are named with the date and time, to avoid confusion it is recommended that before changing the date and time you save the configuration, backup any existing log files (if required) and also delete them from the Gateway.

3.6 Log Level

The Gateway can log operational information for activity monitoring and troubleshooting purposes. It is possible to choose the level of information that is logged. The message logging levels should usually be set to the value "Warning" or "Error" to only log messages about abnormal situations. However, when first commissioning a Gateway it is useful to set the level to "Info" in order to view all connection activity. Once the Gateway is operating normally for the required application, the level should be set back to "Warning" or "Error" to reduce the amount of logging which

uses storage, processor and (when remotely logging) network bandwidth.



3.7 IP Security (IPSEC)

Depending on the version of the FarLinX Gateway that you have purchased, your Gateway may or may not include support for IPSEC (IP Security: encryption and authentication of communications over an IP network). If your Gateway does support IPSEC, then an IPSEC link will be displayed in the Configuration section of the navigation bar.

IPSEC provides communications security in IP networks, especially when public Internet connections are used for part of the communications. IPSEC operates between a pair of devices. Each must know details of the remote device in order to authenticate communications and successfully encrypt/decrypt data being sent and received.

Each FarLinX Gateway can be configured with the details of multiple partner Gateways. The pre-shared key used to authenticate must be matched to the correct partner. Once authentication has completed, an exchange of encryption keys takes place under the pre-shared key. The session key is used only for a single session. The pre-shared keys are only used to exchange a limited amount of information but best practice is to change them regularly.

The IPSEC configuration shows a summary list of the remote partner Gateways. Each remote Gateway must be configured with the details of the local Gateway, with matching pre-shared keys.

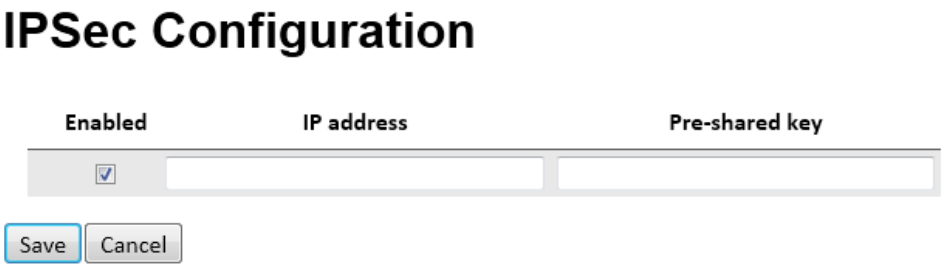
IPSEC is only supported in conjunction with IPv4 addressing.

3.7.1 IPSEC Summary Page

The following shows the IPSEC summary page. Use this control to add and edit IPSEC definitions.



3.7.2 IPSEC Add/Edit entry dialog



The remote system can be referenced in the configuration using its IPv4 address (note that its IPv6 address cannot be used). The ***Enabled*** check box indicates whether this particular IP Security entry is enabled or disabled.

Once a pre-shared key has been entered, the summary table will show the text “<Configured>” instead of the actual value. The actual value is only ever shown whilst entering a value into the pre-shared field.

3.7.3 Example configuration for three Gateways

This configuration assumes that all three Gateways communicate directly with each other.

At Gateway 1 (Local IP address 192.168.1.1)

Remote IP Address	Pre-shared Key
192.168.1.2	aaaaaaa
192.168.1.3	bbbbbbb

At Gateway 2 (Local IP address 192.168.1.2)

Remote IP Address	Pre-shared Key
192.168.1.1	aaaaaaa
192.168.1.3	ccccccc

At Gateway 3 (Local IP address 192.168.1.3)

Remote IP Address	Pre-shared Key
192.168.1.1	bbbbbbb
192.168.1.2	ccccccc

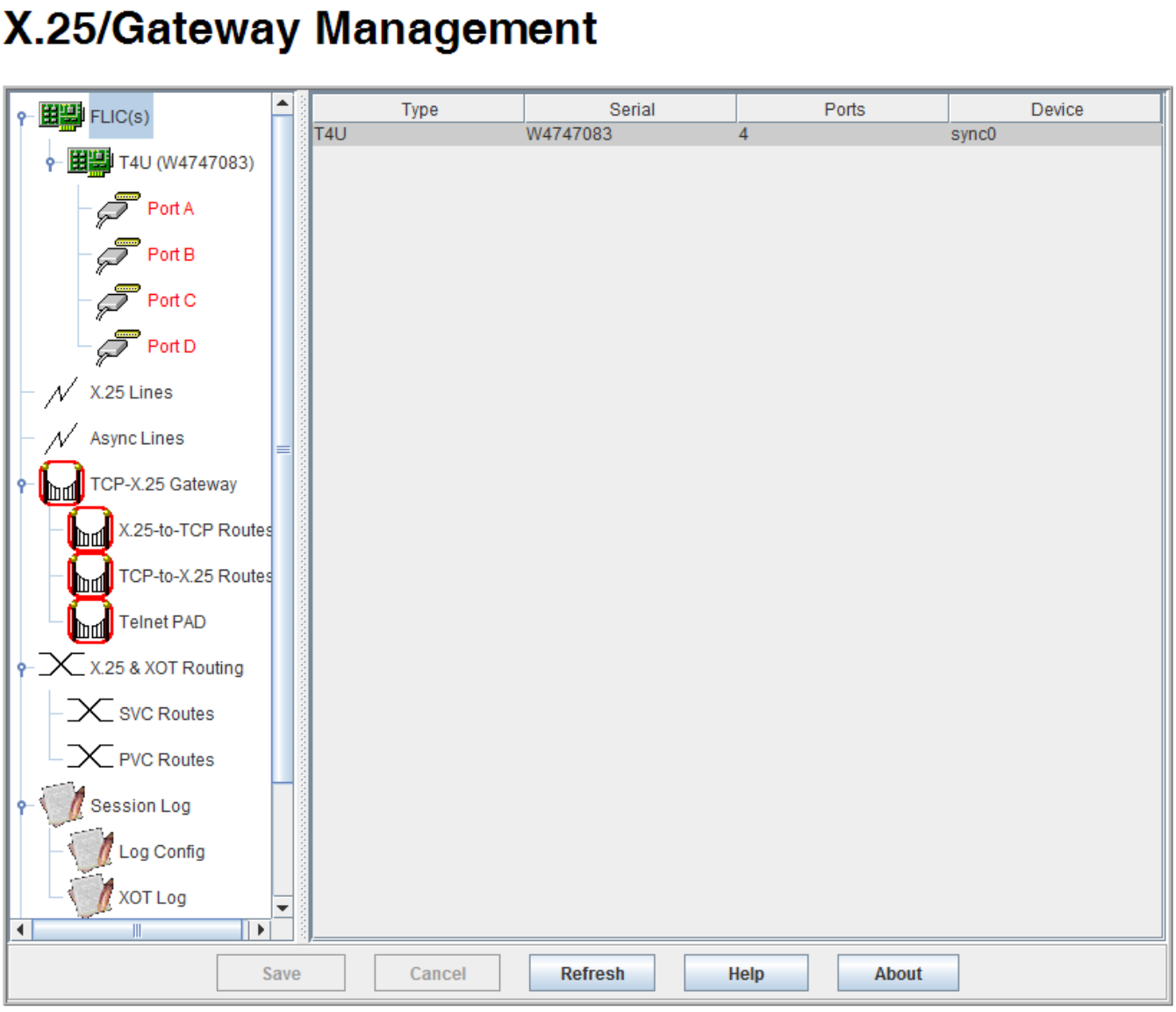
4 X.25 LINE CONFIGURATION

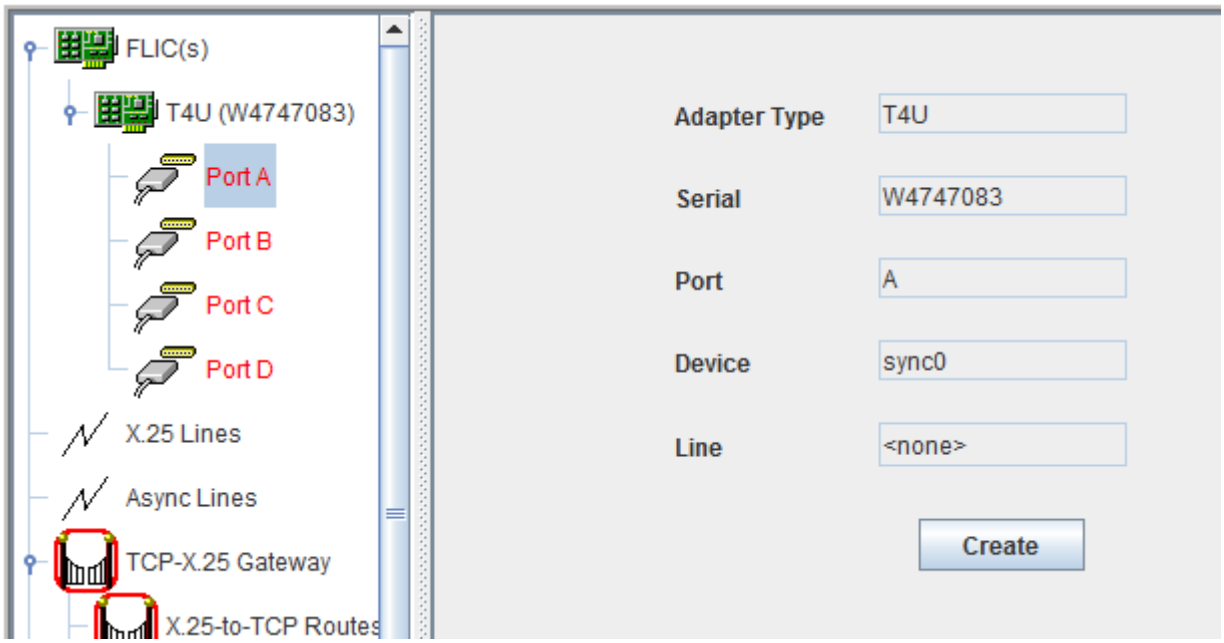
For a Gateway with one or more physical X.25 lines, it is necessary to set up the Gateway's X.25 line configuration to enable the Gateway to connect to the target X.25 networks or hosts. When configuring the Gateway, the term "port" is used to refer to the physical DB25 port on the back of the Gateway. The term "line" is used to refer to a X.25 line configuration definition which is associated with the physical port. The line definition must match the network configuration (as used by the network provider or connected host equipment) of the physical line that is plugged into the corresponding Gateway "port".

If the Gateway is only being used for XOT operation and not physical X.25 connection, then it is not necessary to create and configure an X.25 line.

4.1 Create an X.25 Line

Click X.25/Gateway Management, under the **Configuration** section of the navigation bar, to launch the Java applet to allow the configuration of the X.25 line's network parameters. The first time you run this applet you may need to accept security warnings to trust the publisher. Here is an example display:





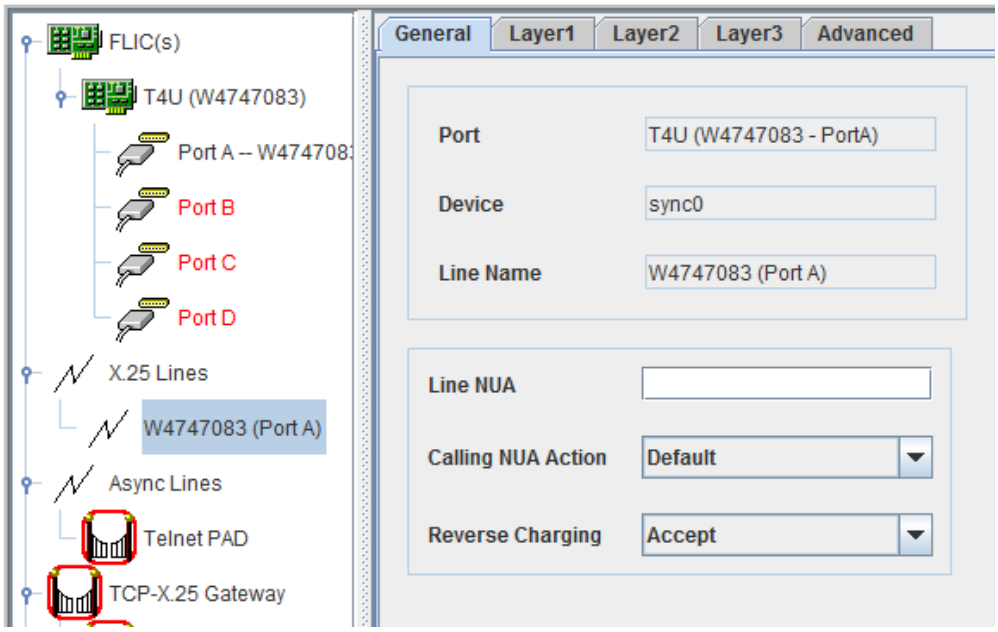
The 'FLIC(s)' (FarLinX Line Interface Cards) section shows all of the X.25 adapters and X.25 ports on the Gateway. Normally there is one adapter with up to four X.25 ports installed on the Gateway, but the Gateway can be upgraded to 2 adapters with up to 8 X.25 ports.

The configuration for each X.25 port is created by right-clicking on a port and choosing ‘Create’ or selecting the port name and clicking the “Create” button in the right hand window. Either action will produce a new item under the 'X.25 Line' sub-branch with all of the parameters of each layer then available to be modified on the page. After all parameters of each layer are configured, click the 'Save' button to save the configuration.

The following sections detail the configurable parameters available in the X.25 line definition.

Once all the parameters have been configured, click the 'Save' button to commit the configuration. If you encounter any issues saving the configuration, for example changed parameters appearing not to have been saved, then please refer to section 3.3 to check the Java settings once again.

4.2 Configure X.25 Line



Once a line has been created the right hand window will show a series of tabs containing the available parameters.

4.2.1 General

Line NUA: The Line NUA is optionally used to define the root NUA for the line (i.e. the NUA without subaddress). It allows an application to define a name as the sub-address only; the Gateway then adds on the Line NUA when routing incoming calls. The Line NUA is used when the Calling NUA action is set to "Insert Line NUA".

Calling NUA Action: There are 4 different options for how the calling NUA action in an outgoing call request is treated:

Default (Calling NUA Unchanged)
Remove Line NUA
Insert Line NUA
Omit Entirely (Calling NUA set to zero length)

Reverse Charging: This option determines whether incoming X.25 calls with the reverse charging facility will be accepted or refused.

4.2.2 Layer1 Settings

4.2.2.1 Layer1 Settings for Synchronous Serial Lines

Baud Rate: This value specifies the clock rate (i.e. line speed) in Bits per Second to be used on this line. This is always required if the internal clock option is used since it explicitly determines the rate at which clocks are generated on the line by the Gateway. Even if internal clock generation is disabled, this value should match the externally generated clocking rate of the line since this value is used to calculate a suitable value for the LAPB T1 parameter of the link.

Line Interface: The Gateway supports following network interface selections:

X.21	Also known as V.11
X.21D	A variant of X.21 with dual clock lines
V.24	Also known as RS232C and X.21bis
V.35	V.35 DTE operation
UX35C	A cable presenting a V.35 DCE interface
RS530	Uses RS-422 level signalling
UXD1	A special cable for Nortel DMS100 switch connection
E1	When the Gateway is fitted with a card supporting an E1 interface

Line Termination: This option, available for certain port types only, is used to select the port's termination characteristics – either None or Resistive.

Encoding: This option defines the type of encoding to be used on the line – either NRZ, NRZI, FM0 or FM1. Some options are available only for certain port types.

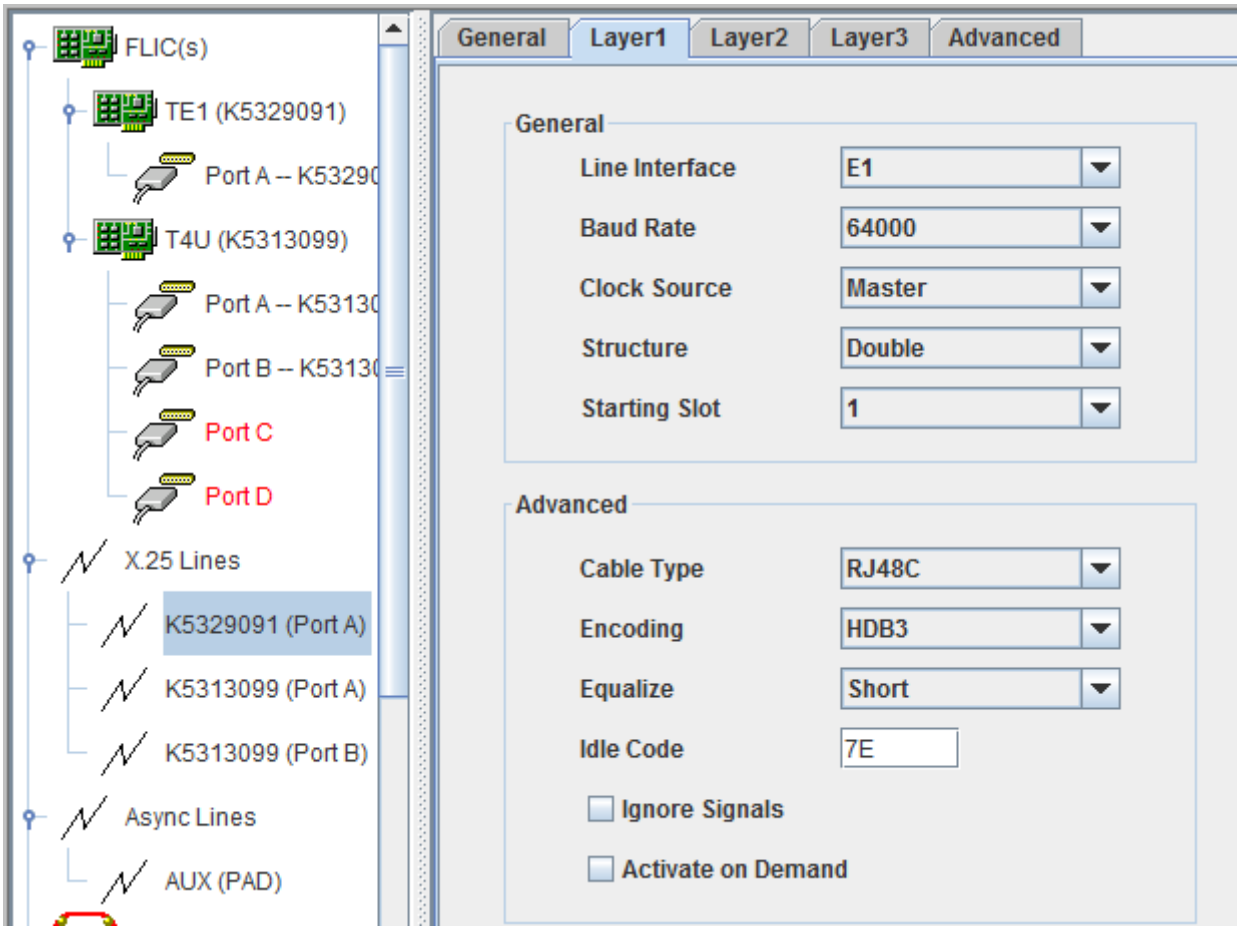
Internal Clock: Select this option to enable internal clock generation for the line (at the rate specified in *Baud Rate*). If clocking is generated by the peer then deselect this option. If the Gateway is acting as a DCE then an internal clock is normally required.

Rx Clock Inversion: This option is used to change the phase of the internal clock by 180 degrees for (received) data. If you are seeing excessive receive errors and the port is configured for internal clocking then inverting the clock may solve the problem.

Ignore Signals: The Gateway will always attempt to activate the line regardless of the state of the signals. However, if this option is deselected, a subsequent transition from the active state of the carrier signal (DCD in the cases of V25 and V.35 or Indicate in the case of X.21) to the inactive state will cause the link to be deactivated, resulting in all virtual circuits being cleared.

4.2.2.2 Layer 1 Settings for E1 interfaces

The following parameters are displayed only when the port is an E1 interface card:



Line Interface: This parameter defines if the E1 interface is to operate in an E1, T1 or J1 mode.

Baud Rate: This parameter defines the data rate of the line. E1 unframed will run at a fixed 2048 Kbps, T1/J1 unframed will run at a fixed 1544 Kbps. E1 framed can operate at any 64 Kbps multiple between 64 Kbps and 1984 Kbps. T1/J1 framed can operate at any 64 Kbps multiple between 64 Kbps and 1536 Kbps.

Clock Source: This parameter is used to select the clock source for the E1 interface. The Master setting configures the E1 interface to use an internal crystal oscillator as the transmit clock source. The Slave setting configures the E1 interface to use the recovered receive clock as the transmit clock source.

Structure: This parameter defines whether the E1 interface operates in a framed or unframed mode. E1 supports Double, CRC4 and CRC4-Modified multiframe modes. T1/J1 supports F4, F12, F24 and F72 multiframe modes.

Starting Slot: When a fractional rate has been defined, this parameter can be used to specify the time slot number in which transmissions are started. The starting timeslot is not used in unframed modes.

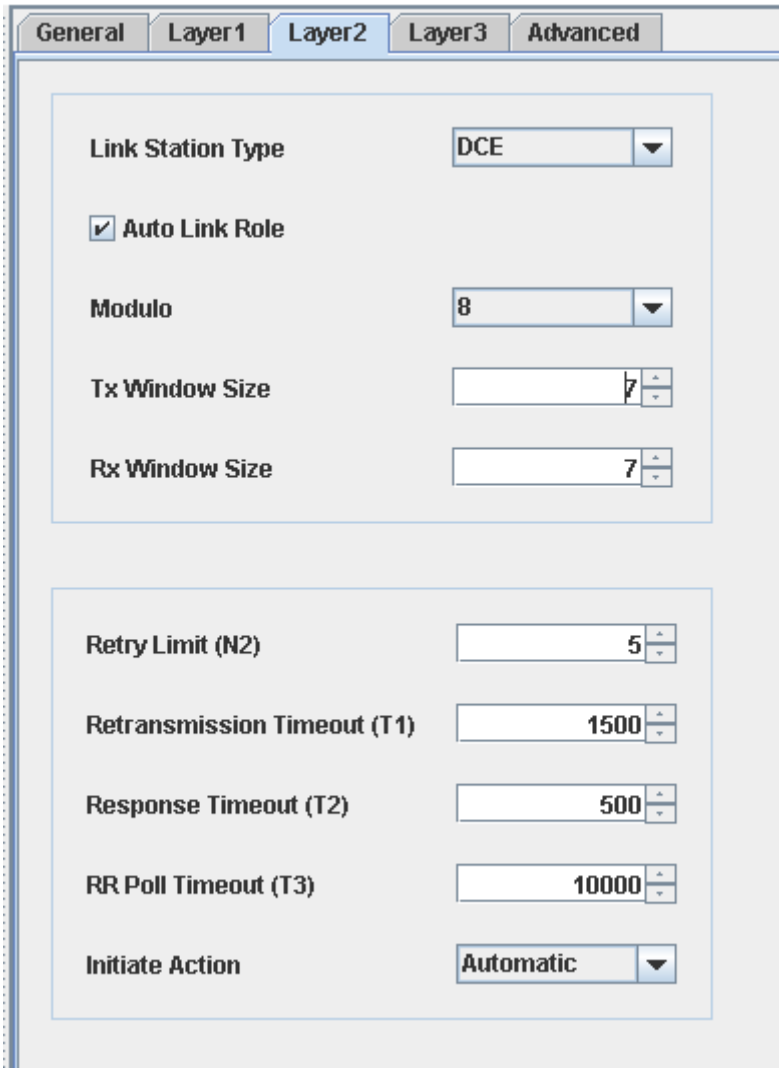
Cable Type: This parameter should be set according to the physical Interface type in use. RJ48C is for shielded or unshielded twisted pairs (E1: 120 ohm, T1/J1: 100ohm). BNC is used with two coaxial cables for E1 only (E1: 75 ohm).

Encoding: This parameter defines the Line Coding scheme to be used. Only those codes available for the particular framing scheme will be available. Note: AMI coding is not recommended, since it requires that user data meets certain ones-density requirements.

Equalize: This parameter is used to define whether the circuit length is Short or Long haul.

Idle Code: The parameter specifies the data pattern transmitted in unused timeslots. The Idle Code can be any hex value between 00 and FF. NOTE: Care should be taken in selection of the Idle Code when using AMI Coding. In most cases the default value of 0x7E should suffice.

4.2.3 Layer2 Settings



Link Station Type: Determines whether the line is to operate as an X.25 DTE or DCE. When connecting to a network, the network is normally a DCE, in which case the line should be configured as a DTE. If the Gateway is acting as the X.25 network, configure the line as a DCE.

Auto Link Role: When Auto Link Role is selected, the Gateway automatically determines whether it should act as a DTE or DCE and so allows the X.25 link to operate even if the Link Station Type has been misconfigured. If Auto Link Role is set FALSE, then the link Station Type must be configured correctly.

Modulo: Determines whether to use normal (modulo 8) or extended (modulo 128) sequence numbering for information transfer. Modulo 8 is much more common than modulo 128.

Tx Window Size: Determines the transmit window size, a window size of 7 is most common. The window size value must be less than the configured sequence number modulus, thus window sizes of greater than 7 may only be selected if modulo 128 sequence numbering is also selected. When Modulo 8 sequence numbering is selected, the window size is almost always 7.

Rx Window Size: Determines the receive window size, a windows size of 7 is most common. The window size value must be less than the sequence number modulus, thus window sizes of greater than 7 may only be selected if modulo 128 sequence numbering is also selected. When Modulo 8 sequence numbering is selected, the window size is almost always 7.

Retry Limit (N2): Defines the number of retries after which the link state will change. After receiving no response to N2 RR polls, the link will be reset and a SABM frame transmitted. Then after N2 SABMs with no response, the link will go to the Down state and link initiation will be attempted. The parameter is not normally relevant when the link is operational.

Retransmission Timeout (T1): The T1 period is the duration to wait for an acknowledgement, before attempting to retransmit a frame. T1 must therefore be long enough to allow a complete window of I frames to be transmitted in both directions. If T1 is too short, it will result in unnecessary retransmissions and inefficient operation of the link. On the other hand, if T1 is too long, error recovery will be slower than necessary. Errors, however, are generally infrequent, so it is much better for T1 to be too long instead of too short. T1 is normally calculated automatically from the line speed, the maximum frame size and the level 2 window size. Its value, however, can be overridden by explicit configuration.

Response Timeout (T2): The period to wait (in the hope of being able to send an implicit acknowledgement within an I frame) before sending an RR response frame after receiving an I frame. If set to zero, then an RR response is always sent immediately.

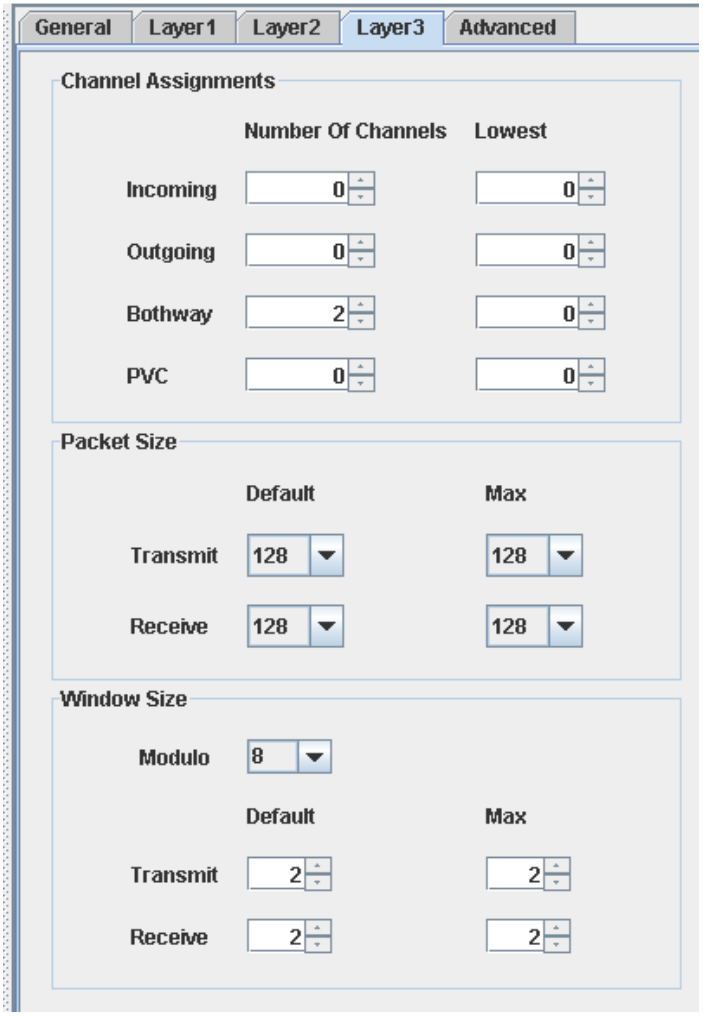
RR Poll Timeout (T3): Determines the period of RR polling when the link is idle.

Initiate Action: Determines the action to take when starting the link. Possible actions are:

- Automatic
- Transmit SABM
- Transmit DISC
- Transmit DM
- Wait (for peer to transmit SABM)

Automatic link initiation is strongly recommended, as it should cause the link to be started whatever the configuration of the peer.

4.2.4 Layer3 Settings



The X.25 line channel configuration **must** match the corresponding configuration of the peer/network. Bothway channels are the easiest to setup since you simply configure the same range at both ends of the link.

Number of Incoming Channels:The number of logical channels that are allocated for Incoming only virtual circuits. When connecting to a network, the channel assignments will be defined by the network provider and the configuration must correspond with those assignments. Incoming only channels are rarely used, the channels are much more likely to be Bothway.

Lowest of Incoming Channels: The number of the lowest channel in the range assigned for incoming only calls, defined as a 12-bit number (i.e. 0–4095). Channel 0 is not normally available for assignment on most networks.

Number of Outgoing Channels: The number of logical channels that are allocated for Outgoing only virtual circuits. When connecting to a network, the channel assignments will be defined by the network provider and the configuration must correspond with those assignments. Outgoing Channels are rarely used – the channels are much more likely to be Bothway.

Lowest of Outgoing Channels: The number of the lowest channel in the range assigned for outgoing only calls, defined as a 12-bit number (i.e. 0–4095). Channel 0 is not normally available for assignment on most networks.

Number of Bothway Channels: The number of logical channels that are allocated for two-way SVC operation; i.e. they may be used for incoming or outgoing calls. When connecting to a network, the channel assignments will be defined by the network provider and the configuration must correspond with those assignments. If the Gateway is acting as the network, the configurator of the Gateway must decide what channel numbers are required.

Lowest of Bothway Channels: The number of the lowest channel in the range assigned for both incoming and outgoing calls, defined as a 12-bit number (i.e. 0-4095). Channel 0 is not normally available for assignment on most networks.

Number of PVCs: The number of available Permanent Virtual Circuits. When connecting to a network, the channel assignments will be defined by the network provider and the configuration must correspond with those assignments. If the Gateway is acting as the network, the configurator of the Gateway must decide what PVC channel numbers are required.

Lowest of PVCs: The number of the lowest channel in the range assigned for permanent virtual circuits, defined as 12-bit number (i.e. 0–4095). Channel 0 is not normally available for assignment on most networks

Default Transmit Packet Size: Determines the transmit packet size if no packet size negotiation takes place when a virtual circuit is established.

Max Transmit Packet Size: Determines the maximum transmit packet size that may be used when negotiating the packet size during virtual circuit establishment.

Default Receive Packet Size: Determines the receive packet size if no packet size negotiation takes place when a virtual circuit is established.

Max Receive Packet Size: Determines the maximum receive packet size that may be used when negotiating the packet size during virtual circuit establishment.

Window Modulo: Determines whether to use normal (modulo 8) or extended (modulo 128) sequence numbering for data transfer. Most networks use Modulo 8, the exception being some Satellite networks and other networks with long network end-to-end response times.

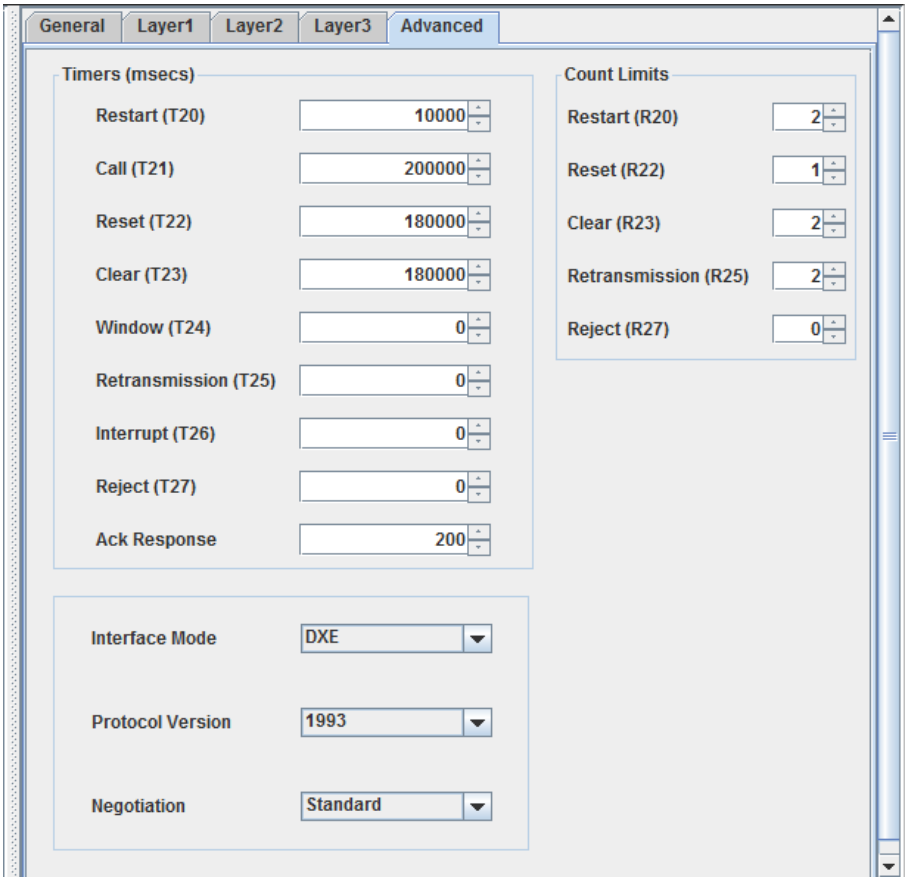
Default Transmit Window Size: Determines the transmit window size if no window size negotiation takes place during virtual circuit establishment. Note that the window size value must be less than the sequence number modulus, thus window sizes of greater than 7 may only be selected if modulo 128 sequence numbering is selected. Window sizes of 4 and above normally result in more efficient (faster data transfer) network operation.

Max Transmit Window Size: Determines the maximum transmit window size that may be used when negotiating the window size during virtual circuit establishment. Note that the window size value must be less than the sequence number modulus, thus window sizes of greater than 7 may only be selected if modulo 128 sequence numbering is selected.

Default Receive Window Size: Determines the receive window size if no window size negotiation takes place when a virtual circuit is established. Note that the window size value must be less than the sequence number modulus, thus window sizes of greater than 7 may only be selected if modulo 128 sequence numbering is selected. Window sizes of 4 and above normally result in more efficient (faster data transfer) network operation.

Max Receive Window Size: Determines the maximum receive window size that may be used when negotiating the window size during virtual circuit establishment. Note that the window size value must be less than the sequence number modulus, thus window sizes of greater than 7 may only be selected if modulo 128 sequence numbering is selected.

4.2.5 Advanced Settings



Note: all timeout periods are specified in milliseconds.

Restart(T20): The timeout period to wait for a response after transmitting a restart request packet. After a T20 expiry, another restart request will be transmitted, up to R20 times.

Restart(R20): The number of attempts at retransmitting a restart request packet before giving up.

Call(T21): The timeout period to wait for a response after transmitting a call request packet. After a T21 expiry, the call will be cleared.

Reset(T22): The timeout period to wait for a response after transmitting a Reset Request packet. After T22 expiry, another reset request will be transmitted, up to R22 times, after which the call will be cleared (assuming it is an SVC).

Reset(R22): The number of attempts at retransmitting a reset request packet before giving up. In the case of an SVC, the call will be cleared.

Clear(T23): The timeout period to wait for a response after transmitting a clear request packet. After a T22 expiry, another clear request will be transmitted, up to R23 times, after which it is assumed that the logical channel is available for a new call.

Clear(R23): The number of attempts at retransmitting a clear request packet before giving up and allowing the logical channel to be used for a new call.

Window(T24): When this is set to a non-zero value, the Gateway will transmit an RR packet every T24 period on active virtual circuits in order to provide window status information to the peer. Normally this isn't required and window rotation should occur perfectly adequately without it.

Retransmission(T25): When this is set to a non-zero value, the Gateway will retransmit any unacknowledged data packets after the expiry of T25. This timer should only be set to a non-zero value in special circumstances, as data packet retransmissions are not normally expected.

Retransmission(R25): The number of attempts at retransmitting a data packet before giving up (assuming T25 is non-zero). After R25, the virtual circuit will be reset.

Interrupt(T26): The timeout period, after transmitting an interrupt packet, to wait for an interrupt confirmation response, after which it is possible to send a new Interrupt packet. If set to zero, then it is not possible to send a new interrupt packet without receiving an interrupt confirmation.

Reject(T27): The timeout period, after transmitting a reject packet, to wait for retransmitted data packets.

Reject(R27): The number of attempts at retransmitting a reject packet before giving up and resetting the virtual circuit.

Ack Response: The time after which an RR packet will be sent in response to a data packet. The Gateway delays sending RR packets in the hope of being able to send an implicit acknowledgement in a data packet. The Gateway is thus optimized for two-way data transfer. If the virtual circuit is being used for unidirectional data transfer (i.e. received data only), then it may be desirable to set this timer to zero.

Interface Mode: Determines whether the layer 3 software should act as a DTE or DCE. If the special value DXE is selected, then the Gateway will determine the mode for itself. (The main difference between a DTE and DCE is in the selection of the channel number when making a call on a bothway channel – DTEs will select the highest available channel; DCEs the lowest. This minimizes call collisions; DTE and DCE also take different actions in the event of a call collision.)

Protocol Version: Defines the version of the CCITT X.25 protocol recommendation to which the software should conform. Possible values are 1980,1984 and 1993. The 1984 recommendation allows interrupt packets of up to 32 octets, instead of being limited to a single octet.

Negotiation: The X.25 standard specifies that negotiation should take place towards a packet size of 128 octets. For example, the caller may request a packet size of 32 and the peer accepting the call will then have the choice of packet size of 32, 64 or 128 octets. Some networks do not follow the X.25 standard and always negotiate towards the lowest packet size (this is referred to as "Downwards" negotiation). So in the example above, the peer would have the choice of a packet size of 32 or 16 octets. The Gateway can be configured for either mode of negotiation. Alternatively, negotiation may be disabled completely.

4.3 X.25 Line Operation

A number of operations can be carried out on the line by right clicking the line's tree node and selecting the required option from the popup menu:

4.3.1 Delete

Right click the X.25 line and click 'Delete' to remove the line.

4.3.2 Rename

Right click the X.25 line and click 'Rename', and then input the new name for the X.25 line.

4.3.3 Restart

Stop (deactivate) and then start (activate) the line again

4.3.4 Start

Start (activate) the X.25 line

4.3.5 Stop

Stop (deactivate) the X.25 line

4.3.6 Attach to Port …

The Gateway allows the user to move the X.25 line from one physical port to another physical port. Right-click the X.25 line and click ‘Attach to Port…’. Right-click the new port that the line going to move to and click ‘Attach Line XXX…’

4.3.7 Clone …

The Gateway allows the user to copy the X.25 line parameters from an existing line to another physical port. Right-click the X.25 line and click ‘Clone…’. Right-click the required physical port that the line configuration is going to be copied to and click ‘Clone Line XXX’

4.3.8 Monitor Start/Stop

This option allows the user to monitor the traffic being sent/received on the X.25 line. Refer to Section 12 for more details.

5 PAD

5.1 Overview

The Gateway supports a triple-X PAD (X.28, X.29, X.3). This allows applications already written to use a PAD to use the FarLinX PAD without making any changes.

The PAD data is routed to X.25 or XOT according to the SVC Routing Table (see Section 6.1).

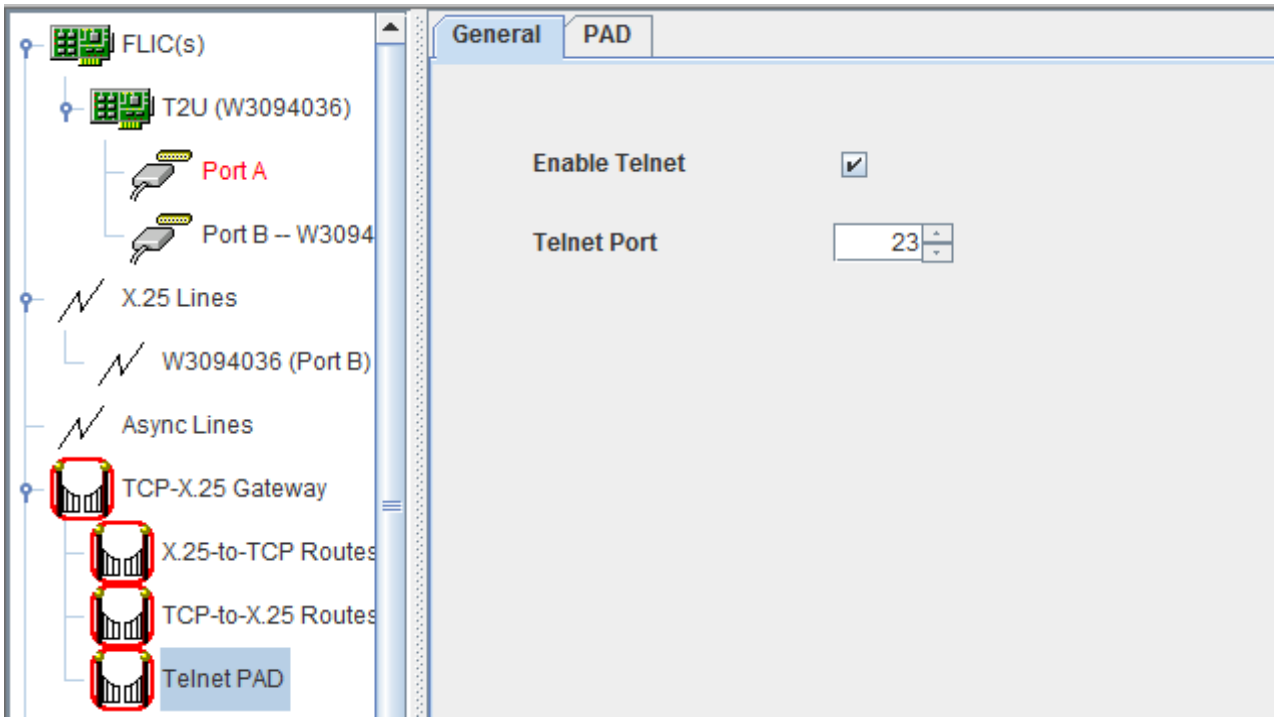
The PAD interface is accessible via:

- the built in asynchronous port (AUX) on the FarLinX
- Telnet access over TCP/IP

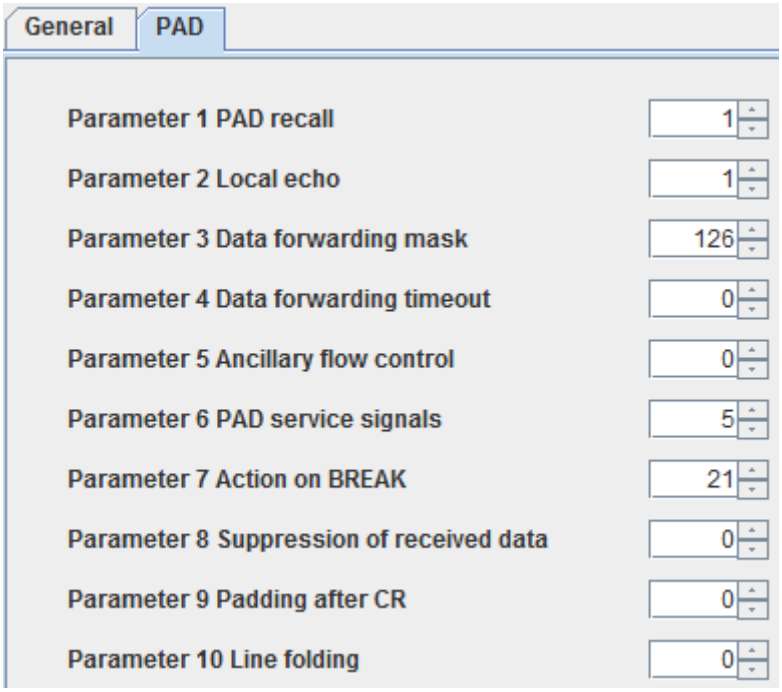
Each PAD port uses its own set of PAD parameters that are independent of the other PAD port.

5.2 Configure Telnet PAD

Click the X.25/Gateway Management, under the **Configuration** section of the navigation bar, to launch the Java applet and click the **Telnet PAD** tree node under **TCP-X25 Gateway**. The Telnet option will then appear in the right-hand window. The Telnet PAD service is disabled by default. Click the **Enable Telnet** check box to enable it. The Telnet port number can be changed if required.



Click on the **PAD** tab to configure default values for any of the set of 22 X.3 parameters for this PAD. Every parameter has a default value which is set when the Telnet sessions are created. The values in a session could then be changed dynamically via standard X.28/X.29 commands. It is important to note that these changed values only then take effect during the lifetime of the session. After the session is closed, the configured default values will be used when the PAD session is started again.

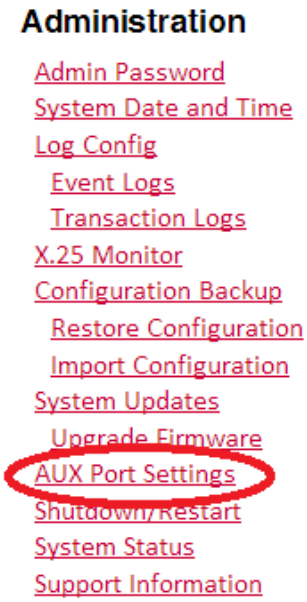


Note that if the Telnet PAD support is disabled (by deselecting the **Enable Telnet** check box) when there are **active** Telnet PAD sessions, these sessions will be allowed to continue until disconnected by the session partners i.e. they will **not** be disconnected/aborted by the Gateway just as a result of disabling the feature. However, no new Telnet PAD sessions will be able to be started until the feature is subsequently enabled again.

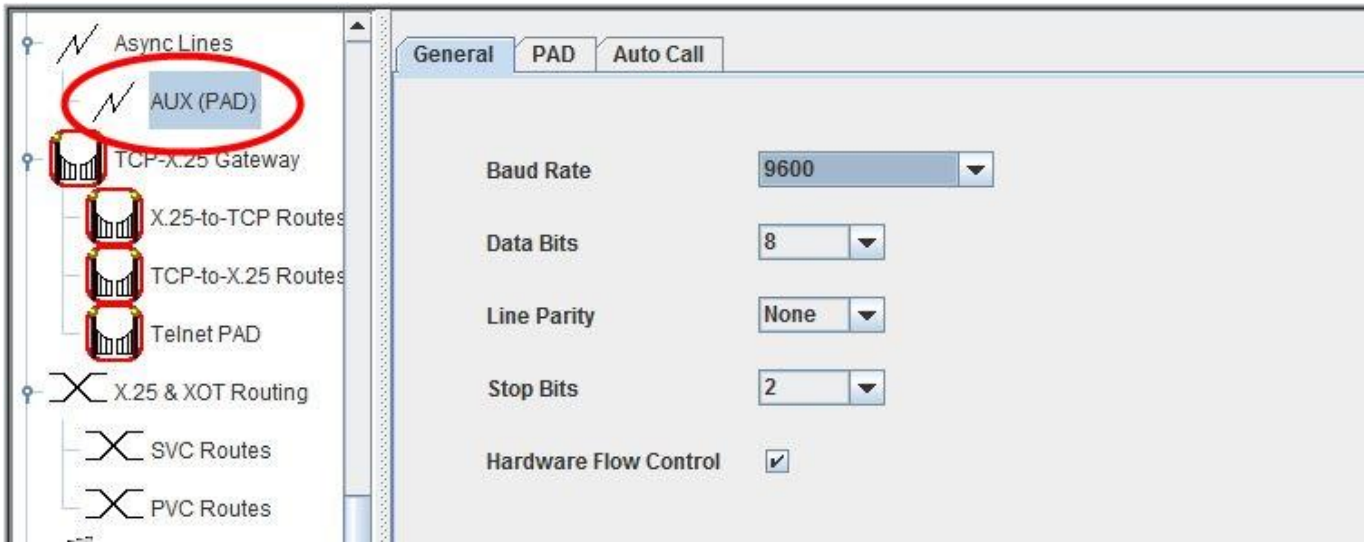
5.3 Configure AUX Port

The AUX port can be used for the Live/Standby Gateway function (see section 15) or can be used as a PAD port.

To configure it for use by the PAD click **Configure AUX Port** under Administration, select **PAD** and then save.



Once the AUX port has been assigned as a PAD port, when the X.25/Gateway Management applet is started the AUX port appears under the **Async Lines** tree node. Clicking on it displays the async serial parameters that are currently in use. These are the standard async serial parameters which must be configured to the same values as the connected device.



Click on the **PAD** tab to configure default values for any of the set of 22 X.3 parameters for this PAD. They will be used when the PAD session is created on connecting the user's terminal device. The values could be changed during the session via standard X.28/X.29 commands but these modified values will only be used during the lifetime session. After the session is closed, the configured default values will be used when the PAD session is started again.

5.4 Auto Call

The PAD Auto Call function can be enabled so that when an incoming PAD connection (from an async terminal) has been detected, a preconfigured NUA is called automatically and the PAD terminal is connected to the remote PAD application without the need for any manual intervention.

The Auto Call tab contains the following fields:

5.4.1 Autocall Enable

If Autocall enabled, when PAD terminal is opened/connected, pad call will be started automatically.

5.4.2 Destination NUA

The Destination NUA field is mandatory and its length can be up to 15 valid NUA bytes. Note that related SVC route will be added/modified automatically according to this field.

5.4.3 Destination Line

Destination Line can be either a local X.25 line or XOT line. If XOT line has been selected, the Destination IP or hostname must be specified.

5.4.4 Destination IP Address or Destination Host

Specify Destination IP or hostname for xot connection in this field.

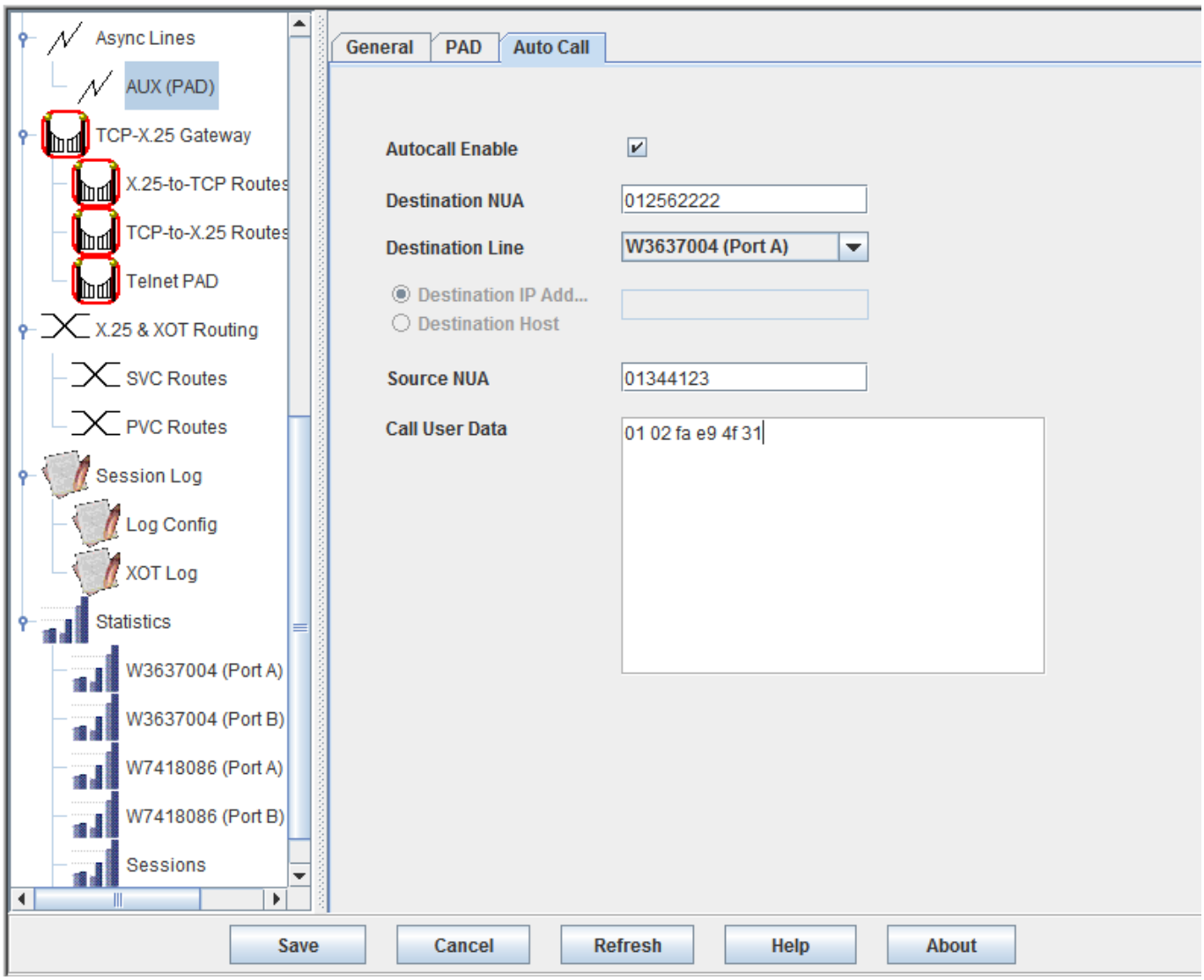
5.4.5 Source NUA

The Source NUA field is optional and its length can be up to 15 valid NUA bytes, the parameter is used as the calling NUA in X.25 Call Requests.

5.4.6 Call User Data

Call User Data is optional which can be inserted into X.25 Call Requests,its length can be up to 128 valid bytes. The

input field should be in hex byte format, e.g. 01 02 fa e9 4f 31 etc, with a space character between each hex byte.



5.5 Default Values for PAD Parameters

As discussed above, the Gateway supports a set of default PAD parameters values for both telnet and serial terminal sessions. The default values are used for each new PAD session. To configure the defaults, open the PAD pane for either the AUX or Telnet PAD, modify the values as required, scroll down to the end and click the **Save As Default Values button**. The **Reset To Default Values** button can be used to reset the values back to their factory preset defaults.



5.6 X.28 Command Reference

This section contains a brief summary of the X.28 commands supported by the Gateway. Refer to the X.28

recommendation for more details of X.28 commands. These commands must be entered while the PAD is in command mode during which you are presented with a PAD prompt (*). The PAD session always starts in command mode. Once a call is active the PAD switches to data mode. You can then manually switch between command and data modes by entering CTRL-P or self-defined character according to RFC1053 at the terminal.

5.6.1 CLR – Clear a Call in Progress

The CLR command is used to clear the current X.28 call.
This command has no effect if issued from command mode when there is no call currently in progress.

Syntax:
CLR

5.6.2 INT – Generate a Call Interrupt Packet

The INT command is used to send an interrupt packet on the currently established virtual circuit.
This command has no effect if a virtual circuit is not established.

Syntax:
INT

5.6.3 PROF

The PROF command display the session's default X3 parameter profiles. This default profile is configured as described in Section 5.4.

Syntax:
PROF

5.6.4 PAR? – Display Numeric Terminal Parameters

The PAR? command is used to display the current settings of the session's X.3 parameters. Only the current session's X.3 values can be displayed.

Syntax:
PAR?[parameter_number[,parameter_number...]]

Examples:

To display all of the parameter settings for the current terminal:
* PAR?
PAR 1:1, 2:1, 3:126, 4:0, 5:2, 6:5, 7:21, 8:0, 9:0, 10:0, 11:14, 12:1,13:4, 14:0, 15:0, 16:127, 17:21, 18:18, 19:2, 20:192, 21:0, 22:0

To display settings for parameters 1, 2, 21 and 11 for this terminal:
* PAR?1,2,21,11
PAR 1:1, 2:1, 21:0, 11:14

5.6.5 RESET – Generate a Reset Packet

The RESET command is used to reset the user's current virtual circuit. This command has no effect if a virtual circuit is not established.

Syntax:
RESET

5.6.6 SET – Set Numeric Terminal Parameters

The SET command is used to modify the active session's current X.3 parameter values. Note that multiple parameter values can be set simultaneously.

Syntax:
SET parameter:value[,parameter:value...]

Example:
*set 1:0
*
SET? – Set and Confirm Numeric Terminal Parameters
The SET? command is used to modify and display the active session's current X.3 parameter values.

This command performs the same functions as the SET command but also then displays the new parameter setting(s).

Syntax:
SET? parameter:value[,parameter:value...]

Example:
*set? 1:1
PAR 1:1

*

5.6.7 STAT – Examine Call Status

The STAT command is used to display the current call status. Possible returned status values are ENGAGED and FREE.

Syntax:
STAT

5.6.8 Initiate a Call

To initiate a call, just type the NUA at the Command Mode prompt. The SVC routing table will then be used to determine how the call will be made e.g. whether to use X.25 or XOT.

Example:
To call address 321458:
* 321458

5.6.9 Terminating a telnet-based PAD session

When using the PAD via a telnet session, the session can be terminated by entering the QUIT command at the PAD prompt.

Example:
* quit

5.7 X.3 PAD Parameters

The FarLinX Gateway supports the following set of X.3 PAD parameters.

Refer to RFC1053 to get detailed information of each parameter.

Parameter No	Function	Value range	Default value

1	PAD state convert	0 ,1,32-126	1
2	Echo	0 – 1	1
3	Data forwarding signals	0 – 255	126
4	Idle timer delay	0 – 255	0
6	PAD service signals	0 , 1, 5	5
8	Discard output	0 – 1	0
10	Line folding	0 – 255	0
11	Binary speed of Terminal	0 – 27	Read only
12	Flow control of the PAD Terminal	0 – 1	1
13	Linefeed insertion after carriage return	0 – 7,	0
15	Editing	0 – 1	0
16	Character delete	0 – 255	127
17	Line delete	0 – 255	24
18	Line display	0 – 255	18
20	Echo mask	0 – 255	0
21	Parity Treatment	0,1,2	0
22	Page Wait	0 – 255	0

The parameters 5, 7, 9, 14 and 19 are not supported by the Gateway. Their configured values are just simply ignored by the Gateway.

6 XOT & X.25 SWITCHING CONFIGURATION



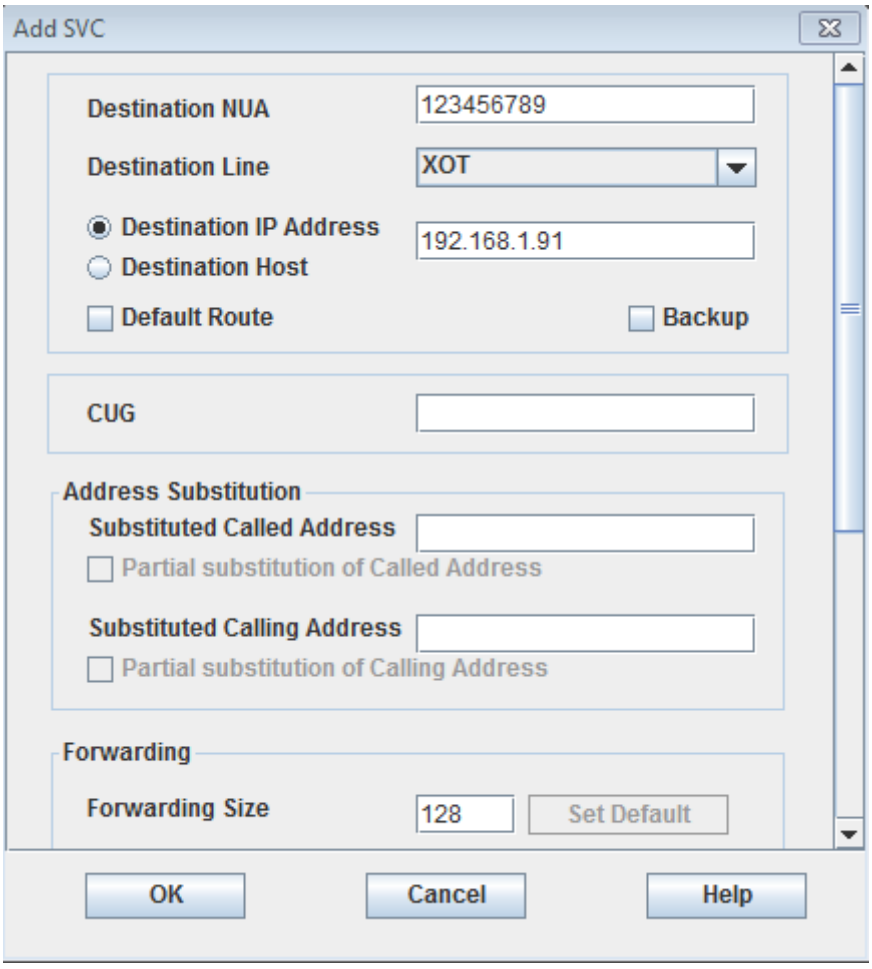
Before the Gateway can route any call from X.25 or XOT, its routing configuration must be set up. The routing entries are contained in 2 routing tables: one for SVC routes and one for PVC routes. The tables can be configured via the links located under **X.25 & XOT Routing**.

6.1 SVC Routing Table

On clicking **SVC Routes**, the current configuration of the SVC routing rules will appear. To add a new SVC routing rule, click the button **Add Entry** that is shown under the routing table summary display. To edit an existing routing rule, select the rule and click **Edit Entry**. To delete an existing routing rule, select the rule entry and then click **Delete Entry**.

6.1.1 Precise Routes

To add a new, specific/'precise' route, click **Add Entry** – the following dialog will appear:

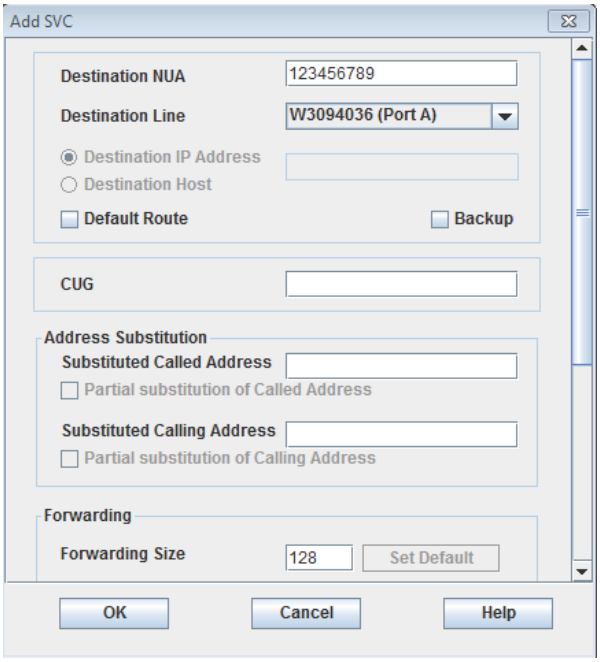


This rule will configure the Gateway to route an incoming call based on its destination NUA (Called DTE Address).

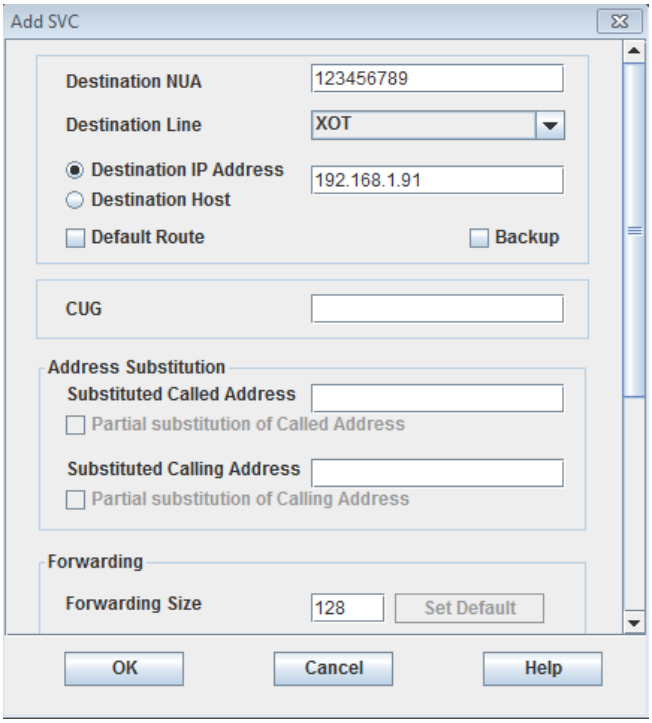
Destination NUA: The destination/called NUA of the incoming call that the Gateway will route to the specified Destination Line. In other words, the Destination NUA will determine which incoming calls will be matched with this route. Note that if this value is identical to the local X.25 NUA configured in an existing X.25-to-TCP route (see Section 7.1), the X.25-to-TCP route will take a higher priority.

Destination Line: The Gateway can route the incoming call to either the X.25 line or to XOT. The Destination Line field identifies which should be used. When XOT is selected as the destination line, the IP address or host name of the remote server must also be specified.

Here are two configuration examples for X.25 and XOT.

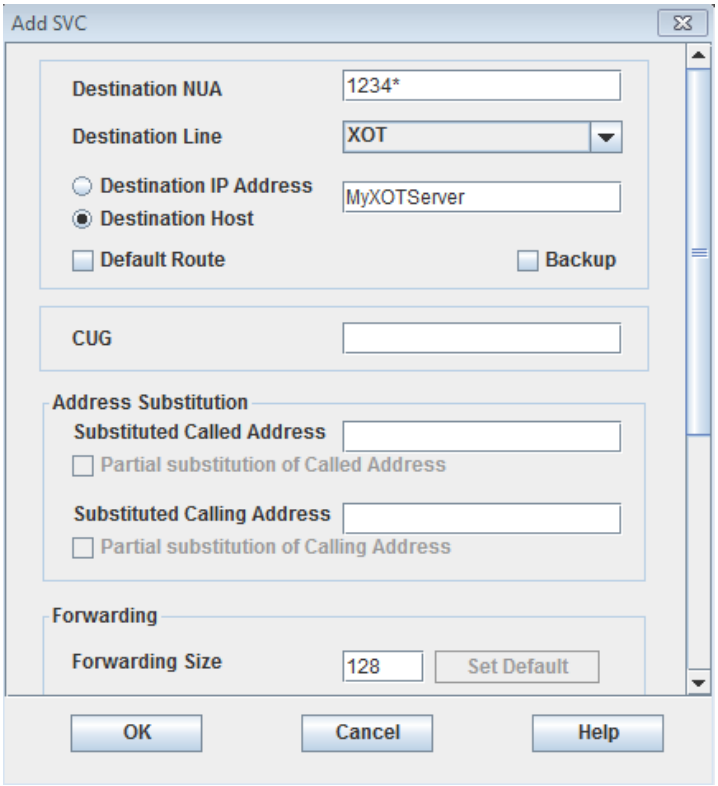


This configuration states that when the Gateway receives an incoming call request with a called address of 123456789, the Gateway will forward the call to the Gateway's X.25 line.



This configuration states that when the Gateway receives an incoming call request with a called address of 123456789, the Gateway will forward it over the TCP/IP network (using XOT) to the remote XOT server with the IP address of 192.168.1.91.

6.1.2 Wild-card Routes:



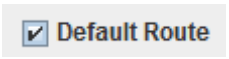
The Gateway also supports wild-card routes for use when multiple similar addresses are to be routed to the same destination. It is not necessary to input the entire NUA in **Destination NUA**, instead the wildcard characters '*' or '?' can be used in this field. The following demonstrates an example use of wild-card routes.

Here the Gateway will route calls in which the called NUA starts with '12348' (e.g. 1234800, 1234801, 1234812345678) to 'MyXOTServer' over the TCP/IP network. When a Host Name is used, it is assumed that the Gateway has been configured with the address of DNS servers (see the LAN Configuration menu) that can resolve the name to an IP address.

6.1.3 Default Route

The Gateway supports an optional default route to be used when no other route matches the called NUA of an incoming call. In this case the Gateway can be configured to route these calls to a specific line (X.25 or XOT).

To define a default route, simply check the **Default Route** box:



Note that the **Destination NUA** field value of the default route will always be changed to '*' regardless of the original input.
If a default route has not been defined, all incoming calls, that do not match one of the routes defined, are cleared by the Gateway.

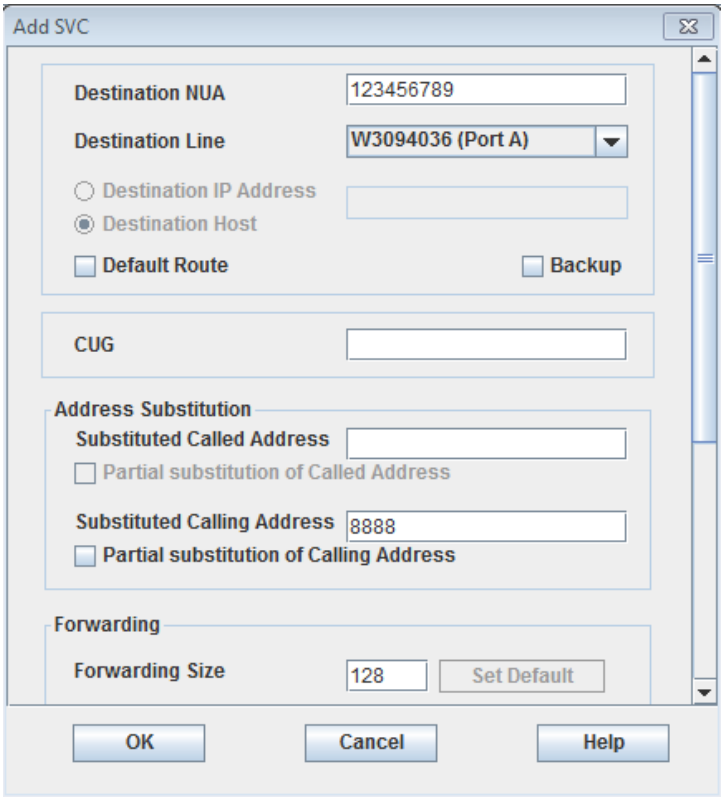
6.1.4 Calling Address Translation

The Gateway can be configured to replace all or part of the calling address field in the X.25 call packet with a specified value. This is done using the **Substituted Calling Address** and **Partial substitution of Calling Address** fields.

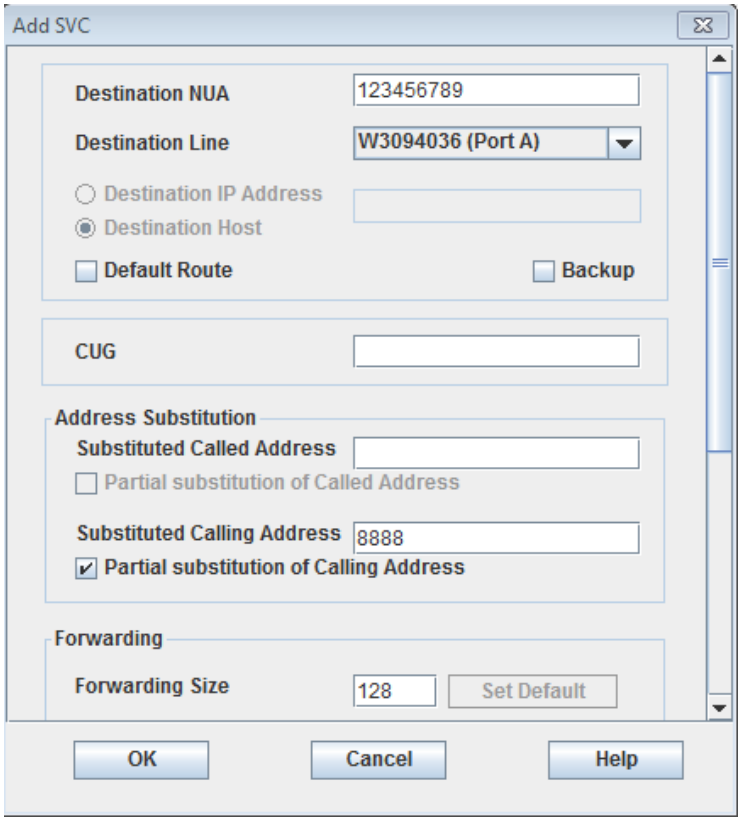
If **Partial substitution of Calling Address** is checked, the Gateway will replace only the number of digits in the **Substituted Calling Address** value, from the start of the calling NUA. The remaining digits in the calling address

field will be left unaltered. Otherwise the Gateway will replace the entire calling NUA with the contents of the **Substituted Calling Address** value.

Consider the following two examples:



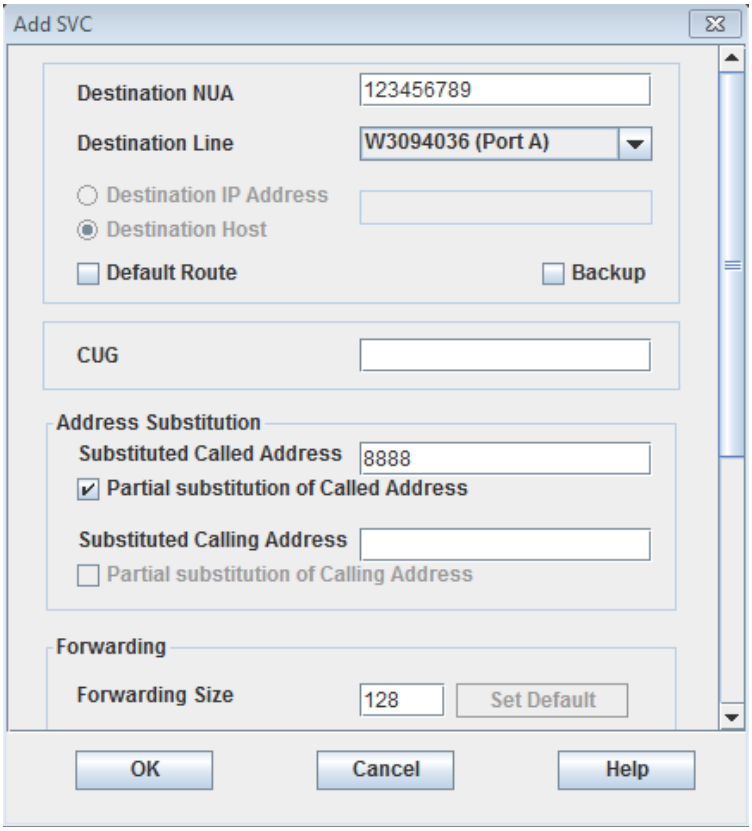
In this first example, because **Partial substitution of Calling Address** is **not** checked, the Gateway will replace the total calling NUA with '8888'. In other words, the calling NUA will be changed to '8888' regardless of its original value.



In the second case, **Partial substitution of Calling Address** is checked, so the Gateway will replace the first 4 digits only. Therefore if the original calling NUA is ‘123456789', the calling address will become '**8888**56789’.

6.1.5 Called Address Translation

The Gateway can be configured to replace all or part of the called address field in the X.25 call packet with a specified value. This is done using the **Substituted Called Address** and **Partial substitution of Called Address** fields.



If **Partial substitution of Called Address** is checked, the Gateway will replace only the number of digits in the **Substituted Called Address** value, from the start of the called NUA. The remaining digits in the called address field will be left unaltered. Otherwise the Gateway will replace the entire called NUA with the contents of the **Substituted Called Address** value.

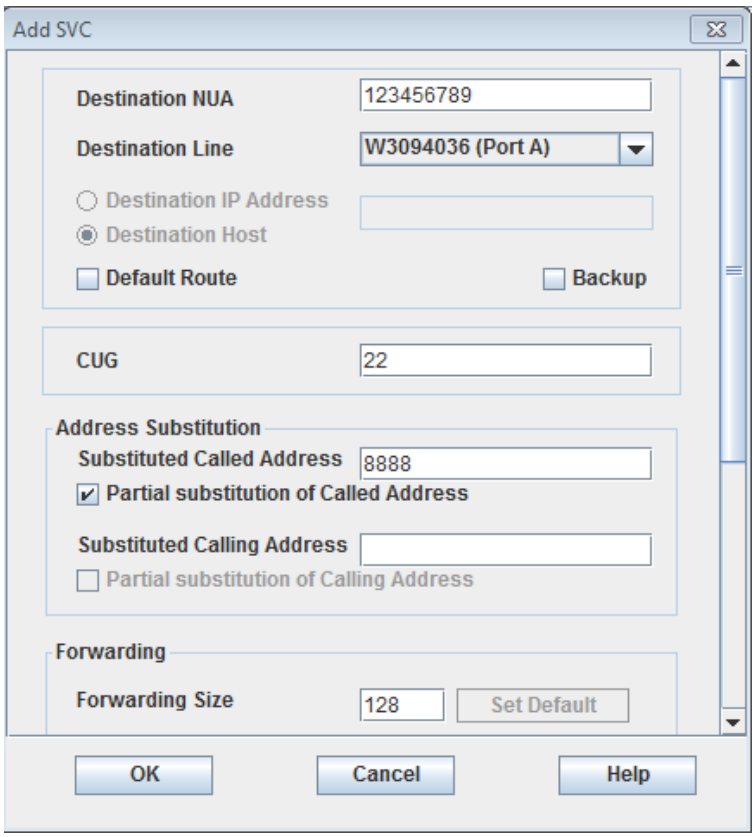
In this example, because the **Partial substitution of Called Address** is **not** checked, the Gateway would replace the total called NUA '123456789' with '8888'.

If instead, **Partial substitution of Called Address** were checked, the Gateway would replace just the first 4 digits of the called NUA and leave the remainder unchanged. Therefore the called address would become '**8888**56789'

6.1.6 Closed User Group (CUG)

The Closed User Group (or **CUG**) is one of the facilities supported in the X.25 protocol.

An X.25 call packet may optionally contain a Closed User Group (**CUG**). If the Gateway has a route that matches the called/destination NUA value but that route also specifies a CUG value, then the Gateway will only forward incoming call requests to that NUA when the call also contains the configured **CUG** value.



In this example, assuming this is the only route defined for **Destination NUA** 123456789, all call requests to that NUA will be refused unless they also contain CUG '22' in the facility field of the X.25 call packet.

6.1.7 Forwarding Size

The **Forwarding Size**, displayed and configured just below the Address Substitution fields, determines the quantity of data that is processed by the Gateway before being forwarded (Note that complete X.25 messages i.e. without the M-bit set, are always forwarded immediately from the X.25 interface, regardless of the configured **Forwarding Size**).

In situations in which transit delays and response times are important, the **Forwarding Size** should be relatively small (but see note below about X.25 packet sizes). On the other hand, a larger value for the **Forwarding Size** can potentially help achieve maximum throughput for bulk data transfer.

Note that there is no benefit in reducing the **Forwarding Size** below the X.25 data packet size used for the connection – the optimum **Forwarding Size** for minimum transit delay should be the same size as the X.25 packet size. It is, however, not always possible to predict the packet size when X.25 packet size negotiation is employed, as it will depend upon the remote X.25 equipment. Some tuning of this parameter in line with actual experience may therefore be required.

6.1.8 TCP Keep Alives

Since XOT links use TCP, the Gateway enables TCP Keep Alives to be optionally configured for any of its configured XOT-based routes. With TCP Keep Alives disabled, in the absence of receiving TCP FIN or RST packets (for example if a cable is unplugged or an intermediate node fails), the FarLinX Gateway will detect the remote disconnection of a TCP session only when it is attempting to send data. Even then, it can take 15 minutes or so to detect the disconnection.

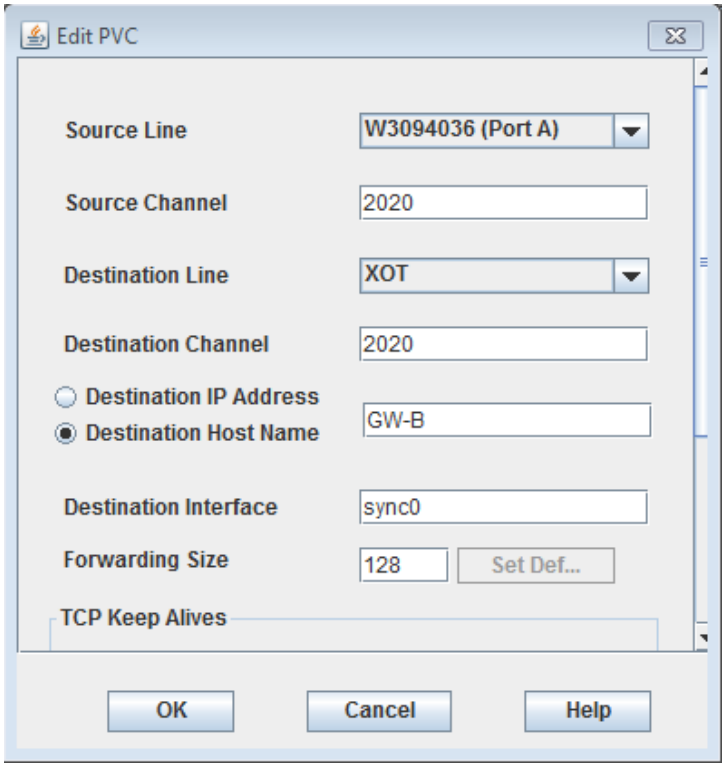
With TCP Keep Alives enabled, the Gateway will be able to detect disconnections within about 3 minutes, even when a session is idle.

It is recommended that TCP Keep Alives are normally enabled (which is the default) and only disabled if there are any known related incompatibilities with the peer.

6.2 PVC Routing Table

On clicking **PVC Routes**, the current configuration of the PVC routing table will appear. To add a new PVC routing rule/entry, click the button **Add Entry** that is shown under the routing table summary display. To edit an existing routing rule, select the rule and click **Edit Entry**. To delete an existing routing rule, select the rule entry and then click **Delete Entry**.

On clicking **Add** (or **Edit**) **Entry** – the following dialog will appear:



This enables you to configure a rule to determine how the Gateway will route a specific PVC. In the example above we are routing the PVC on channel 2020 of the X.25 line via XOT to a destination serial interface called “sync0” on a host named “GW-B”.

The following parameters can be configured:

Source Line: The X.25 line of the source PVC

Source Channel: The logical channel number of the source PVC.

Destination Line: The X.25/XOT line selection of the PVC to be used when being routed by the Gateway.

Destination Channel: The logical channel number of the PVC to be used on the destination line.

Destination IP Address or **Destination Host Name**: The IP address or hostname for the XOT peer system.

Destination Interface: The interface value as configured at the destination XOT peer (only used for XOT-based routes). The FarLinX X.25 Gateway always uses a local interface value of **sync0**. Therefore, if the remote XOT host is another FarLinX X.25 Gateway, then this field should be also be set to **sync0**. Alternatively, if the remote XOT host is a Cisco Router, then this field would be in the style “Serial0/1” (with its actual value being determined by the corresponding configuration on the Cisco Router).

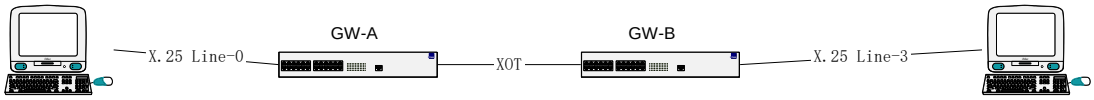
Forwarding Size: This size determines the quantity of data that is processed by the Gateway before being forwarded. (Note that complete X.25 messages i.e. without the M-bit set, are always forwarded immediately from the X.25 interface, regardless of the configured **Forwarding Size**).

In situations in which transit delays and response times are important, the **Forwarding Size** should be relatively small (but see note below about X.25 packet sizes). On the other hand, using a larger **Forwarding Size** can potentially help achieve the maximum throughput for bulk data transfer.

Note that there is no benefit in reducing the **Forwarding Size** below the X.25 data packet size used for the connection – the optimum **Forwarding Size** for minimum transit delay should be the same size as the X.25 packet size. It is, however, not always possible to predict the packet size when X.25 packet size negotiation is employed, as it will depend upon the remote X.25 equipment. Some tuning of this parameter in line with actual experience may therefore be required.

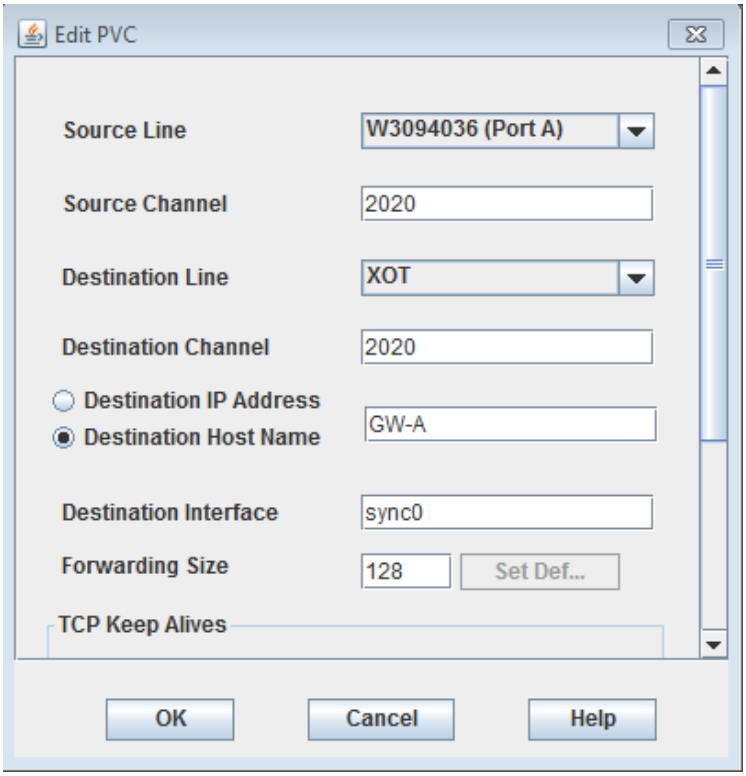
TCP Keep Alives: as with SVCs (see Section 6.1.8), TCP Keep Alives can be optionally enabled/disabled for XOT-based PVC routes. Note that it is strongly recommended to use TCP Keep Alives in conjunction with PVCs so that they can be re-connected after un-notified disconnections.

The following is an example of a PVC connection:



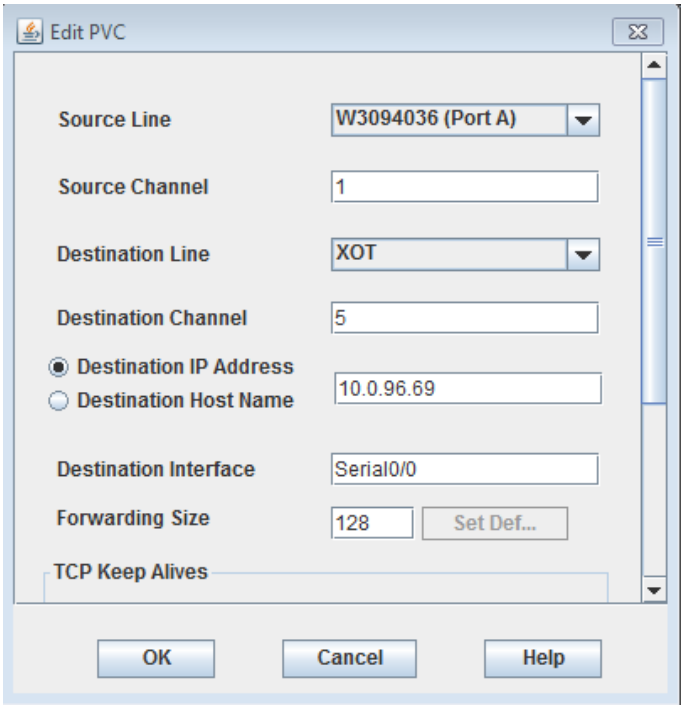
Here we are routing PVCs on the X.25 lines attached to two FarLinX Gateways, GW-A and GW-B, which are themselves connected over XOT.

A typical PVC rule at GW-B, for example to route channel 0, would be:



Note that the **Destination Interface** is set to **sync0**. The rule at GW-A would be the same except **Destination Host Name** would be set to **GW-B**.

If instead you were connecting to a Cisco router over XOT then the **Destination Interface** defined on the FarLinX Gateway should be set to the corresponding **Interface Name** as defined on the Cisco router e.g.



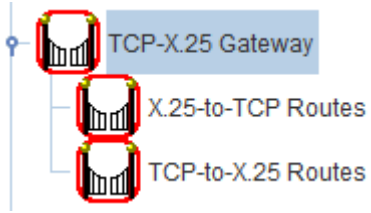
The corresponding configuration on the Cisco would then reference the FarLinX Gateway's details e.g.

interface **Serial0/0**
…
x25 pvc 5 xot 10.0.96.136 interface Serial **sync0** pvc 1

6.3 XOT-to-XOT Routing

The Gateway will not route calls to XOT if they have been received over XOT. In this scenario the call will instead be rejected by the Gateway with Cause=0x0b Diag=0x94 and the session log will report **Route to XOT invalid from: XOT<…> NUA=<…>**.

7 TCP-X.25 CONFIGURATION



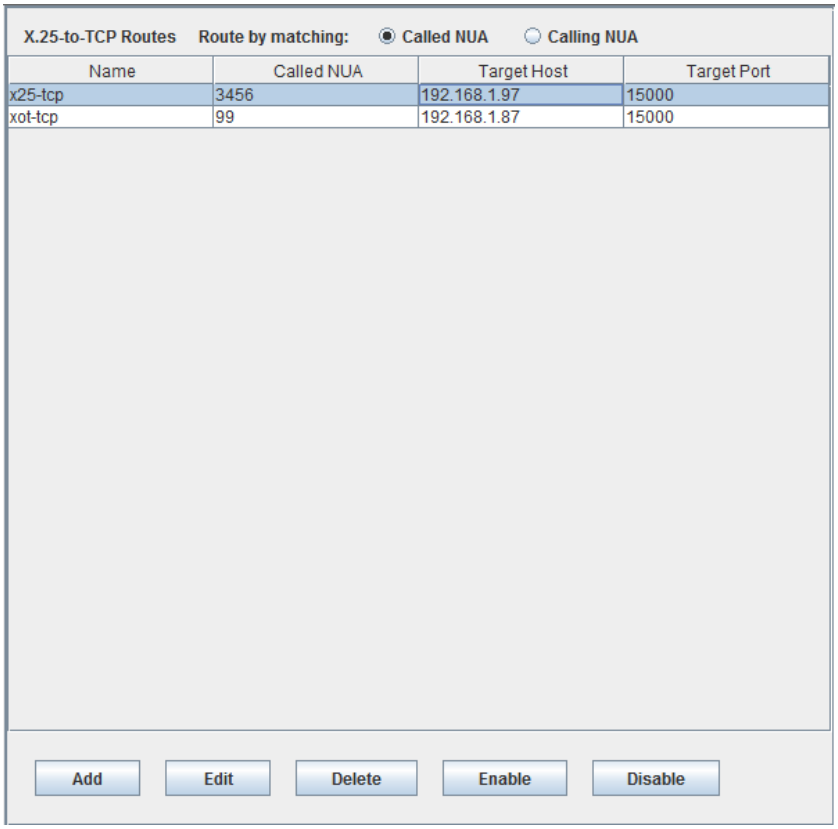
The Gateway can be configured to support both X.25-to-TCP and TCP-to-X.25 connections. Rules for making connections and how to process the resulting data for TCP-X.25 sessions listed under the TCP-X25 Gateway tree node.

When using a PVC channel with X.25 to TCP conversion, the same PVC channel on the same X.25 line cannot be specified in a TCP to X.25 route definition and an X.25 to TCP route definition. Only one can be defined and the GUI will prevent both routes being declared. The correct route to define will depend on the application requirements for initiation of the connection.

7.1 X.25-to-TCP

Route by matching: This setting applies to all X.25 to TCP routes. You can set the gateway to route by matching the called NUA (the destination NUA from the perspective of the remote device making the call), or by matching the calling NUA (the source NUA of the remote device making the call).

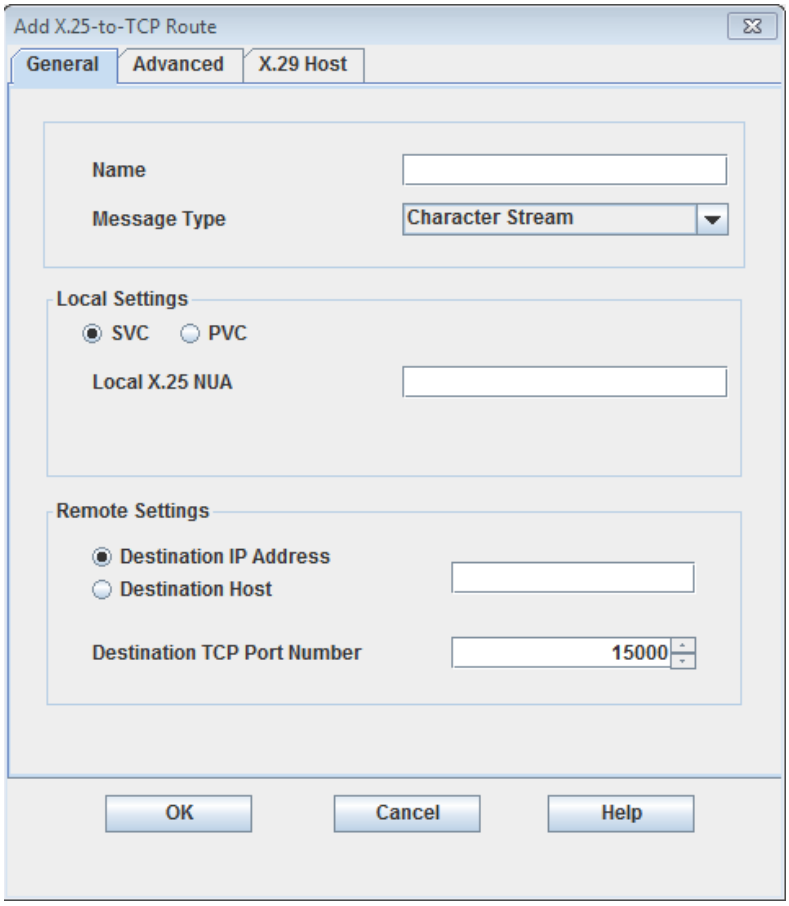
Use the available buttons on the Gateway's X.25-to-TCP routes screen to add, edit, delete or enable/disable the routing entries. Double clicking an existing entry will open it for editing.



Selecting **Add** or **Edit** will open the host entry dialog.

The X.25-to-TCP route parameters are described in the following section.

7.1.1 General Settings



7.1.1.1 Name

This parameter allows an arbitrary name to be associated with this host entry for identification purposes. This field can consist of a string of mixed-case, latin-based alphanumeric characters e.g. London21. Each route must be configured with a unique **Name** value.

7.1.1.2 Message Type

How the traffic on this route is transferred by the Gateway between X.25 and TCP is determined by the configured **Message Type**. The available types and their behaviour are discussed further in Sections 2 and 8. The default type is Character Stream. The available **Message Type** settings are as follows:

Message Type	Description
Character Stream	The X.25 data is treated as a character stream and the packet boundaries are not significant. There is no data transformation between TCP and X.25 – the payloads for each are exactly the same.
Cisco RBP	Cisco Record Boundary Preservation protocol – useful for general purpose interconnection. See Section 8.4 for more details.
DRPD	Dynamic Routing Packetised Data format, compatible with the DRPD Personality Module of the older Windows FarSync TCP-X.25 Gateway. See Section 8.2 for more details.
XDRPD	Extended Dynamic Routing Packetised Data format, a superset of the older DRPD mode. See Section 8.1 for more details.
RFC-1006	RFC-1006 operation – inter-connects with ISO 8073 Transport over X.25, used for OpenFT FTAM and X.400 messaging

X.25 Data Switching over TCP	X.25 Data Switching over TCP – allows X.25 devices to be interconnected via an IP network without the layer 3 overheads associated with XOT (requires a FarLinX X.25 Gateway at each end of the network). See Section 8.5 for more details.
ETX-Terminated	All messages are terminated with the ETX character
CR-Terminated	All messages are terminated with the CR character
Conv hdr 2-bin	The messages are encapsulated within the TCP payload using a 2-byte header, which is the length of the message (excluding the header itself) encoded in binary, in network byte order (most significant byte first). See Section 8.7.1 for more details.
Conv hdr 2-ascii	The messages are encapsulated within the TCP payload using a 2-byte header, which is the length of the message (excluding the header itself) encoded in ASCII (most significant digit first). See Section 8.7.2 for more details.
Conv hdr 4-bin	The messages are encapsulated within the TCP payload using a 4-byte header, which is the length of the message (excluding the header itself) encoded in binary, in network byte order. See Section 8.7.3 for more details.
Conv hdr 4-ascii	The messages are encapsulated within the TCP payload using a 4-byte header, which is the length of the message (excluding the header itself) encoded in ASCII (most significant digit first). See Section 8.7.4 for more details.
OFTP	OFTP operation – inter-connects with RFC-5024 transport over x.25, the TCP payload using a 4-byte header, which is the length of the message encoded in binary, in network byte order (most significant byte first). See Section 8.7.1 for more details

7.1.1.3 Local X.25 NUA / Source Channel

This parameter can be specified either as an NUA (for SVCs) or a channel number (for PVCs). It is used by the Gateway to identify which host entry to use when routing an incoming X.25 call. If the called NUA of the incoming call matches this value then the host address to use for routing this call is then extracted from the entry's configured **Destination Host Name/IP Address** value.

In the case of SVCs, if this value is identical to the configured **Destination NUA** value of an existing SVC route (see Section 6.1.1), this X.25-to-TCP route takes priority.

Note that the Gateway configuration program does not support the configuration of a default destination if the Gateway fails to match the called NUA to the local X.25 NUA of any of the configured host entries.

In the case of PVCs, the source line is always "X25" (see 7.1.1.4).

7.1.1.4 X.25 Line

This parameter only applies when using PVCs. In this case, it is always set to reference the X.25 line since X25-to-TCP routes can only use X.25 (not XOT).

7.1.1.5 Destination Host Name / IP Address

This parameter contains the IP address (or the DNS name) for the host to which the connection should be made.

7.1.1.6 Destination TCP Port Number

This parameter contains the TCP port number that the destination IP Host is listening on.

7.1.1.7 Load balancing

X.25-to-TCP Routes Route by matching: (•) Called NUA ( ) Calling NUA

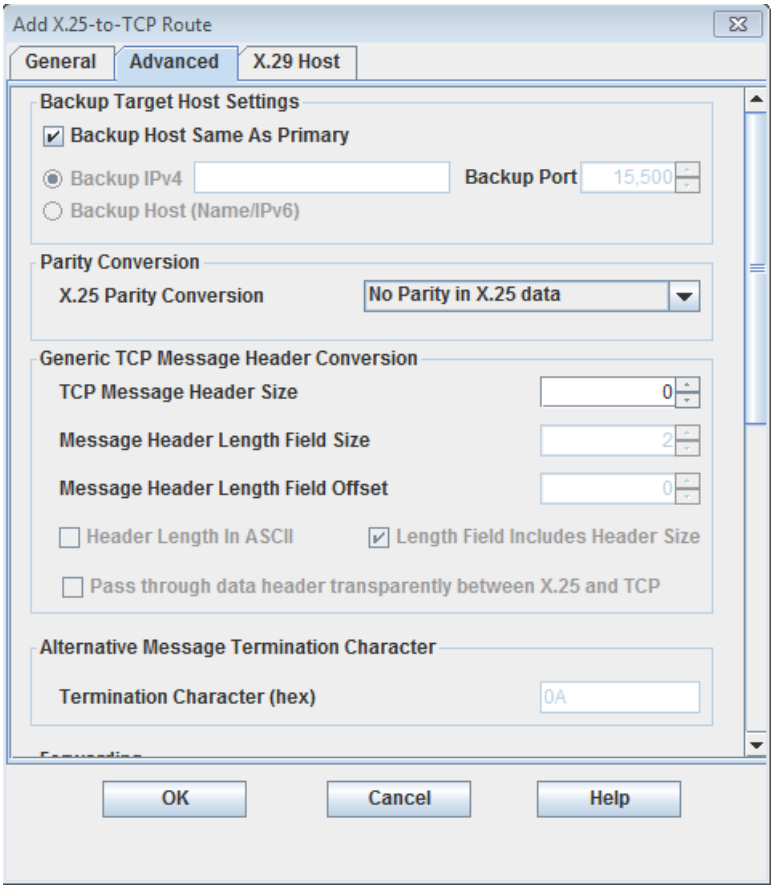
Name	Called NUA	Target Host	Target Port
load_balancing_route1	666	192.168.1.97	15000
load_balancing_route2	666	192.168.1.98	15000
load_balancing_route3	666	192.168.1.100	15000

The Gateway allows load sharing between multiple end points (Gateways or IP Hosts) when using XOT or X.25-to-TCP. In this case, the Gateway allows multiple IP addresses to be defined for a single NUA. When calls are made to that NUA, the connections will be shared between all the IP addresses configured for that NUA, thus providing a fairly basic form of load balancing and resilience.

The actual end point to be called is selected based on the least number of active sessions and the lowest number of failed calls to that end point. When a call to one IP address fails, the 'failed calls count' for that end point will be incremented and another end point will be tried, until all end points have been tried. An end point's 'failed calls count' will be set to 0 when a call to that end point succeeds.

An end point's 'failed calls count' will be reset to 0 after 10 minutes have elapsed since the last failed call.

7.1.2 Advanced Settings



7.1.2.1 Backup Target Host settings

The Gateway supports the configuration of a backup destination to be used if the Gateway fails to establish a TCP connection to the primary target. When an incoming X.25 call arrives, the Gateway makes two attempts to establish a TCP connection to a target host – firstly to the primary and secondly to the backup destination (which can be configured to be the same as the primary)

Backup Host same as primary:
If this checkbox is selected, the backup host settings are the same as the primary target host and the following fields are greyed out:

Backup IP/Host:
A string containing the IP address or hostname of the backup target host.

Backup Port:
The port number that the backup target host is listening on.

7.1.2.2 Parity conversion

The Gateway can remove the parity bit from data received from the X.25 payload and add it to data transmitted out on the X.25 virtual circuit.

The available types of parity conversions are:

- No parity in X.25 data
- Odd parity in X.25 data
- Even parity in X.25 data

7.1.2.3 Generic TCP Message Header conversion

The Gateway can add a message header to messages to be transmitted over the TCP connection and remove the corresponding header from messages received in the TCP payload (using information within the header to determine the message length).

The Gateway uses the setting of the message header size field to override any existing message type configuration. Therefore the **TCP Message Header Size** field should always be set to 0, except when this type of conversion override is explicitly required. If the **TCP Message header Size** value is non-zero then the **Message Type** field is greyed out.

TCP Message Header Size:
Sets the length of the message header

Message Header Length Field Size:
Sets the size of the length field within the header (if the header is larger than the length field – normally the length field occupies the entire header)

Message Header Length Field Offset:
Sets the offset of the length field within the header (normally zero, unless the header is larger than the length field)

Header Length in ASCII:
Enable this option if the length is encoded in ASCII

Length Field Includes Header Size:
Enables this option if the length field includes the size of the header itself

Pass Through data header:
Transparently pass the header with the data between TCP and X.25. This option becomes available when specifying a message type (e.g Conv hdr 2-bin, Conv hdr 2-ASCII, Conv hdr 4-bin, Conv hdr 4-ASCII) that supports TCP headers, or a custom TCP message header size.

Example data received on X.25:
00:1d:06:0b:07:4d:47:2d:30:2e:33:32:09:02:47:48:f1:0a:41:65:72:6f:48:2d:52:43:33:37:c6:01:01
31 bytes total, 2 header + 29 bytes, 001d = 29

Then sent on TCP:
00:1d:06:0b:07:4d:47:2d:30:2e:33:32:09:02:47:48:f1:0a:41:65:72:6f:48:2d:52:43:33:37:c6:01:01
31 bytes total, 2 header + 29 bytes, 001d = 29

Example data received on TCP:
00:04:08:c6:01:02
6 bytes total, 2 header + 4 bytes, 0004 = 4

Then sent on X.25:
00:04:08:c6:01:02
6 bytes total, 2 header + 4 bytes, 0004 = 4

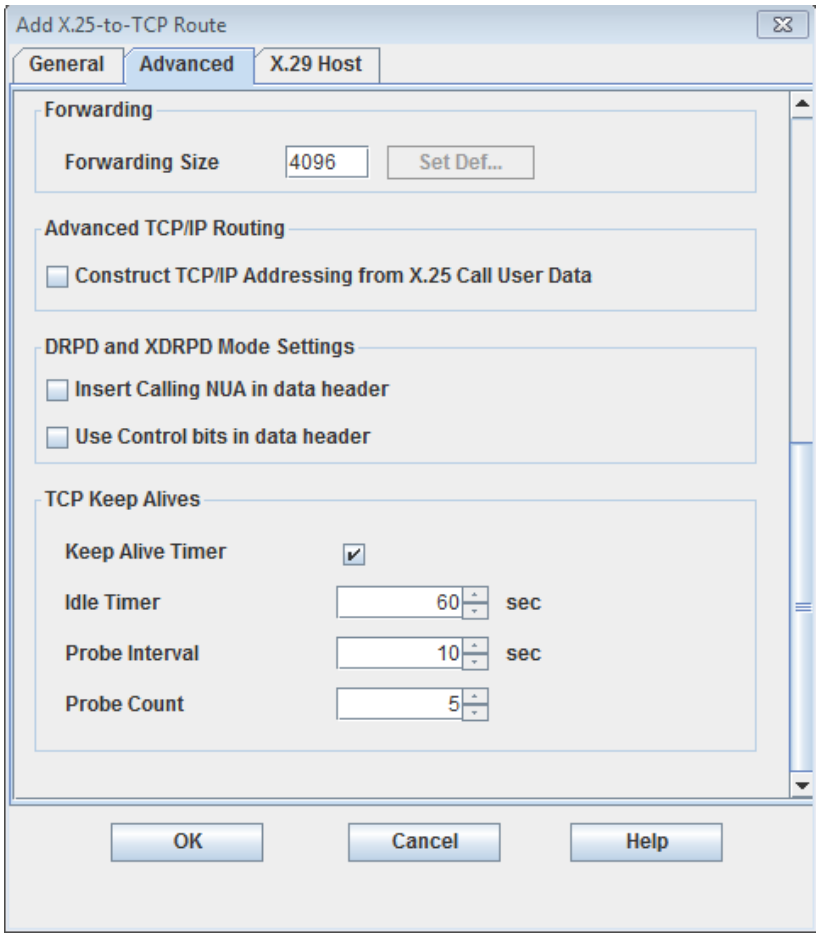
7.1.2.4 Alternative Message Termination Character

If the **Message Type** field is set to **CR Terminate** then an alternative message termination character can be configured.

Value:
ASCII code for termination character in hex
Example:
0d

7.1.2.5 Construct TCP/IP Addressing From X.25 CUD

This option allows the forwarding destination for an X.25-to-TCP route to be determined from the X.25 call user data. It requires the destination IP address and port number to be specially encoded into each X.25 call's user data, which will allow the Gateway to dynamically forward X.25 calls without requiring pre-existing routes to be set up.



The Call User Data in an incoming X.25 call will consist of 6 bytes. These 6 bytes are the destination IP address and target port number for the TCP connection.

Byte	Usage
1	IP address octet 1
2	IP address octet 2
3	IP address octet 3
4	IP address octet 4
5	TCP port high byte
6	TCP port low byte

Example Call User Data received in incoming X.25/XOT connection:

c0:a8:ca:fb:23:32

hex	dec
c0	192
a8	168
ca	202
fb	251

destination IP address = 192.168.202.251

23	35
32	50

Destination TCP port number = 35 * 256 + 50 = 9010

The TCP connection would be made to 192.168.202.251:9010

7.1.2.6 Forwarding Size

The **Forwarding Size** determines the quantity of data that is processed by the Gateway before being forwarded (Note that complete X.25 messages i.e. without the M-bit set, are always forwarded immediately from the X.25 interface, regardless of the configured **Forwarding Size**).

In situations in which transit delays and response times are important, then the **Forwarding Size** should be relatively small (but see note below about X.25 packet sizes). On the other hand, the **Forwarding Size** should be increased to achieve the maximum throughput for bulk data transfer.

Note that there is no benefit in reducing the **Forwarding Size** below the X.25 data packet size used for the connection – the optimum X.25 Forwarding Size for minimum transit delay should be the same size as the X.25 packet size. It is, however, not always possible to predict the packet size when X.25 packet size negotiation is employed, as it will depend upon the remote X.25 equipment. Some tuning of this parameter in line with actual experience may therefore be required.

7.1.2.7 Insert Calling NUA in data header

See section 8.1.1.1 for the use of this parameter.

7.1.2.8 Use control bits in data header

See section 8.1.1.2 for the use of this parameter.

7.1.2.9 TCP Keep Alive

With TCP Keep Alive disabled, in the absence of receiving TCP FIN or RST packets (for example if a cable is unplugged or an intermediate node fails), the FarLinX Gateway will detect the remote disconnection of a TCP session only when it is attempting to send data. Even then, it can take 15 minutes or so to detect the disconnection.

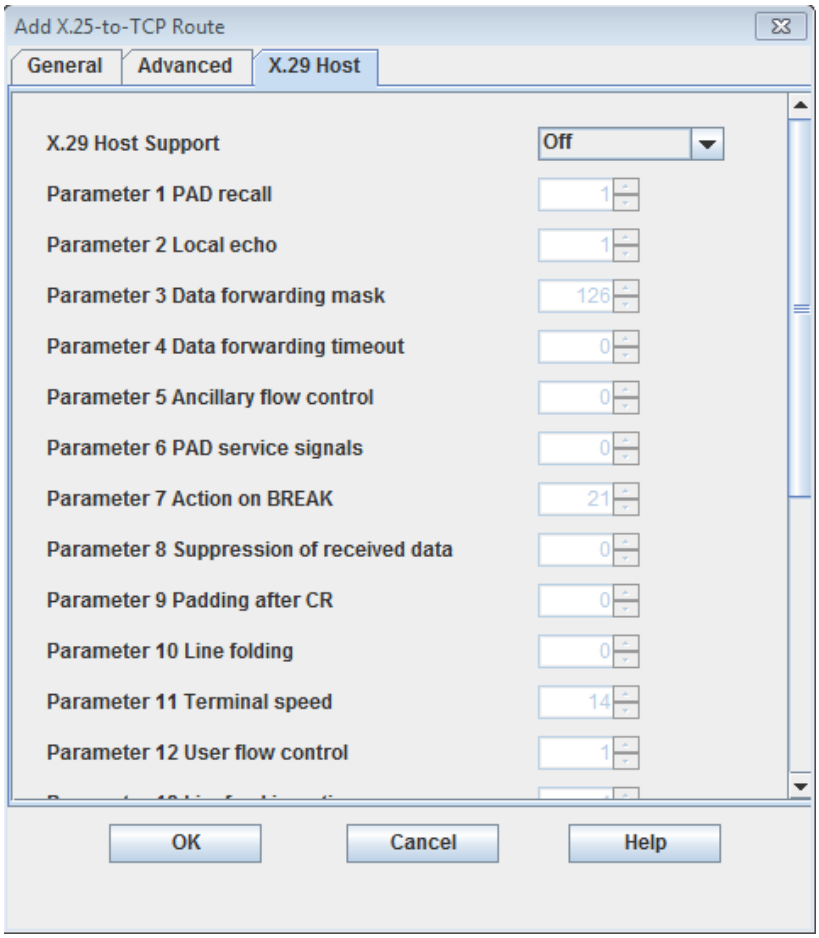
With TCP Keep Alive enabled, the Gateway will be able to detect disconnections within about 3 minutes, even when a session is idle.

It is possible, however, that some remote hosts may not support TCP Keep Alives, in which case enabling TCP Keep Alives might result in idle sessions being disconnected unnecessarily.

When a TCP session is terminated due to no response being received to TCP Keep-Alive probes, a message will be written to the Gateway log similar to the following example:

Connection 6 Closed- TCP Connection closed: Resource temporarily unavailable

7.1.3 X.29 Host



The Gateway can support the control of the X.3 parameters of a remote PAD which is connected to the Gateway over X.25.

This X.29 Host Support has 3 modes:

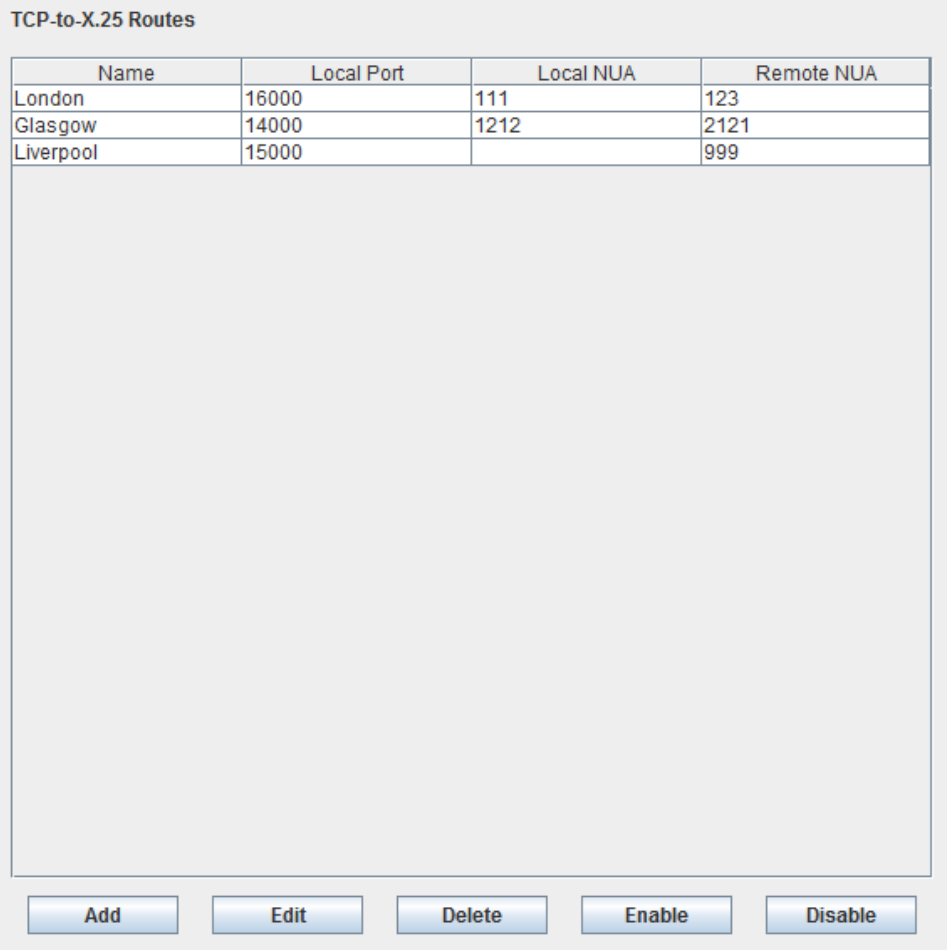
- Default: Off
- Enable on PID: The X.29 set command will be sent when the Call Request includes the X.29 PID in the Call User Data (CUD)
- Enable: The X.29 set command will always be sent.

When X.29 Host Support is enabled, then each of the 22 PAD Parameters specified by the 1988 CCITT recommendation of X.3 will be configurable.

Please refer to RFC1053 for a detailed description of each of the available parameters.

7.2 TCP-to-X.25

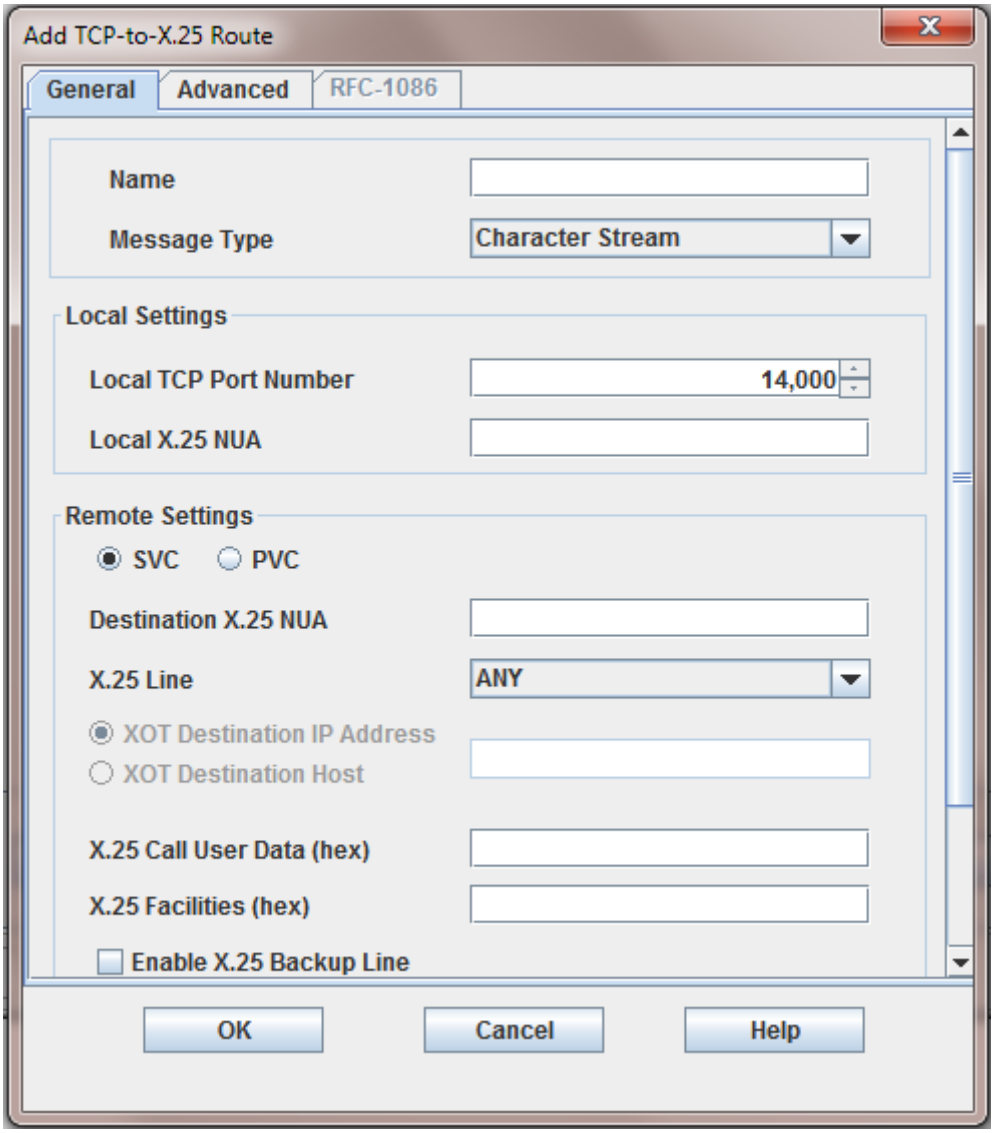
Use the available buttons on the Gateway's TCP-to-X.25 routes screen to add, edit, delete or enable/disable the routing entries. Double clicking an existing entry will open it for editing.



Selecting **Add** or **Edit** will open the host entry dialogue.

The TCP-to-X.25 route parameters are described in the following sections.

7.2.1 General Settings



7.2.1.1 Name

This parameter allows an arbitrary name to be associated with this host entry for identification purposes. This field can consist of a string of mixed-case, latin-based alphanumeric characters e.g. London11. Each route must be configured with a unique **Name** value.

7.2.1.2 Message Type

How the traffic on this route is transferred by the Gateway between TCP and X.25 is determined by the configured **Message Type**. The available types and their behaviour are discussed further in Sections 2 and 8.

The default type is Character Stream. The available **Message Type** settings are as follows:

Message Type	Description
Character Stream	The X.25 data is treated as a character stream and the packet boundaries are not significant. There is no data transformation between TCP and X.25 – the payloads for each are exactly the same.
Cisco RBP	Cisco Record Boundary Preservation protocol – useful for general purpose interconnection. See Section 2.3.1 for more details.

DRPD	Dynamic Routing Packetised Data format, compatible with the DRPD Personality Module of the older Windows FarSync TCP-X.25 Gateway. See Section 8.2 for more details.
XDRPD	Extended Dynamic Routing Packetised Data format, a superset of the older DRPD mode. See Section 8.1 for more details.
RFC-1006	RFC-1006 operation – inter-connects with ISO 8073 Transport over X.25, used for OpenFT FTAM and X.400 messaging
X.25 Data Switching over TCP	X.25 Data Switching over TCP – allows X.25 devices to be interconnected via an IP network without the layer 3 overheads associated with XOT (requires a FarLinX Gateway at each end of the network). See Section 8.4 for more details.
ETX-Terminated	All messages are terminated with the ETX character
CR-Terminated	All messages are terminated with the CR character
Conv hdr 2-bin	The messages are encapsulated within the TCP payload using a 2-byte header, which is the length of the message (excluding the header itself) encoded in binary, in network byte order (most significant byte first). See Section 8.6.1 for more details.
Conv hdr 2-ascii	The messages are encapsulated within the TCP payload using a 2-byte header, which is the length of the message (excluding the header itself) encoded in ASCII (most significant digit first). See Section 8.6.2 for more details.
Conv hdr 4-bin	The messages are encapsulated within the TCP payload using a 4-byte header, which is the length of the message (excluding the header itself) encoded in binary, in network byte order. See Section 8.6.3 for more details.
Conv hdr 4-ascii	The messages are encapsulated within the TCP payload using a 4-byte header, which is the length of the message (excluding the header itself) encoded in ASCII (most significant digit first). See Section 8.6.4 for more details.
OFTP	OFTP operation – inter-connects with RFC-5024 transport over x.25, the TCP payload using a 4-byte header, which is the length of the message encoded in binary, in network byte order (most significant byte first). See Section 2.5 for more details
RFC-1086	RFC-1006 operation – A host on an IP based internet can communicate with hosts on X.25 based networks providing the hosts are running ISO protocols. This also supports an optional Authentication procedure with a Radius server.

7.2.1.3 Local TCP Port Number

The Gateway listens on a unique TCP port number for each host that is configured. That value is configured in this parameter.

7.2.1.4 Local X.25 NUA

When configured, this parameter is used as the calling NUA in the X.25 Call Request made by the Gateway to this host when a corresponding incoming TCP connection request is received.

7.2.1.5 Destination X.25 NUA / Channel

In the case of SVCs, this parameter is used to specify the NUA of the X.25 target to be called by the Gateway when a corresponding incoming TCP connection request is received. When using PVCs this parameter should instead contain the PVC logical channel number.

7.2.1.6 X.25 Line

This parameter determines the line to be used for the outgoing X.25 call. In the case of SVCs, this can specify the physical X.25 line or XOT. If XOT is selected then the Destination IP/Hostname must also be configured. For PVCs, only the physical X.25 line can be used.

7.2.1.7 X.25 Call User Data

If it is necessary to include Call User Data in the X.25 Call Request to the host, this parameter can be configured. The Call User Data string needs to be configured in hexadecimal format.

7.2.1.8 X.25 Facilities

Facilities only need to be configured if X.25 particular facilities, such as Closed User Groups or Network User Identification, are required by the host or the X.25 Network. They are defined in hexadecimal format and are encoded as they would be in the X.25 call itself.

Common facilities are coded as follows:

throughput class negotiation 02 <n>
closed user group selection:
basic format 03 <n>
extended format 47 <n> <n>
closed user group (outgoing access):
basic format 09 <n>
extended format 48 <n> <n>
bilateral closed user group 41 <n> <n>
reverse charging 01 01
NUI selection C6 <len> <nui string>
charging information request 04 <n>
RPOA selection:
basic format 44 <n> <n>
extended format C4 <len> <RPOA>
transit delay selection 49 <n> <n>

Other non-standard facilities may be included, providing that they follow the appropriate options marker, 00 <n>, and appear after all X.25 network facilities.

7.2.1.9 Enable X.25 Backup Line

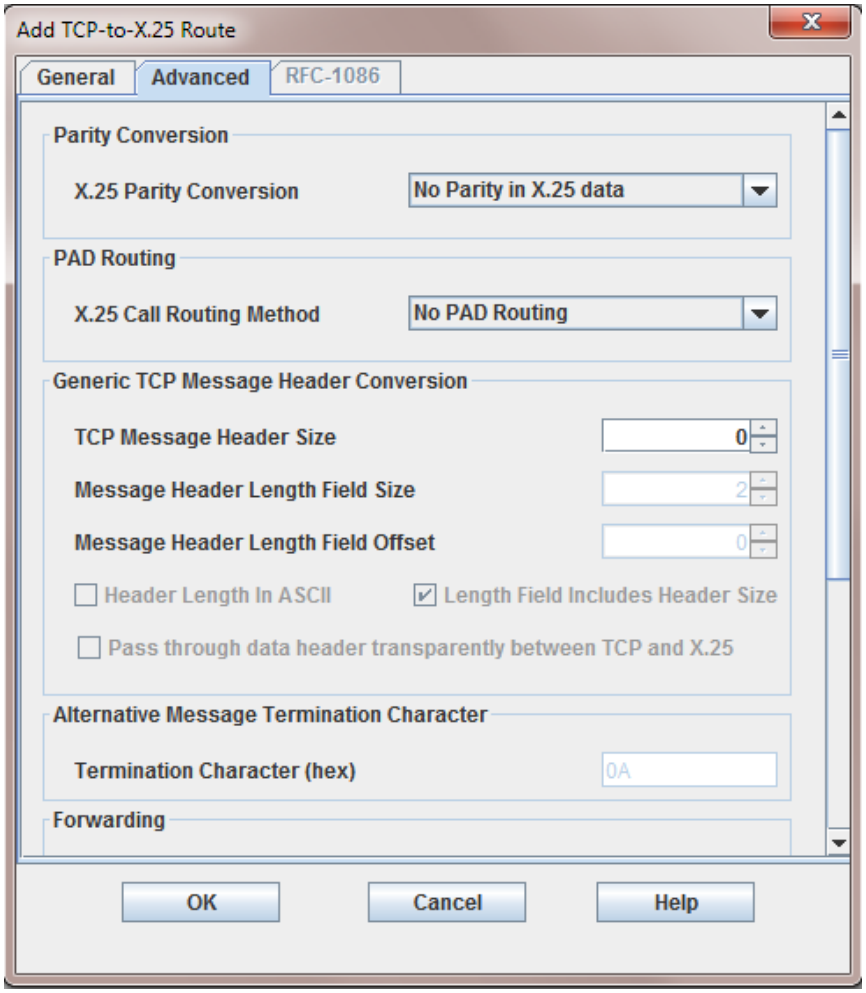


A backup line configured in the TCP-X.25 route only takes effect when the X.25 Line configured in this route is not up and the Activate on Demand option (configured in the Layer1) is not enabled.

When a TCP call is made, if the configured X.25 Line of this TCP-X.25 route is up, the Gateway will transfer the incoming TCP call to the configured X.25 Line, regardless of whether the backup line is configured. If the configured X.25 line is not up and the Activate on Demand option is not enabled, the incoming call will be transferred to the configured X.25 Backup Line if it has been enabled.

If the Activate on Demand option has been enabled on the configured line, the Gateway will ignore the line state. It will always try to activate the configured line and never fall back to the backup, even if it is enabled.

7.2.2 Advanced Settings



7.2.2.1 Parity conversion

The Gateway can remove the parity bit from data received from the X.25 payload and add it to data transmitted out on the X.25 virtual circuit.

The available types of parity conversions are:

- No parity in X.25 data
- Odd parity in X.25 data
- Even Parity in X.25 data

7.2.2.2 PAD Routing – X.25 Call Routing Method

As an alternative to explicit static routing, PAD or dynamic routing may be employed to allow the TCP client application to specify the call parameters (Destination X.25 DTE Address, Call User Data) within the data stream. Refer to Section 8.1.10 for more details.

When using PAD Routing, the Call User Data is specified in ASCII and follows the X.29 PID [01000000] in the X.25 Call Request packet.

Call Address Block Routing is similar to PAD routing, but the X.25 Address format is slightly different. There is the choice of whether or not the X.29 PID is included in the X.25 Call Request packet.

DRPD allows dynamic routing to be used with message formats other than the DRPD message format – for example, when configuring the fixed TCP header conversion.

Shell Routing is currently reserved.

Default: No PAD Routing (i.e. just use Static Routing)

7.2.2.3 Generic TCP Message Header conversion

The Gateway can add a message header to messages to be transmitted over the TCP connection and remove the corresponding header from messages received in the TCP payload (using information within the header to determine the message length).

The Gateway uses the setting of the message header size field to override any existing message type configuration. Therefore the **TCP Message Header Size** field should always be set to 0, except when this type of conversion override is explicitly required. If the **TCP Message header Size** value is non-zero then the **Message Type** field is greyed out.

TCP Message Header Size:
Sets the length of the message header

Message Header Length Field Size:
Sets the size of the length field within the header (if the header is larger than the length field – normally the length field occupies the entire header)

Message Header Length Field Offset:
Sets the offset of the length field within the header (normally zero, unless the header is larger than the length field)

Header Length in ASCII:
Enable this option if the length is encoded in ASCII

Length Field Includes Header Size:
Enable this if the length field includes the size of the header itself

7.2.2.4 Alternative Message Termination Character

If the **Message Type** field is set to **CR Terminate** then an alternative message termination character can be configured.

Value:
ASCII code for termination character in hex
Example:
0d

7.2.2.5 Forwarding Size

The **Forwarding Size** determines the quantity of data that is processed by the Gateway before being forwarded. (Note that complete X.25 messages i.e. without the M-bit set, are always forwarded immediately from the X.25 interface, regardless of the configured **Forwarding Size**).

In situations in which transit delays and response times are important, then the **Forwarding Size** should be relatively small (but see note below about X.25 packet sizes). On the other hand, the **Forwarding Size** should be increased to achieve the maximum throughput for bulk data transfer.

Note that there is no benefit in reducing the **Forwarding Size** below the X.25 data packet size used for the connection – the optimum **Forwarding Size** for minimum transit delay should be the same size as the X.25 packet size. It is, however, not always possible to predict the packet size when X.25 packet size negotiation is employed, as it will depend upon the remote X.25 equipment. Some tuning of this parameter in line with actual experience may therefore be required.

7.2.2.6 Use control bits in data header

See section 8.1.1.2 for the use of this parameter.

7.2.2.7 TCP Keep Alive

With TCP Keep Alive disabled, in the absence of receiving TCP FIN or RST packets (for example if a cable is unplugged or an intermediate node fails), the FarLinX Gateway will detect the remote disconnection of a TCP session only when it is attempting to send data. Even then, it can take 15 minutes or so to detect the disconnection.

With TCP Keep Alive enabled, the Gateway will be able to detect disconnections within about 3 minutes, even when a session is idle.

It is possible, however, that some remote hosts may not support TCP Keep Alives, in which case enabling Keep Alives might result in idle sessions being disconnected unnecessarily.

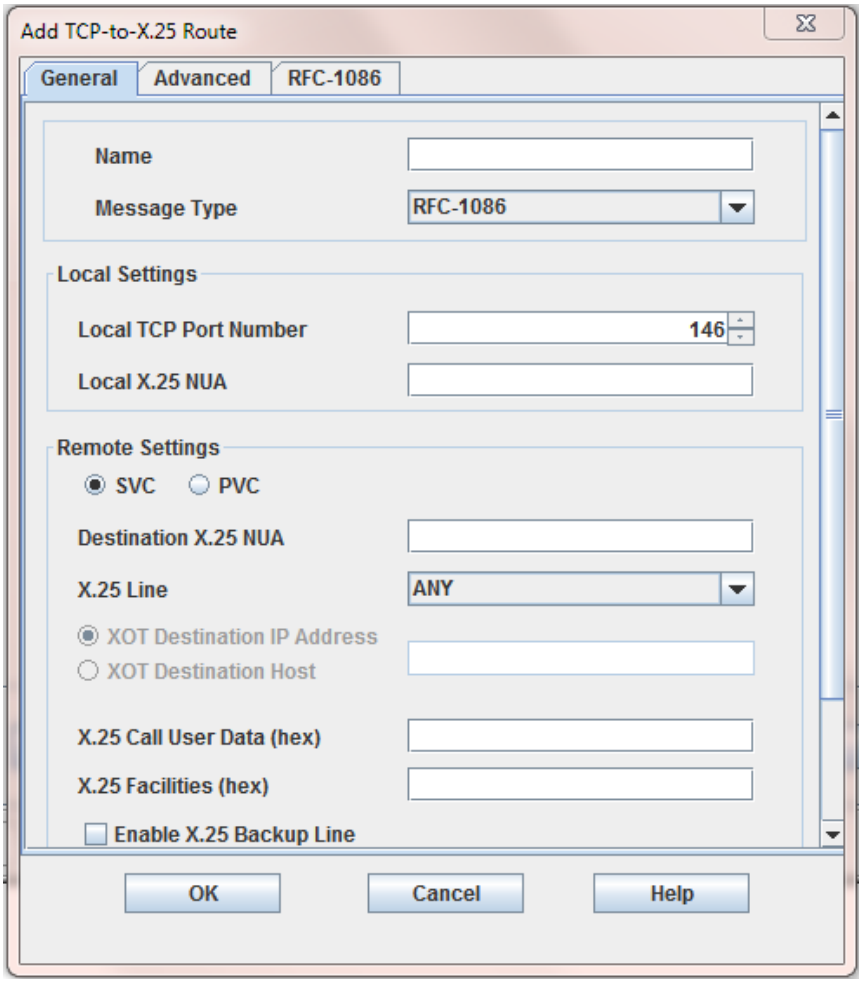
Note that it is strongly recommended to use keep-alives in conjunction with PVCs so that they can be re-connected after un-notified disconnections.

When a TCP session is terminated due to no response being received to TCP Keep-Alive probes, the Gateway log will be as follow example:

16:43:24:GW2: Connection 6 Closed- TCP Connection closed: Resource temporarily unavailable

7.3 RFC-1086 Settings

The third tab of the TCP-to-X.25 Route dialog box is where RFC-1086 settings are made. This tab is normally grayed out, and will only become editable when a message type of RFC-1086 has been selected on the General tab.



7.3.1 RFC-1086 Overview

The FarLinX Gateway supports only the first of the two possible functions as described in RFC-1086. Only the function value = 1 is supported, i.e.to make an X.25 call from information received in a Registration packet. In additions to this the FarLinX Gateway also supports an optional authentication stage, where an Identification Block may precede the Connection Block, and information from this block is used to construct a Radius authentication request to a nominated Radius server. These initial data flows are illustrated below:

Registration Connection Block



01				Registration
00	03			Address type (03 = X25)
xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx			LB	X21 Address (16 bytes)
xx xx xx xx	LB			Protocol ID (4 bytes)
xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx			LB	Call User Data (16 bytes)
xx xx xx xx xx xx	LB			X.25 Facilities (6 bytes)

The fields are all fixed length fields, and thus the total block length is 49 bytes. The LB bytes are an indication of the number of bytes in that field that contain relevant data.

The connection block will always start with a byte of 0x01, followed by two bytes of address type, which for X.25 will always be 0x00 and 0x03.

The next field is of a fixed length of 16 bytes, and will contain the Called NUA address. The field is followed by a length byte marking how many bytes of NUA are present.

The next field is of a fixed length of 4 bytes, and will contain a Protocol Id followed by length byte marking how many bytes of Protocol Id are present.

The next field is of a fixed length of 16 bytes, and will contain the Cal User Data. The field is followed by a length byte marking how many bytes of Call User Data are present.

The final field is of a fixed length of 6 bytes, and may contain X.25 facilities followed by a length byte marking how many bytes of Facilities are present.

The FarLinX Gateway will process the X.21 address for use as the Called NUA, and if there is any Call User Data and Facilities present, these are also used in the C.25 Call Request.

The Protocol Id field is not processed.

Optional Identification Block



04				Special Registration
01				Protocol Version
A B C D E F G H I J			LB	Not Processed (10 bytes)
A B C D E F G H I J			LB	Not Processed ASCII (10 bytes)
A B C D E F G H I J K L M N O P Q R S T			LB	Username ASCII (20 bytes)
A B C D E F G H I J K L M N O P Q R S T			LB	Password ASCII (20 bytes)
0	0			Interface

If the Identification block is present it will be marked by the first byte of the data following the connection having a value of 0x04. If there is no Identification block, then the first byte of the data will be 0x01.

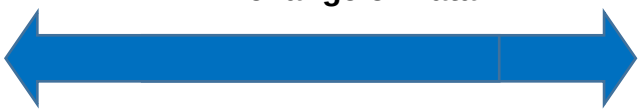
The fields are all fixed length fields, and thus the total block length is 68 bytes. The LB bytes are an indication of the number of bytes in that field that contain relevant data.

The two fields following the Protocol Version are sometimes used to contains further address information about the originator of the RFC-1086 message. But these are not processed by the Gateway.

The next two fields of this block will contain a Username and a Password, which are used in a request to a Radius Server for validation.

The last field defines a Interface type, but are not processed by the Gateway.

Exchange of Data



03	00	Address type (03 = X25)
LH	LL	Data length (= length of the frame, including the header bytes)
xx xx xx xx xx xx xx xx xx xx		Data

When the FarLinX has established the X.25 call, the TCP data will contain normal RFC-1008 headers. These are processed and removed before they are forwarded as packets on the X.25 interface. And data packets received on the X.25 interface will have an RFC-1008 header pre-pended to them before being forwarded on the TCP interface.

For a route that has been configured with a message type of RFC-1086, then the first TCP/IP data bytes is examined the check for the optional Identification block. The byte is checked for the following values:

1) = 0x01, then what follows is a registration connection block.
2) = 0x04, then what follows is an Identifcation block then a connection block.
3) Any other value is an error, and the TCP connection is cleared.

7.3.2 Radius Authentication

When a Radius authentication request is made but is declined, then the TCP call is cleared. If the authentication request succeeds OR if the request times out, then the X.25 call is made using the information in the Registration Connection Block.

7.3.3 RFC-1086 Configuration

The RFC-1086 parameters are shown below:

7.3.3.1 Connection Retry

- **Enable**
  If this option is selected, then the FarLinX Gateway will try and establish the X.25 connection more than once.

- **Number of retries**
  The number of times a connection attempt will be made before clearing the RFC-1086 connection attempt.

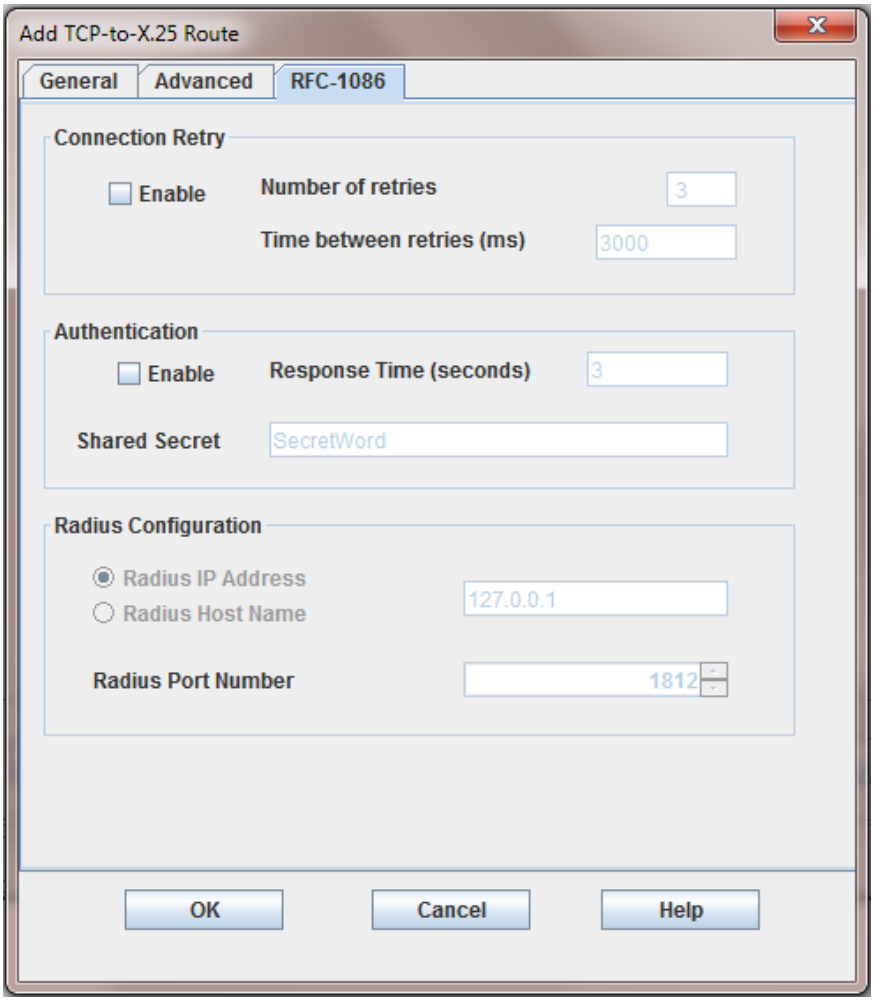
- **Time between retries**
  The time delay between connection attempts.

7.3.3.2 Authentication

- **Enable**
  If this option is selected and the RFC-1086 Identification Block has been received, then and authentication request is made to a Radius server.

- **Response Time**
  This value (in seconds) is the amount of time that the FarLinX Gateway will wait for a response from the server. If there is no response then it will be assumed that no authentication was required after all, and the X.25 connection will proceed.

- **Shared Secret**
  The shared secret used in the authentication request.



7.3.3.3 Radius Configuration

- **Radius Server IP Address/Hostname selector**
  A radio button that selects if the associated field is an IP address or a host name.

- **IP address or Hostname**

- **Radius Port Number**
  The port number to be used in the Radius Authentication request.

7.4 Configuration of POS Extensions

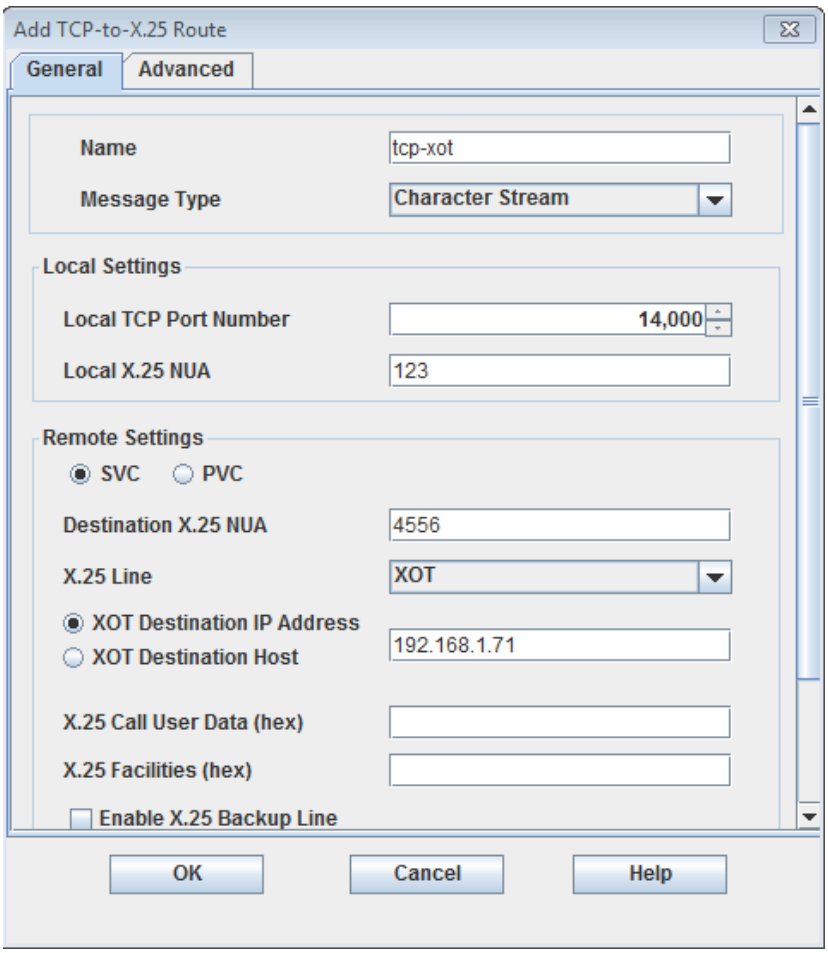
The POS extension to the Gateway operation provides the following additional POS related message types and conversions. Note that these extensions are only available with a suitable license.

Message Type	Description
APACS TPAD-Host	APACS messages are sent without an LRC in PAD-HOST mode and no APACS control frames (ENQ, ACK, NAK, DLE EOT) are required. For more information, see Section 2.3.1
APACS T/T-TPAD	APACS messages are sent with the LRC in PAD-HOST mode and APACS control frames (ENQ, ACK, NAK, DLE EOT) are utilized. For more information, see Section 2.3.1
APACS CTL-Online Hdr-3	The APACS messages are encapsulated within the TCP payload using a 3-byte header, the first byte of which is a recognition character (0xff), and the remaining 2 bytes are the length of the message, excluding the size of the header itself, encoded in binary in network byte order (most significant byte first). In addition, the parity bit is removed from each byte from the incoming X.25 data stream.
APACS CTL-Online Hdr-2	The APACS messages are encapsulated within the TCP payload using a 2-byte header, which contains the length of the message, excluding the size of the header itself, encoded in binary in network byte order (most significant byte first). In addition, the parity bit is removed from each byte from the incoming X.25 data stream.
APACS Conv-TPAD	APACS TPAD conversion is performed – the T/T-TPAD format is used on the incoming connection side and the TPAD-Host format used on the outgoing side of the connection.
APACS Conv-HPAD	APACS HPAD conversion is performed – the T/T-TPAD format is used on the outgoing connection side and the TPAD-Host format used on the incoming side of the connection.
ISO 8583 Hdr2	The ISO 8583 messages are encapsulated within the TCP payload using a 2-byte header – the header is a 2-byte message length, including the size of the header itself, encoded in binary in network byte order. This header is removed when forwarding the messages over the X.25 virtual circuit.
ISO 8583 Hdr4	The ISO 8583 messages are encapsulated within the TCP payload using a 4-byte header – the header is a 4-byte message length, including the size of the header itself, encoded in binary in network byte order (most significant byte first).
ISO 8583 (CTL)	The ISO 8583 messages are encapsulated within the TCP payload using a 2-byte header – the message length excludes the header size (the header is actually the same format as for APACS CTL-Online messages).
SIBS	The Portuguese Asynchronous POS Communication protocol format used by SIBS, with PAD-like call setup, where the client device supplies the destination DTE address.
HGEPOS	The HGEPOS message format

7.5 TCP-to-XOT

The FarLinX X.25 Gateway can also support TCP routing to XOT, for SVCs only.

When selecting a TCP-to-XOT route the X.25 Line option must be set to **XOT** and the **Destination X.25 NUA** and **XOT Destination IP** address (or **XOT Destination Host Name**) values must be configured.



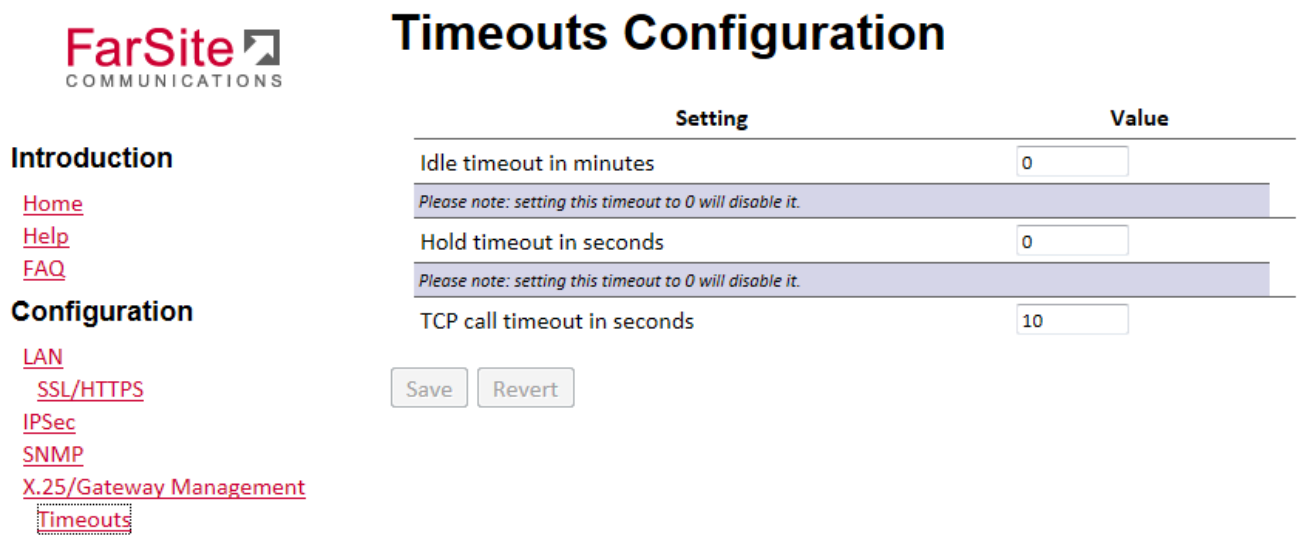
Click OK to save the route definition. The Gateway will check whether an XOT SVC route already exists for the specified NUA (see Section 6.1). If there is no existing route then one will be created.

TCP-to-XOT operation does not support PVCs or Dynamic Routing. As a result the following message types are not supported with TCP-to-XOT routing:

- DRPD
- XDRPD

7.6 Timeouts

The gateway provides a list of timers for special requirements of handling the TCP and X.25 connections. Click the link “Timeouts” to configure these values.



7.6.1 Idle timeout

When set, this parameter specifies the number of minutes after which an idle session is automatically closed. An idle session is any established session with no data being transferred. Use a value of zero here to disable this feature.

7.6.2 Hold timeout

When set, this parameter specifies the number of seconds the Gateway should hold open a session when one side closes. For example, if the session is closed on the TCP side, the Gateway can hold the X.25 session open for a number of additional seconds. This feature can allow the Gateway to complete its transmission on the open side of the session. A value of zero here means that both sides of the session are immediately closed.

7.6.3 TCP call

When set, this parameter specifies the number of seconds before a TCP call from the Gateway to a remote TCP host will abort due to timing out. It is used for X.25/XOT to TCP sessions, when the Gateway connects to TCP server, in order to avoid long connection times.

8 DEVELOPER REFERENCE

This Section of the manual is for those who wish to develop their own application to interface to X.25 networks via the Gateway over TCP/IP.

The XDRPD (and older DRPD) message types can be used by applications wishing to employ dynamic routing – that is, the TCP/IP client provides some or all of the X.25 Call parameters. XDRPD (and DRPD) can also be used by server applications needing to know details from the Incoming X.25 Call.

PAD and Call Address Block routing provide a more limited way of controlling the X.25 Call parameters for TCP clients – just the destination X.25 address and Call User Data fields can be controlled. These routing methods can, however, be used in conjunction with various different message types.

Other modes of connection described in this Section can only be used with static routing (where all the X.25 connection parameters are configured within the Gateway), or in conjunction with Call Address Block routing.

8.1 XDRPD (Extended Dynamic Routing with Packetised Data)

XDRPD is a superset of the older DRPD mode (see Section 8.2), providing additional in-band control and status messages and also providing TCP/IP Servers the opportunity to refuse incoming connections. It is recommended that XDRPD mode be used in preference to DRPD mode.

Note that by using a different number space for XDRPD messages from that of DRPD messages, the Gateway is able to switch from one mode to the other. Switching mode happens on receipt of the first message from the host and so is applicable only for connections between TCP Clients and the Gateway. In other words, when a connection is established to a DRPD or XDRPD listening socket, the Gateway is able to determine which connection mode to use by whether it receives the DRPD Connect Request message or the XDRPD X-C message. It therefore makes no difference as to whether XDRPD or DRPD is configured for TCP-to-X.25 Routes, but the correct message type must be selected for X.25-to-TCP Routes.

It is possible for the TCP/IP Host to specify the X.25 Clearing Cause and Diagnostic codes when disconnecting by using the X-F message. Similarly for PVCs, it is possible for the TCP/IP Host to Reset the PVC, specifying the Reset Cause and Diagnostic Codes, by using the X-R message.

Message Primitives

The following message primitives are supported (all of which are bidirectional):

0x00:	Data
0x10:	X-C (Connect)
0x11:	X-A (Accept)
0x12:	X-F (Finish)
0x13:	X-R (Reset)
0x14:	X-P (PVC Connect)

8.1.1 Message Format

The XDRPD message format is the same as DRPD message format. Each message in the TCP data stream has a 4-byte header containing:

- (2 bytes) length field, in network byte order (i.e. most significant byte first).
  This contains the length of the message body (i.e. not including the size of the 4-byte header)
- (1 byte) message type field
- (1 byte) spare field (must be zero)

See sections 8.1.1.1 and 8.1.1.2 for variants of this header usage, depending on configuration options chosen.

The message element parameters in the non-data message bodies are encoded in ASCII, separated by commas. The parameter is self-identifying as follows:

$C:<cause & diagnostic codes>
$D:<destination NUA>
$E:<error explanation>
$F:<facilities>
$O:<origination NUA>
$P:<PVC channel number>
$U:<call user data>
$R:<PVC established by Reset, with cause and diagnostic codes>
$X:<line type: 0=physical, 1=XOT>

Note that the $A, $L and $N parameters, although applicable on the multi-port versions of the FarLinX Gateway, are not used by the FarLinX X.25 Gateway since it is a single (sync) port device – these parameters should not be included in messages sent to the FarLinX X.25 Gateway.

The parameters apply to each message type as follows:

To Gateway:	From Gateway:
X-C: $D, $F, $X, $O, $P, $U	**X-C**: $D, $X, $O, $P, $U
X-P: $P, $X	**X-P**: $P, $X, $R
X-A: (none)	**X-A**: (none)
X-F: $C	**X-F**: $C, $E
X-R: $C	**X-R**: $C

All parameters are ASCII strings; the Facilities and Call User Data are specified in hexadecimal. The Gateway converts the strings into the appropriate binary representation.

8.1.1.1 Insert Calling NUA in data header

If the configuration option ***Insert Calling NUA in data header*** has been selected then header usage is amended as follows:

- (2 bytes) length field, in network byte order (i.e. most significant byte first). This contains the length of the message body (i.e. not including the size of the 4-byte header; the length does not include the inserted calling NUA length)
- (1 byte) message type field (this is used to determine whether the message is data to/from the X.25 network, or a dynamic routing message)
- (1 byte) calling NUA length is defined in low-order bits. When this value is zero, the message is treated like a normal XDRPD/DRPD message. When this value is non-zero, it contains the length of the calling NUA that has been inserted into the header, before the data

The Gateway handles this message in the same way as an XDRPD or DRPD message. When the Gateway receives characters from the TCP data stream, it assembles the header and removes the calling NUA from the message body field and then calculates the message body size.

When it receives a message from the X.25 network, the Gateway constructs the header by inserting the calling NUA into the message body and then forwards the header and the message body over the TCP data stream.
For example:
In this scenario, the calling NUA is 54898. The DRPD message contains:

DRPD message without calling NUA:
00 07 00 00 24 50 3a 34 30 30 31

DRPD message having inserted calling NUA **54898**:
00 07 00 05 **35 34 38 39 38** 24 50 3a 34 30 30 31

8.1.1.2 Use control bits in data header

If the configuration option ***Use control bits in data header*** has been selected then header usage is amended as follows:

- (2 bytes) length field, in network byte order (i.e. most significant byte first). This contains the length of the message body (i.e. not including the size of the 4-byte header) If used with the Insert Calling NUA option,

this length does not include the inserted calling NUA length
- (1 byte) message type field (this is used to determine whether the message is data to/from the X.25 network, or a dynamic routing message)
- (1 byte) The control bit is defined in high-order bits (binary).
  0000 XXXX: Data
  0001 XXXX: Q bit
  0010 XXXX: reserve
  0100 XXXX: reserve
  1000 XXXX: reserve

When it receives a message from the X.25 network, the Gateway constructs a header by detecting whether the specified bit has been set on the message body and then forwards the header with the specified bit set over the TCP data stream.

Example:

Data, len=1, id=4, Rx X.25 Complete without Q bit
31
Data, len=5, id=4, Tx TCP Stream
0001000031
Data, len=44, id=4, Rx X.25 Complete with Q bit
0365000000000000-0000000000000000-0000000000000000-0000000000000000
0000000000000000-00000000
Data, len=48, id=4, Tx TCP Stream
002c001003650000-0000000000000000-0000000000000000-0000000000000000
0000000000000000-0000000000000000

8.1.2 SVC TCP-to-X.25 Operation

The Client uses the X-C message primitive (0x10) to request that the Gateway send an outgoing X.25 Call, specifying the call parameters. The $D parameter is mandatory; the others ($O, $F and $U) are optional.

Once the X.25 call has been established, the Gateway provides a positive acknowledgment, by sending the X-A message primitive (0x11). The client would typically wait for this acknowledgement before transmitting a data message, but if it wishes it may do so immediately following the connect request – this will get stored until the connection has been established and will then be sent out immediately the call is connected.

Should the Gateway fail to establish the X.25 connection, the Gateway will transmit the X-F message primitive (0x12), providing information as to why the X.25 call failed.

Successful SVC Connection initiated by TCP client:

```
TCP Client              Gateway
    ---------- X-C -------->
  <---------- X-A --------
    ---------- Data ------->
  <---------- Data -------
```

Unsuccessful SVC Connection initiated by TCP client:

```
TCP Client              Gateway
    ---------- X-C ------->
  <---------- X-F -------
```

For example, to call NUA 12345 with the Closed User Group facility 99 and a Call User Data string of “HELLO”, the message body would contain:

$D:12345,$F:0399,$U:48454C4C4F

The complete message (in hex) would be:

00 1E 10 00 24 44 3A 31 32 33 34 35 2C 24 46 3A 30 33 39 39 2C 24 55 3A 34 38 34 35
34 43 34 43 34 46

SVC X.25-to-TCP Operation
Incoming X.25 Calls are routed in the same was as other types of connections handled by the Gateway – it is necessary to configure a destination IP Address within the X.25-to-TCP Route.

SVC Connection initiated by X.25 client and accepted by Host

```
TCP Server              Gateway
    <---------- X-C --------
     ---------- X-A -------->
```

SVC Connection initiated by X.25 client but rejected by Host

```
TCP Server              Gateway
    <---------- X-C --------
     ---------- X-F -------->
```

Note that the Gateway does not accept the X.25 incoming call until after it receives the X-A from the TCP Server. The Gateway will clear the call if the TCP Server sends an X-F message, or if the TCP connection is closed. The Gateway also clears the call if the TCP connection fails to be established.

When the TCP connection is established, the X-C message sent by the Gateway will contain the Called Address, Calling Address and Call User Data (if required).

$D:<Called Address>,$O:<Calling Address>,$U:<Call User Data (in hex)>

8.1.3 SVC Connection Closure initiated by TCP/IP host

The TCP Host may optionally use the X-F message to close a connection – this allows it to specify clearing cause and diagnostic codes. If it does not wish to do so, then it may simply close the TCP connection.

For example, to clear the call using Cause code 0xab diagnostic code 0xcd, the message body would contain:
$C:ABCD

The complete message (in hex) would be:
00 07 12 00 24 43 3A 41 42 43 44

8.1.4 PVC TCP-to-X.25 Operation

A TCP Client can also use the X-C primitive to establish a PVC connection, in the same way as for an SVC, but also using the $P message element to select the PVC LCN.

TCP Client Gateway
---------- X-C -------->
<---------- X-A --------
---------- Data ------->
<---------- Data -------

However, for consistency with the Gateway-to-Server direction, it is also possible to use the X-P primitive, in which case there is no X-A response: the connection moves to the Open state immediately.

TCP client Gateway
---------- X-P --------->
---------- Data -------->

In addition to the $P parameter, the $X parameter can be used to specify whether the call should be made on the physical line or over XOT.

For example, to attach to PVC channel 1 on the physical X.25 line, the message body would contain:
$P:1

The complete message (in hex) would be:
00 04 03 00 24 50 3a 31

8.1.5 PVC X.25-to-TCP Operation

When an X.25-to-PVC Route is configured to be attached to a PVC, the Gateway will initiate the TCP connection to the TCP server as a result of receiving either a Reset Indication packet or a Data packet. In the case of a Reset packet, the Gateway will include the $R element in the X-P message.

X.25 reset packet caused the connection to be initiated:
TCP Server Gateway
<---------- X-P ---------
<---------- X-R ---------

X.25 data caused the connection to be initiated:
TCP Server Gateway
<---------- X-P ---------
<---------- Data --------

Note that it is possible for X.25 Data and Reset events to be received whilst the Gateway is establishing the TCP connection to the Server. Data events are queued up to be forwarded, but a received Reset Indication causes previously queued Data and Reset events to be discarded; then once the TCP connection is complete, the Gateway would send the X-P and X-R messages with the cause and diagnostic codes from the most recently received Reset Indication.

For example, for a connection on the physical X.25 line, when the connection was initiated as a result of the receipt of a Data packet on channel 4001, the X-P message body would be:

$P:4001

The complete message (in hex) would be:
00 07 14 00 24 50 3A 34 30 30 31

8.1.6 PVC Connection Closure initiated by TCP/IP host

If the Host wishes to detach from the PVC, then it simply shuts down the TCP connection; it may optionally send a Reset Request first by using the X-R message.

8.1.7 Connection Closure initiated by Gateway

When a connection closure is initiated by the Gateway as a result of an X.25 event, or if the an attempt to set up an X.25 call fails, the Gateway will transmit the X-F message to provide an indication for the reason for the connection failure. If the X.25 call was cleared, the X-F message will contain the $C message element parameter, providing the X.25 clearing cause and diagnostic codes. In addition, the $E element will contain a text explanation.

Here is a list of common explanations and their meanings (the NN is a HEX number):

"$C:00NN,$E:X.25 Call cleared by remote DTE: cause 0x00, diagnostic 0xNN"
Call was cleared with a zero cause code (remote DTE clearing) – contact remote system provider for diagnostic code meanings

"$C:NNNN,$E:X.25 Call cleared by network: cause 0xNN, diagnostic 0xNN"
Call was cleared with a non-zero cause code (network clearing) – contact network provider for cause and diagnostic code meanings

"$E:No buffer space available"
No logical channel was available to make the call

"E:X.25 Link <LINK NAME> down"
The named X.25 link is down

"E:Network dropped connection on reset"
The X.25 link was restarted before the call was accepted

For example:
$C:0000,$E:X.25 Call cleared by remote DTE: cause 0x00, diagnostic 0x00

The complete message (in hex) would be:

```
00 42 12 00   24 43 3a 30   30 30 30 24   45 3a 58 2e   32 35 20 43   61 6c 6c 20
63 6c 65 61   72 65 64 20   62 79 20 72   65 6d 6f 74   65 20 44 54   45 3a 20 63
61 75 73 65   20 30 78 30   30 2c 20 64   69 61 67 20   30 78 30 30   0d 0a
```

8.1.8 Data Transfer

If transmitting the following string “Hello” in a data message, the complete message (in hex) would be:

```
00 05 00 00   48 65 6C 6C   6F
```

8.1.9 Reset Packet Transfer

When an X.25 Reset Packet is received by the Gateway, it will send the X-R message to the TCP host, with the $C parameter containing the cause and diagnostic code.
$C: <cause>< diagnostic >

Similarly, if a TCP host wishes to transmit a Reset packet, it may send the X-R message to the Gateway.

For example: a Reset Packet with cause 0x13 diagnostic 0x32, the message body would be $C:1332. The full message in hex would be:

```
00 07 13 00   24 43 3a 31   33 33 32
```

8.1.10 Sample Applications

Sample applications which illustrate the usage of XDRPD mode are provided in the FarLinX Gateway Resources download.

You can find them in folders:
Sample Applications/XDRPDClient
Sample Applications/XDRPDServer

8.2 DRPD (Dynamic Routing with Packetised Data)

Note that DRPD has now been superseded by XDRPD (see Section 8.1). It is recommended that XDRPD mode be used in preference to DRPD mode.

DRPD uses a message header over the TCP connection both for supplying the X.25 call parameters and for subsequent data transfer. This header contains the length of the message body (thus providing message delimitation), plus an explicit indicator of the message type (so that non-data messages can be differentiated from data messages). The TCP-attached client application can specify all of the connections parameters required to establish an X.25 connection.

DRPD routing for outgoing X.25 calls can also be used in conjunction with other message types – the Call Parameters would be supplied as for the DRPD message type, but then subsequent data transfer will take place according to the particular message type configured.

8.2.1 Message Format

Each message in the TCP data stream has a 4-byte header:

- (2 bytes) length field, in network byte order (i.e. most significant byte first). This contains the length of the message body (i.e. not including the size of the 4-byte header)
- (1 byte) message type field (this is used to determine whether the message is data to/from the X.25 network, or a dynamic routing message);
- (1 byte) spare field (must be zero).

See sections 8.1.1.1 and 8.1.1.2 for variants of this header usage, depending on configuration options chosen.

When the Gateway receives characters from the TCP data stream, it assembles the header and then works out the message body size. Once it has received the number of characters of the message body (as indicated by the length in the header), it determines what to do with the message. For Data messages, the message body is forwarded to the X.25 network. The header of the next message is then assembled.

When it receives a message from the X.25 network, the Gateway constructs a header and then forwards the header plus the message body over the TCP data stream.

The message-type field is used by the Gateway to indicate whether the messages thus constructed are data messages or not. The following message types are defined:

0: data
1: incoming connection indication
2: closure indication
3: connect request (used in dynamic routing)
4: connect response (used in dynamic routing if a message other than a connect request is received)

8.2.2 TCP-to-X.25 Operation

The Gateway uses a message sent by the client to construct the call parameters. The message type field in the message header must be set to 3 (Connect Request). Any other message will cause the Gateway to respond with a Connect Response message.

Once the X.25 call has been made, there is no positive acknowledgement – the TCP-attached client application must assume that the call has been successful, unless it receives a closure indication message. The client can send a data message immediately following the connect request – this will get stored until the connection has been established and will then be sent out immediately.

If the Gateway fails to establish the X.25 connection, a Closure Indication message, providing information on why the X.25 call was failed, is returned to the client.

Connect Request Message

The parameters in the dynamic routing message body are encoded in ASCII, separated by commas. If the first character in the parameter is the '$' character, the parameter is self-identifying as follows:

$D:<destination NUA>
$O:<origination NUA>
$F:<facilities>
$U:<call user data>
$P:<PVC channel number>
$X:<line type: 0=physical, 1=XOT>

All parameters are ASCII strings; the Facilities and Call User Data are specified in hexadecimal. The Gateway converts the strings into the appropriate binary representation.

If self-identifying parameters are not used, it is assumed that the first parameter is the destination NUA; other non-identified parameters are ignored.

For example, to call NUA 12345 on the physical X.25 line, with the Closed User Group facility 99 and a Call User Data string of "HELLO", the message body would contain:

$D:12345, $F:0399,$U:48454C4C4F

The complete message (in hex) would be:

00 1E 10 00 24 44 3a 31 32 33 34 35 2c 24 46 3a
30 33 39 39 2c 24 55 3a 34 38 34 35 34 43 34 43 34 46

When connecting to a PVC channel, use $P:<channel> instead of $D:<NUA>. For example, to call PVC channel 1 on the physical X.25 line, the message body would contain:

$P:1

The complete message (in hex) would be:

00 04 03 00 24 50 3a 31

Closure Indication Message

If the X.25 Call fails, the Gateway will respond with a Closure Indication message, which contains the "$E:" followed by a free format text explanation. Note that software should not be written to depend upon the actual text – the strings may change in future versions of the Gateway.

Here is a list of common explanations and their meanings:

"$E:X.25 Call cleared by remote DTE: cause 0x00, diag 0xNN"
Call was cleared with a zero cause code (remote DTE clearing) – contact remote system provider for diagnostic code meanings

"$E:X.25 Call cleared by network: cause 0xNN, diag 0xNN"
Call was cleared with a non-zero cause code (network clearing) – contact network provider for cause and diagnostic code meanings

"$E:No buffer space available"
No logical channel was available to make the call

"$E:X.25 Link <LINK NAME> down"
The named X.25 link is down

"$E:X.25 Link down"
Call attempted on non-existent adapter number ($A)

"$E:Network dropped connection on reset"
The X.25 link was restarted before the call was accepted

For example:

$E:X.25 Call cleared by remote DTE: cause 0x00, diag 0x80

The complete message (in hex) would be:

00 3b 02 00 24 45 3a 58 2e 32 35 20 43 61 6c 6c 20 63 6c 65 61 72 65 64 20 62 79 20
72 65 6d 6f 74 65 20 44 54 45 3a 20 63 61 75 73 65 20 30 78 30 30 2c 20 64 69 61 67
20 30 78 38 30 0d 0a

8.2.3 X.25-to-TCP Operation

Incoming X.25 Calls are routed in the same way as other types of connections handled by the Gateway – therefore it is necessary to configure a destination IP Address.

When the TCP connection is established, the Gateway will send an Incoming Connection Indication message, containing the X.25 Call parameters: Called Address, Calling Address and Call User Data. This is formatted similarly to the Connect Request message, i.e.

$D:<Called Address>,$O:<Calling Address>,$F:<facilities (in hex)>,$U:<Call User Data (in hex)>

Example:

$D:12345,$O:67890,$U:48454C4C4F

The complete message (in hex) would be:

00 1f 01 00 24 44 3a 31 32 33 34 35 2c 24 4f 3a 36 37 38 39
30 2c 24 55 3a 34 38 34 35 34 43 34 43 34 46

For a connection initiated by the Gateway for a PVC channel, it would be:

$P:<channel>

Example, for PVC on LCN 4001 on the physical X.25 line:

$P:4001

The complete message (in hex) would be:

00 07 01 00 24 50 3a 34 30 30 31

8.2.4 Data Transfer

If transmitting the following string “Hello” in a data message, the complete message (in hex) would be:

00 05 00 00 48 65 6C 6C 6F

8.2.5 Sample Applications

Sample applications which illustrate the usage of DRPD mode are provided in the FarLinX Gateway Resources download.

You can find them in folders:

Sample Applications/DRPDClient
Sample Applications/DRPDServer

8.3 PAD and Call Address Block Routing

PAD Routing and Call Address Block Routing are essentially variants of the same thing, being alternative mechanisms to allow a TCP client to specify some of the X.25 Call Parameters dynamically, rather than having all of

the call parameters configured statically within the Gateway.

These routing mechanisms can be used with a variety of message types. Once the X.25 Call has been established, data transfer takes place according to the configured message type.

8.3.1 Call Address Block Routing – no PID

The TCP client connects to the Gateway and then transmits the X.25 Call Parameters, in the format:
NNNN<CR>
or
NNNN,UUUU<CR>

Where:
- *NNNN* is the destination X.25 DTE address (NUA) in ASCII
- *UUUU* is optional and is up to 16 characters of Call User Data.
- ',' is the ASCII comma character (0x2c), required only if CUD is supplied to separate the CUD from the destination NUA. The ASCII 'D' character (0x44) can be used instead of the comma character.
- <CR> is the ASCII Carriage Return character (0x0d)

The optional Call User Data is inserted into the Call User Data field of the resulting X.25 Call Request packet.

Note that the contents of the Call User Data is encoded in binary and can contain almost any value (including the ASCII comma character 0x2c), but excluding the Carriage Return and Null characters (0x0d and 0x00).

Once the X.25 Call parameters have been supplied, the Gateway transmits an X.25 Call Request.

There is no response from the Gateway, regardless of whether the call is successful.

The client may start transmitting data for forwarding over the X.25 virtual circuit immediately after sending the call parameters (exactly how the data should be formatted and how it is treated is determined by the configured message type).

If the X.25 Call fails, the TCP connection is closed by the Gateway.

If the X.25 Call succeeds, then the TCP client will be able to receive data from the remote X.25 DTE, as well as transmitting further data of its own.

Examples:

Data from Client	Parameters in X.25 Call Request packet
123456,ABCD<CR>	Called NUA=123456, CUD in hex=41424344
12345678<CR>	Called NUA=12345678, no CUD
123456D01<CR>	Called NUA=123456, CUD in hex=3031

Sample Application:
A sample application which illustrates the usage of Call Address Block Routing is provided in the FarLinX Gateway Resources download.
You can find it in the folder Sample Applications/CABRClient.

8.3.2 Call Address Block Routing – X.29 PID

Call Address Block Routing with included X.29 PID is the same as the No PID variant, except that the 4-byte X.29 PID (which in hex is 01000000) is automatically inserted by the Gateway at the start of the Call User Data field within the resulting X.25 Call Request packet. The client is therefore limited to providing up to 12 bytes of CUD.

The X.29 PID is always included, regardless of whether the optional Call User Data is supplied by the client.

Note that Call Address Block Routing with included X.29 PID is also almost exactly the same as PAD routing. The only difference is that that no PAD prompt is transmitted by the Gateway when the client connects.

Examples:

Data from Client	Parameters in X.25 Call Request packet
123456,ABCD<CR>	Called NUA=123456, CUD in hex=0100000041424344

12345678<CR>	Called NUA=12345678, CUD in hex=01000000
123456D1234<CR>	Called NUA=123456, CUD in hex=0100000031323334

Sample Application:

A sample application which illustrates the usage of Call Address Block Routing is provided in the FarLinX Gateway Resources download.
You can find it in the folder:

Sample Applications/CABRClient.

8.3.3 PAD Routing

As observed above, PAD routing is almost exactly the same as Call Address Block Routing with X.29 PID, but there follows a complete description.

When the TCP client establishes a TCP connection with the Gateway, the Gateway immediately transmits the PAD prompt character: "*". (This is the only difference to Call Address Block Routing with X.29 PID.)

The client then transmits the X.25 Call Parameters, in the format: **NNNN<CR>** or **NNNN,UUUU<CR>**

Where:

- NNNN is the destination X.25 DTE address (NUA) in ASCII
- UUUU is optional and is up to 12 characters of Call User Data.
- ',' is the ASCII comma character (0x2c), required only if CUD is supplied to separate the CUD from the destination NUA. The ASCII 'D' character (0x44) can be used instead of the comma character.
- <CR> is the ASCII Carriage Return character (0x0d)

The optional Call User Data is inserted into the Call User Data field of the resulting X.25 Call Request packet, following the 4-byte X.29 PID (01000000 in hex); the X.29 PID is always included, regardless of whether the optional Call User Data is supplied.

Note that the contents of the Call User Data is encoded in binary and can contain almost any value (including the ASCII comma character 0x2c), but excluding the Carriage Return and Null characters (0x0d and 0x00).

Once the X.25 Call parameters have been supplied, the Gateway transmits an X.25 Call Request. There is no response from the Gateway, regardless of whether the call is successful.

The client may start transmitting data for forwarding over the X.25 virtual circuit immediately after sending the call parameters (exactly how the data should be formatted and how it is treated is determined by the configured message type).
If the X.25 Call fails, the TCP connection is closed by the Gateway.
If the X.25 Call succeeds, then the TCP client will be able to receive data from the remote X.25 DTE.

Examples:

Data from Client	Parameters in X.25 Call Request packet
123456,ABCD<CR>	Called NUA=123456, CUD in hex=0100000041424344
12345678<CR>	Called NUA=12345678, CUD in hex=01000000
123456D1234<CR>	Called NUA=123456, CUD in hex=0100000031323334

Sample Application:

A sample application which illustrates the usage of PAD Routing is provided in the product FarLinX Gateway Resources download.
You can find it in the folder:

Sample Applications/PADRoutingClient.

8.4 Cisco Record Boundary Preservation (RBP) protocol

The Record Boundary Preservation protocol implements a 6-byte record header that specifies the amount of data following and indicates whether that data should be considered the final part of a logical record.

Record Header Format	
Byte	**Description**
Byte 0	Protocol identifier. This byte must contain the value 0xD7.
Byte 1	Protocol identifier. This byte must contain the value 0x4A.
Bytes 2 and 3	Payload length, in bytes, not including the header. Byte 2 contains the most significant byte of the length; byte 3 contains the least significant byte.
Byte 4	"More data" flag. This byte must contain one of the following values: • 0x00—Indicates that this record is the final part of the data unit. • 0x01—Indicates that this record is not the final part of the data unit.
Byte 5	Must contain the value 0x00.

Received X.25 data packets are received into logical records as indicated by use of the X.25 M-bit and the contents of the data packets are forwarded to the TCP destination. The boundaries of these records are preserved by the record header. The "more data" flag in the record header will reflect the value of the M-bit in the final X.25 data packet. This process of combining packets results in a series of zero or more records whose "more data" flag is set to the value 1 followed by a record whose "more data" flag is set to 0.

When the Gateway receives the records from the TCP session, it strips the record header and, on the basis of the information in the record header, reassembles the records into X.25 data packets. The data is interpreted as a fixed-length header followed by a variable-length payload whose length is specified in the record header. If the protocol ID or flag field in the header is invalid, the TCP connection will be closed and the X.25 circuit will be cleared or reset. The payload length may be greater than the X.25 packet size and need not be a multiple of the X.25 packet size.

A record that has the "more data" flag set will be logically combined with following records until a record that has the "more data" flag cleared is received. This process results in a sequence of maximum-sized X.25 data packets, each with the M-bit set, followed by an X.25 data packet containing the remaining data that does not have the M-bit set. The Gateway will not wait for an entire record to be received before sending a maximum-size X.25 data packet.

As the records are reassembled into X.25 data packets, the packets are forwarded to the corresponding X.25 circuit.

Data received by a Gateway from a TCP session will be buffered while waiting for the other connection to be established. If the connection attempt fails, the data will be discarded. When a TCP connection is closed, the X.25 circuit will be cleared or reset and any data not yet sent on the X.25 circuit will be discarded.

8.5 X.25 Data Switching over TCP

This mechanism, which is broadly similar in function to XOT, was originally developed for a previous generation of the FarLinX Gateway that did not support XOT operation.

Nevertheless, it is potentially suitable for developers as an alternative to DRPD/XDRPD. Because it was designed for use between 2 Gateways it is, unlike DRPD/XDRPD, symmetrical in operation between the client and the server. (The Gateway can take on either or both of the client/server roles.) The operation between 2 Gateways is described below.

When an Incoming X.25 calls arrives at the Gateway, it will attempt to establish a TCP connection with its peer; should the TCP connection fail, the X.25 call will be cleared.

Once the TCP connection had been established successfully, the client sends a Connection Request (CR) packet to the server, containing the X.25 Call parameters from the incoming X.25 Call – these will be used for the corresponding outgoing X.25 call by the peer Gateway. On completion of the outgoing X.25 call, the server Gateway will transmit a Connection Confirm packet to the client Gateway if the X.25 call was successful, or a Disconnect

Request (DR) packet if X.25 calls was cleared. The client Gateway will then either accept or clear the incoming X.25 call as appropriate.

Once a call has been set up successfully end-to-end, data is exchanged by means of DT packets, which contain an EOT (end-of-transmission) indicator equivalent to the X.25 M-bit, thus allowing record boundaries to be preserved.

If the TCP connection is lost, the corresponding X.25 connection is dropped; if the X.25 connection is lost the corresponding TCP connection is dropped.

8.5.1 Packet Format over TCP

A packet consists of two parts: a packet-header and a packet body (packet payload). The format of the header is constant regardless of the type of packet. The format of the packet-header is as follows:

Packet Header Format	
Byte	**Description**
Byte 0	Protocol identifier. This byte must contain the value 0xFD.
Byte 1	Packet Type (see below) plus EOT bit for DT packets
Bytes 2 and 3	Payload length, in bytes, excluding the header itself. Byte 2 contains the most significant byte of the length; byte 3 contains the least significant byte.

The packet type can take the following values:
0xF1 – DT (Data), final fragment
0xF0 – DT, non-final fragment (more to follow)
0xE0 – CR (Connection Request)
0xD0 – CC (Connection Confirm)
0xC0 – DR (Disconnect Request)

8.5.2 Connection Request (CR) packet

The CR packet type identifier is 0xC0. The packet body contains one or more of the following message elements (see below):
Called Address
Calling Address
Call User Data
TPDU size

Note: the message body must not be empty and must contain at least one of the above message elements.

8.5.3 Connection Confirm (CC) packet

The CC packet type identifier is 0xD0. The packet body always contains the TPDU size message element.

8.5.4 Disconnection Request (DR) packet

The DR packet type identifier is 0xC0. The packet body contains 2 bytes:
X.25 Clearing Cause
X.25 Clearing Diagnostic

8.5.5 Data (DT) packet

The DT packet identifier is either 0xF0 or 0xF1, depending on whether the EOT bit is set.

The EOT (End-of-Transmission) bit is used to indicate the final fragment in the sequence – it is therefore the equivalent to (but the inverse of) the X.25 M-bit.

8.5.6 Message Elements

These self-identifying options are used by the CR and CC TPDUs to contain X.25 Call parameters.

Each option contains an ID byte, a length byte (the length of the element payload) and zero or more payload data bytes.

0xC0 TPDU size
0xC1 Calling Address
0xC2 Called Address
0xC5 Call User Data

The following values are reserved for future use:

0x80 Output X.25 Adapter & Line Number (2 –byte data field)
0x81 X.25 Line Name (as an alternative to adapter/line number)
0xC8 X.25 Facilities

8.5.7 Examples

Here some examples of how messages would be encoded:

CR – Called Address 9876, Calling Address 4321:
FD E0 00 0C C2 04 39 38 37 36 C1 04 34 33 32 31

CR – Called Address 123456, CUD=”ABCDE”:
FD E0 00 0F C2 06 31 32 33 34 35 36 C5 05 41 42 43 44 45

CC:
FD D0 00 03 C0 01 0A

DR – Cause 0x00 Diagnostic 0x88:
FD C0 00 02 00 88

DT – “Hello”
FD F1 00 05 48 65 6C 6C 6F

8.6 PDHFSGW Message Type

This message type is for TCP-to-X.25 Routes to allow a PDH client to connect to an X.25 attached ATM.

When the PDH client connects to the Gateway, it sends a special PDHFSGW format message of type “S” (see structure below). The Gateway makes the call, using the NUA supplied in the message for the Call X.25 Address.

If the X.25 connection is successful, the Gateway sends the client a special PDHFSGW format message of type “R” with the status of “0000” (see structure below).

If the X.25 connection is unsuccessful, the Gateway sends over the TCP connection a special PDHFSGW format message of type “R” with the status of “0001” (see details below). The TCP connection will be terminated by the Gateway.

Where the X.25 connection has been successful, data sent from the TCP side has the data formatted according to the PDHFSGW structure, with the message type ‘N’. The PDHFSGW structure is removed before the data is sent over the X.25 connection. The X.25 More data bit is used where the data extends over more than one X.25 packet until the end of the data in that message has been sent.
Data received from the X.25 side has the PDHFSGW structure applied before the data is sent over the TCP connection, with the length of the data plus header inserted into offset 0.

As is standard with the Gateway if the TCP connection is lost the corresponding X.25 connection is dropped; if the X.25 connection is lost the corresponding TCP connection is dropped.

The PDHFSGW header:

Offset	Length and Format	Description
0	4 Byte ASCII	Length of the subsequent data, these 4 bytes not included. Value Range “0011” – “9999”, 0x30303131 – 0x39393939
4	7 Byte ASCII	Header Type = "PDHFSGW"
11	1 Byte ASCII	Message Type Indicator "S" – Special message "R" – Special message response "N" – NDC message both directions
12	3 Byte ASCII	Implementation Version Number, Value = "100"

The header is 15 bytes in size, but note that the Length does not include the size of the length field itself but does include the rest of the header and therefore takes values between 11 and 9999.

The PDHFSGW complete special message used for connection requests:

Offset	Length and Format	Description
0	4 Byte ASCII	Length of the subsequent data, these 4 bytes not included. Value: “0026”
4	7 Byte ASCII	Header Type = "PDHFSGW"
11	1 Byte ASCII	Message Type Indicator, "S" – Special message
12	3 Byte ASCII	Implementation Version Number, Value = "100"
15	15 Byte ASCII	The X.25 Address of the ATM

The PDHFSGW complete special response message used for responding to connection requests:

Offset	Length and Format	Description
0	4 Byte ASCII	Length of the subsequent data, these 4 bytes not included. Value: “0015”
4	7 Byte ASCII	Header Type = "PDHFSGW"
11	1 Byte ASCII	Message Type Indicator, "R" – Special message response
12	3 Byte ASCII	Implementation Version Number, Value = "100"
15	4 Byte ASCII	Result code of the connection attempt “0000”: success “0001”: connection failed

8.7 Generic Header Conversion

8.7.1 Conv hdr 2-bin

The messages are encapsulated within the TCP payload using a 2-byte header, which is the length of the message (excluding the header itself) encoded in binary, in network byte order (most significant byte first).

Example:
00 05 48 65 6C 6C 6F

8.7.2 Conv hdr 2-ascii

The messages are encapsulated within the TCP payload using a 2-byte header, which is the length of the message (excluding the header itself) encoded in ASCII (most significant digit first).

Note that this message type is cannot be used for messages longer than 99 bytes in length.

Example:
30 35 48 65 6C 6C 6F

8.7.3 Conv hdr 4-bin

The messages are encapsulated within the TCP payload using a 4-byte header, which is the length of the message (excluding the header itself) encoded in binary, in network byte order.

Example:
00 00 00 05 48 65 6C 6C 6F

8.7.4 Conv hdr 4-ascii

The messages are encapsulated within the TCP payload using a 4-byte header, which is the length of the message (excluding the header itself) encoded in ASCII (most significant digit first).

Example:
30 30 30 05 48 65 6C 6C 6F

8.7.5 Custom Message Headers

There are five parameters that govern custom message headers:

TCP Message Header Size
Message Header Length Field Size
Message Header Length Field Offset
Header Length in ASCII
Length Field Includes Header Size

It would therefore take too long to provide an exhaustive list of examples here, but here are a few representative ones:

Header Size=4, Length Field Size=4 Offset=0, length in ASCII=No, Length Includes Header Size=No
00 00 00 05 48 65 6C 6C 6F
(This is the same as Conv hdr 4-bin.)

Header Size=4, Length Field Size=4 Offset=0, length in ASCII=No, Length Includes Header Size=Yes
00 00 00 09 48 65 6C 6C 6F

Header Size=4, Length Field Size=2 Offset=2, length in ASCII=No, Length Includes Header Size=No
00 00 00 05 48 65 6C 6C 6F

Header Size=4, Length Field Size=2 Offset=0, length in ASCII=No, Length Includes Header Size=No
00 05 00 00 48 65 6C 6C 6F

Header Size=8, Length Field Size=4 Offset=2, length in ASCII=Yes, Length Includes Header Size=No
00 00 30 30 30 35 00 00 48 65 6C 6C 6F

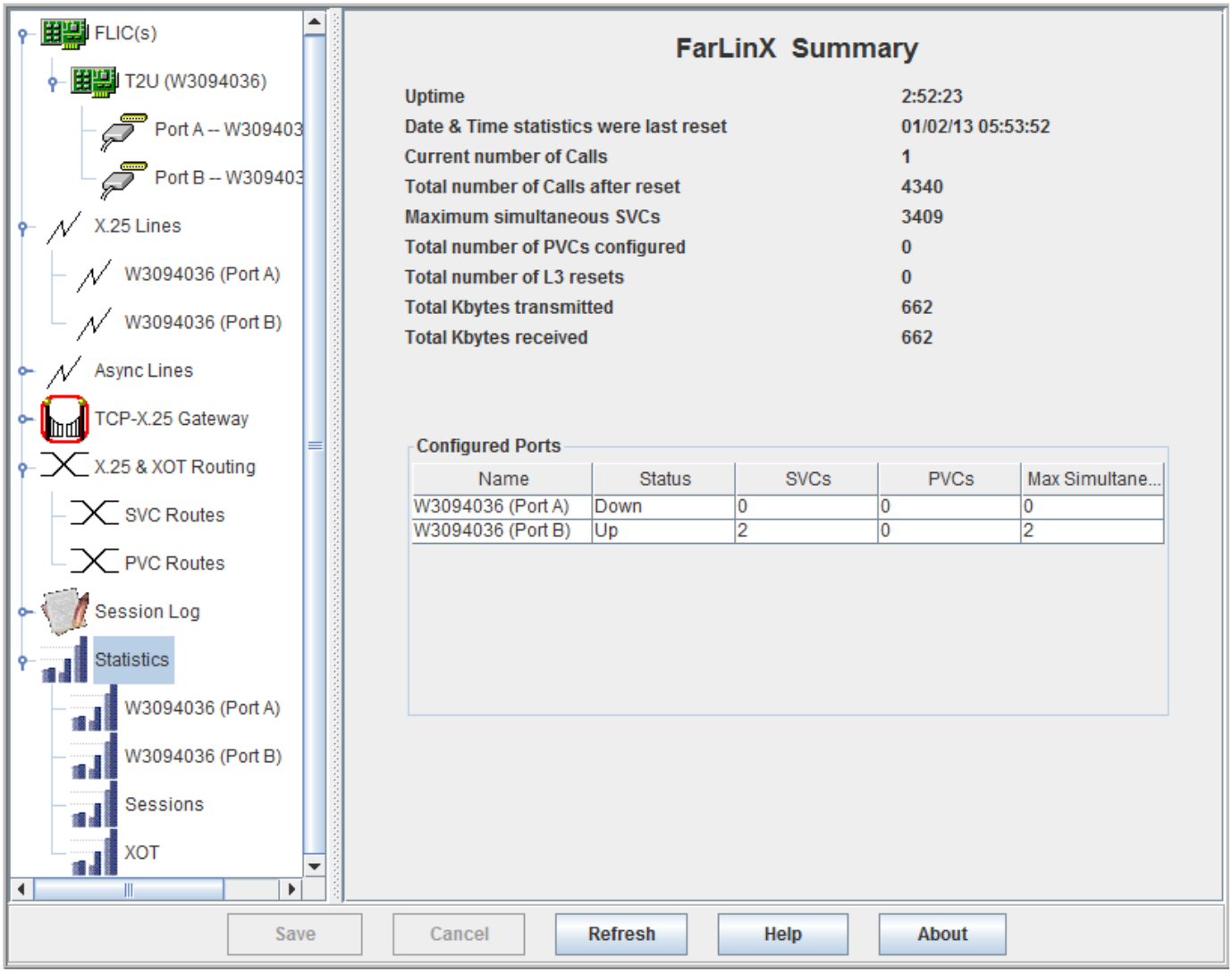
Header Size=8, Length Field Size=4 Offset=2, length in ASCII=Yes, Length Includes Header Size=Yes
00 00 30 30 31 33 00 00 48 65 6C 6C 6F

9 STATISTICS

The Gateway provides detailed statistics of each port through a web browser. The following pages outline the statistics that are available.

9.1 FarLinX Summary Page Statistics

The FarLinX summary page provides the overview information of FarLinX X.25 Gateway; the statistics are displayed as shown below.



The detailed meanings of each of the fields are listed in the following table.

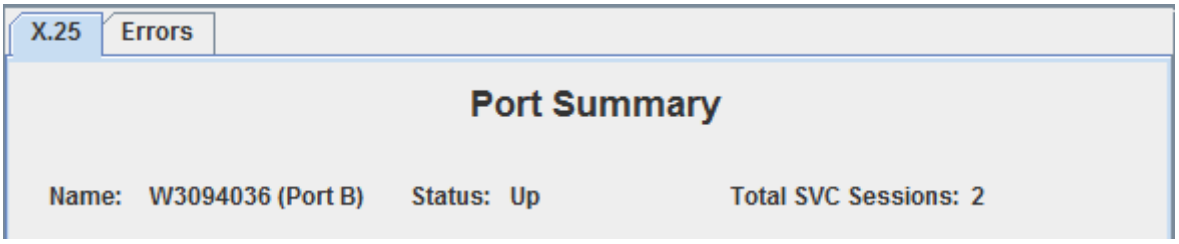
Uptime	Displays the total time that the Gateway has been running since it was last powered up.
Date & Time statistics were last reset	Displays the date and time at which the statistics were last reset.
Current number of calls	This value shows the current number of SVC connections.
Total number of calls after reset	Displays the total number of calls since the statistics were last reset.
Maximum simultaneous SVCs	Displays the peak number of simultaneous SVCs since the statistics were last reset.
Total number of PVCs configured	Displays the number PVC routes configured in PVC routing table.
Total number of L3 resets	This value shows total number of X.25 reset packet events that have occurred.
Total Kbytes transmitted	This displays the total Kbytes of data the Gateway has transmitted over all X.25 lines
Total Kbytes received	This displays the total Kbytes of data the Gateway has received over all X.25 lines

9.1.1 Configured Ports

Name	The name of configured port
Status	The status of configured port: Up means the port is working, Down means the port is not working due to configuration or physical connection problems Unavailable means the port is not working due to the line not being started yet (refer to Section 4.3.4 to start an X.25 line)
SVCs	The number of SVCs sessions in progress
PVCs	The number of configured PVC Routes
Max simultaneous calls	The peak number of simultaneous SVCs that had been in operation since the statistics were last reset

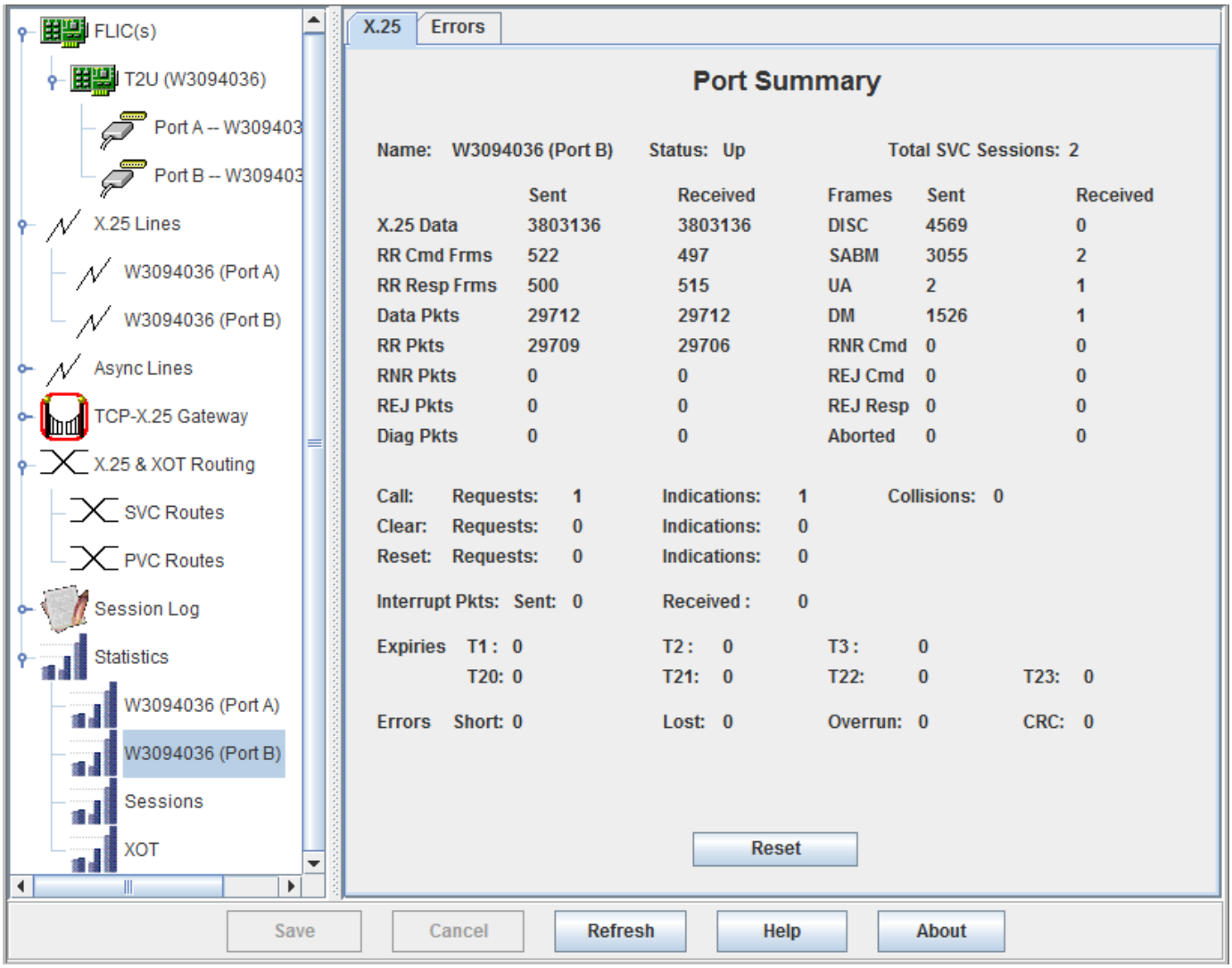
9.2 Port Statistics

Selecting one of the port names in the Statistics branch will display statistics for that particular port. There are two tabs available.



Clicking on each tab will show relevant statistics as described in the following sections.

9.2.1 Port Statistics – X.25

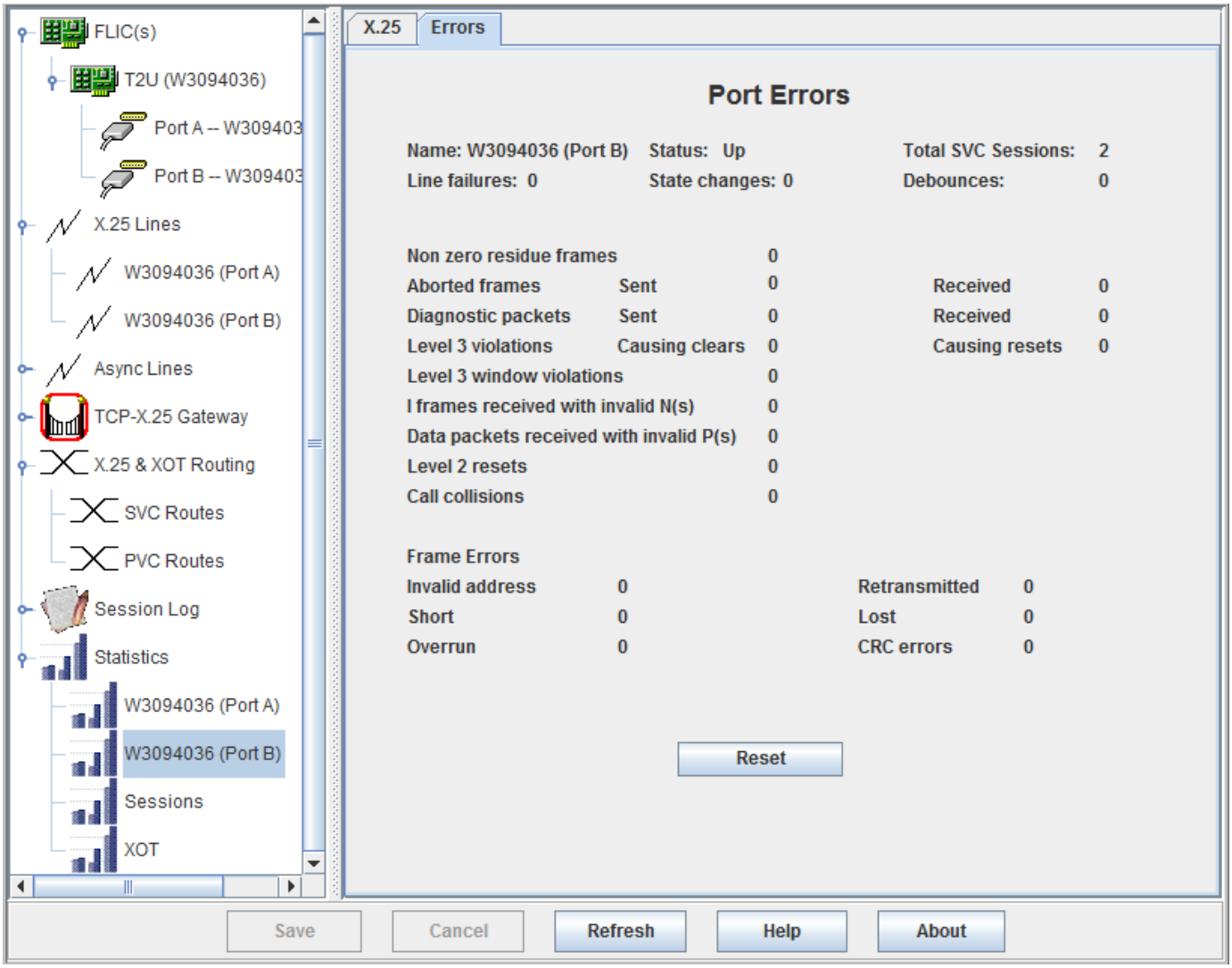


The detailed meanings of the fields are listed in the following table.

Name	The name of the line for which the statistics are currently being displayed
Status	The status of the line. This should normally either be "Up" or "Down"
Total SVC Session	This value shows the total number of SVC sessions established on this line
X.25 Data	This row shows the amount of X.25 data (payload) sent/received on this line
RR Cmd Frms	This rows shows the number of Receive Ready Command frames sent and received on this line. RRs indicate to the other end of a link that the receiver is ready to receive I-frames and can acknowledge previously received I-frames.
RR Resp Frms	This rows shows the number of Receive Ready Response frames) sent and received on this line
Data Ptks	This row shows the number of X.25 data packets sent and received on this line
RR Ptks	This row shows the number of (L3) RR packets sent and received on this line
RNR Ptks	This row shows the number of (L3) RNR packets sent and received on this line. RNR packets are sent to inform the peer to exert flow-control and indicate that no I-frames can currently be processed
REJ Pkts	This row shows the number of (L3) Reject packets sent and received on this line. These are sent to the peer to indicate a sequencing problem with the received data
Diag Ptks	This row shows the number of Diagnostic packets sent and received on this line

DISC Frames	This row shows the number of DISC frames sent and received on this line. DISC frames are used during link establishment
SABM Frames	This row shows the number of Set Asynchronous Balanced Mode (SABM/SABME) frames sent and received on this line. SABM(E) frames are used during link establishment
UA Frames	This row shows the number of Unnumbered Acknowledgement frames sent and received on this line. UA frames are used during link establishment
DM Frames	This row shows the number of Disconnected Mode frames sent and received on this line. DM frames are used during link establishment
RNR Cmd Frames	This row shows the number of (L2) RNR Command frames sent and received on this line
REJ Cmd Frames	This row shows the number of (L2) REJ Command frames sent and received on this line
REJ Resp Frames	This row shows the number of (L2) REJ Response frames sent and received on this line
Aborted Frames	This row shows the number of Aborted frames sent and received on this line
Call	This row shows the number of outgoing X.25 calls (requests), incoming X.25 calls(indications) and X.25 call collisions that have occurred on this line
Clear	This row shows the number of X.25 clears, both outgoing (requests) and incoming (indications), that have occurred on this line
Reset	This row shows the number of X.25 resets, both outgoing (requests) and incoming (indications), that have occurred on this line
Interrupt Pkts	This row shows the number of X.25 Interrupt packets send and received on this line
Expiries	This group of counters indicates the number of times a specific X.25 timer has expired. T1 – The expiry of this timer will initiate the retransmission of a frame T2 – The maximum time before an acknowledgement to a frame must be sent T3 – The period time used for polling an idle link. T20 – The timeout period used to wait for a response after transmitting a Restart Request packet T21 – The timeout period used to wait for a response after transmitting a Call Request packet T22 – The timeout period used to wait for a response after transmitting a Reset Request packet T23 – The timeout period used to wait for a response after transmitting a Clear Request packet
Errors	This row shows counter values for any frame errors that have occurred on this line – including Short frames, Lost frames, Overruns and frames with CRC errors

9.2.2 Port Statistics – Errors



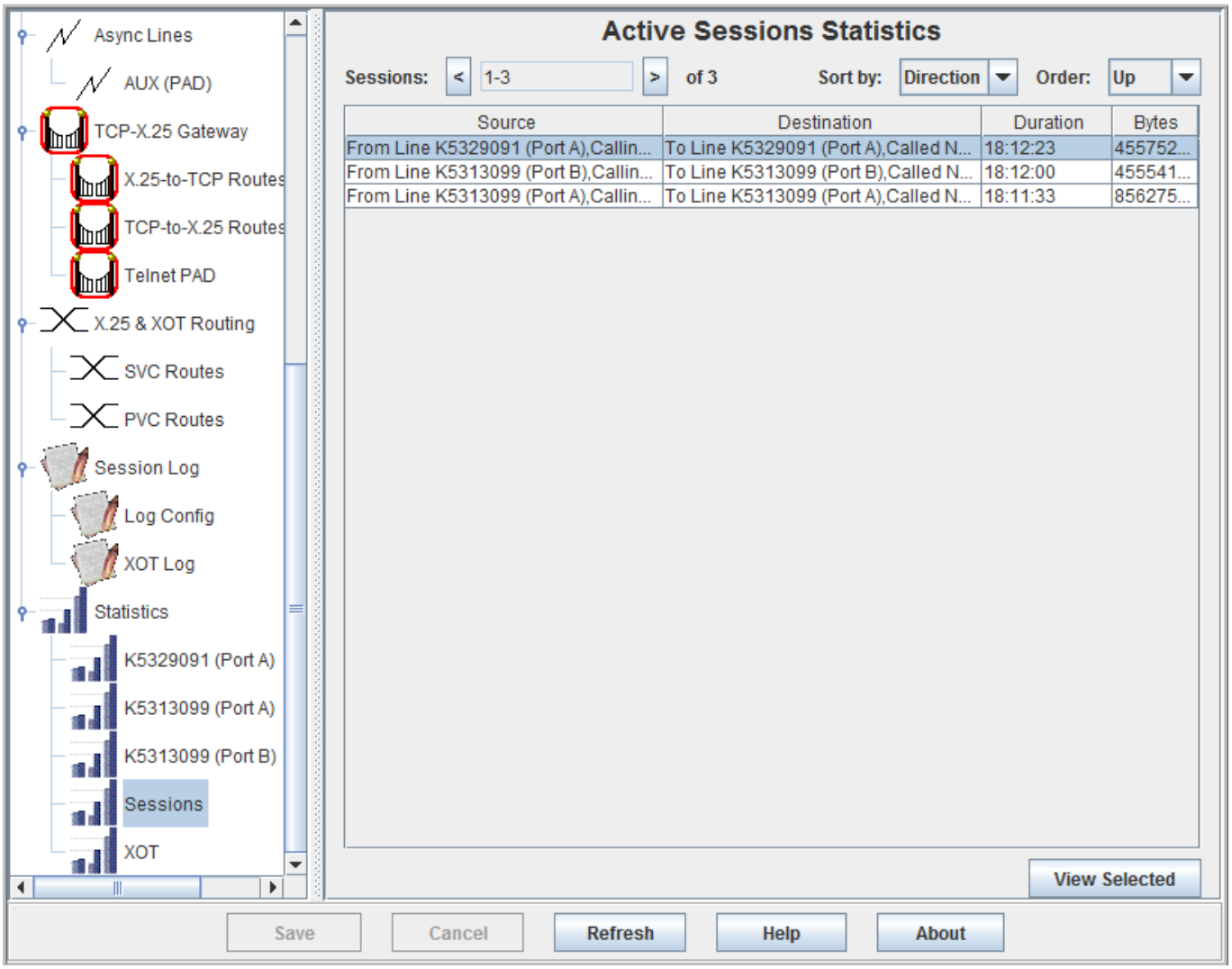
The detailed meanings of the fields are listed in the following table.

Name	The name of the line for which the statistics are currently being displayed
Status	The status of the line. This should normally either be "Up" or "Down"
Total SVC Sessions	This value shows the total number of SVC sessions established on this line
Line Failures	This value shows the number of Line Failures on this line
State changes	This value shows the number of state changes that have occurred on this line
Debounces	This value shows the debounce count for this line
Non Zero Residue Frames	The value shows the number of Non Zero Residue frames that have occurred on this line
Aborted Frames	This row shows the number of Aborted frames sent and received on this line
Diagnostic Packets	This row shows the number of Diagnostic packets sent and received on this line
Level 3 Violations	This row shows the number of L3 violations that have occurred on this line – related to both clears and resets
Level 3 Windows Violations	This value shows the number of L3 window violations that have occurred on this line
I Frames Received With Invalid N(s)	This value shows the number of I Frames received on this line with an invalid N(s) field

Data Packets Received With Invalid P(s)	This value shows the number of data packets received on this line with an invalid P(s) field
Level 2 Resets	This value shows the number of Level 2 resets that have occurred on this line
Call Collisions	This value shows the number of call collisions that have occurred on this line
Frame Errors	This group shows a number of framing errors encountered on this line – including Invalid address, Retransmitted, Short, Lost, Overrun & CRC errors

9.3 Sessions statistics

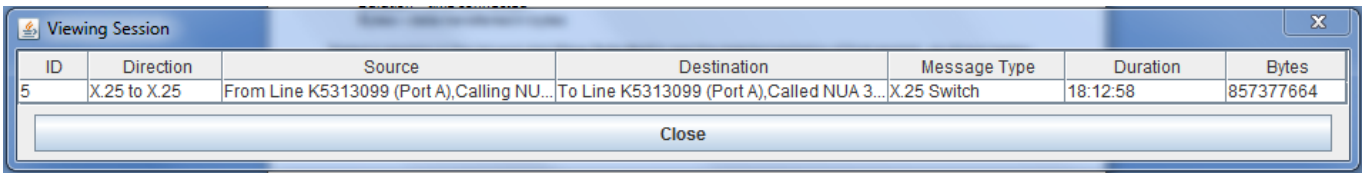
The sessions statistics page shows the status of up to 40 of the Gateway's current X.25 sessions (TCP-to-X.25, X.25-to-TCP and X/25/XOT switch sessions). The display, as shown in the screenshot below, is updated in real-time.



Selecting the column heading will sort the statistics data based on the values in that column:

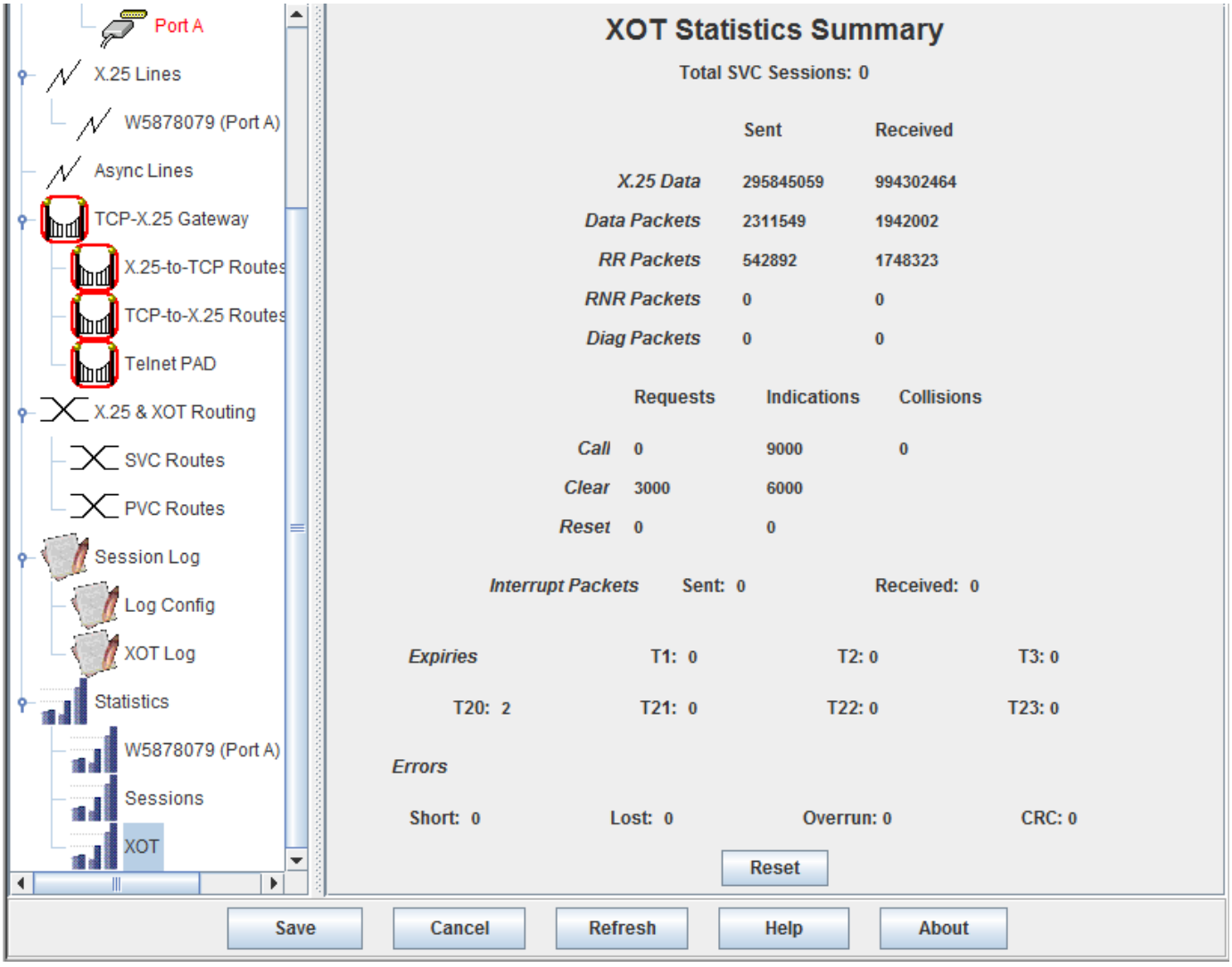
Source – source information
Destination – destination information
Duration – time connected
Bytes – data transferred in bytes

Select a session in the list and click **View Selected** to see the detailed statistics of that session, as shown below:



The session statistics summary information includes:
ID – session id, a reference number for the session, this number is also used in the session log
Direction – **X.25 to TCP**, **TCP-to-X.25** or **X.25-XOT switch**
Source – source details
Destination – destination details
Message type – which message type was used for this session e.g. Character Stream, X.25-XOT Switch
Duration – time connected
Bytes – data transferred in bytes

9.4 XOT statistics



The detailed meanings of the fields are listed in the following table. This page shows the counter values aggregated across all XOT lines.

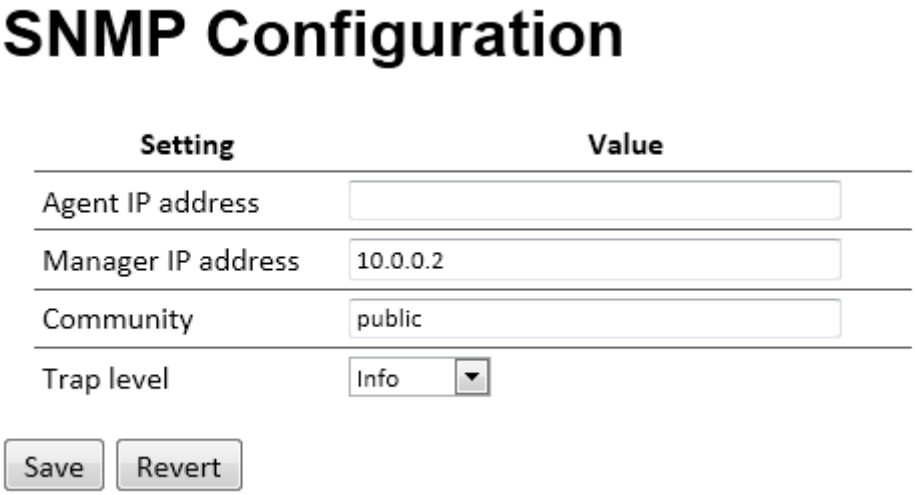
X.25 Data	This row shows the amount of X.25 data (payload) sent/received on this line
Data Packets	This row shows the number of X.25 data packets sent and received on this line
RR Packets	This row shows the number of (L3) RR packets sent and received on this line

RNR Packets	This row shows the number of (L3) RNR packets sent and received on this line. RNR packets are sent to inform the peer to exert flow-control and indicate that no I-frames can currently be processed
Diag Packets	This row shows the number of Diagnostic packets sent and received on this line
Call	This row shows the number of outgoing X.25 calls (requests), incoming X.25 calls(indications) and X.25 call collisions that have occurred on this line
Clear	This row shows the number of X.25 clears, both outgoing (requests) and incoming (indications), that have occurred on this line
Reset	This row shows the number of X.25 resets, both outgoing (requests) and incoming (indications), that have occurred on this line
Interrupt Pkts	This row shows the number of X.25 Interrupt packets send and received on this line
Expiries	This group of counters indicates the number of times a specific X.25 timer has expired. T1 – Not used for XOT T2 – Not used for XOT T3 – Not used for XOT T20 – The timeout period used to wait for a response after transmitting a Restart Request packet T21 – The timeout period used to wait for a response after transmitting a Call Request packet T22 – The timeout period used to wait for a response after transmitting a Reset Request packet T23 – The timeout period used to wait for a response after transmitting a Clear Request packet
Errors	Not used for XOT

10 SNMP TRAPS/ALARMS

The Gateway can generate SNMP v1 traps/alarms which can be sent to a nominated SNMP manager. The IPv4 address of the SNMP manager, to which traps are to be sent, needs to be explicitly configured on the Gateway.

Click SNMP on the navigation bar and configure the values for ‘Agent IP address’ (usually the Gateway’s own IP address), ‘Manager IP address’, ‘Community’ and ‘Trap Level’ then click 'Save' button. The changed configuration will be applied automatically without restarting the Gateway.



Before your SNMP manager can fully decode any traps from the Gateway, you will need to import the FarSite MIB. This MIB file is supplied as \Doc\MIBs\FARSITE-MIB.txt in the FarLinX Gateway Resources download. The OID for FarSite Communications is 1.3.6.1.4.1.18720.1.1

The traps generated by the Gateway include both standard and FarSite-specific traps. Each FarSite-specific trap contains the variables fsTrapSeverity, fsTrapCount and fsTrapDetails.

fsTrapSeverity can be 0, 1,2 or 3, which correspond to the Gateway’s available trap levels: OFF, ERROR, WARNING or INFO.
For filtering unwanted traps, you can change the value of ‘Trap Level’ via the web interface
For example, if ‘Trap Level’ is set to ERROR then WARNING and INFO traps will not be sent.
fsTrapCount indicates how many traps of this type have been generated in the last 5 seconds.
fsTrapDetails includes detailed information regarding the trap.

To avoid unnecessary processing and network traffic, set the trap level to the lowest value that match your SNMP requirements. e.g. select **Off** if /SNMP trap support is not required.

Refer to the FARSITE-MIB for a description of all FarSite-specific trap types.

For further SNMP and SMI general information please reference RFC1157 and RFC1155.

11 LOGS

11.1 Log Configuration – General

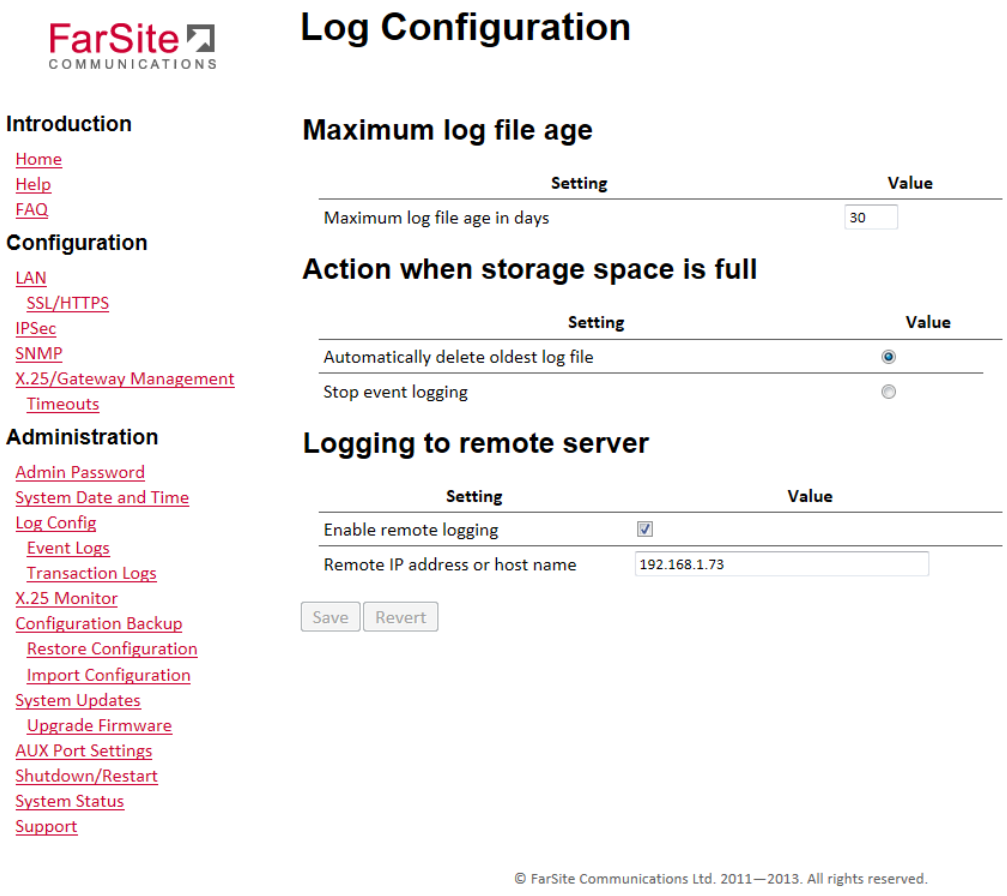
Click 'Log Config' under **Administration** in the navigation area.

Here you can select how long to keep log files before freeing them to make space for new log information. The default is 30 days. Log files older than this will be discarded even if space is still available for logging.

You can also select the action to be taken when the storage space is almost full. The options are to automatically delete older log files or to stop logging further events. Large numbers of local log files increase the Gateway's startup time and can adversely affect performance.

You can also select whether to instead log events to a remote syslog server. This is described further in Section 11.5.

The Gateway supports **rotation** of its log files. This means that, in order to prevent log files becoming too large, the system will periodically save and/or compress existing log files and rename them to enable events to be easily located by date and time. In particular, the Gateway will unconditionally rotate its log files at the end of each day. Sometimes several event log files can be produced and rotated in a single day (if a large number of messages have been recorded). It is recommended that the logs are downloaded regularly and backed up if you need to keep copies of the local logs.



11.2 Event Log

To view locally stored event logs, click 'Event Logs' under **Administration** in the navigation area. All the event log files on the Gateway will be listed. Note that there are two sets of event files: one for Gateway events and one for XOT-specific events. Click 'Save' to save the event file to the Configuring PC. Or click 'View' to view it on the page directly. You can also select the event log files and click 'Delete' to delete them.

The name format of the rotated log files is FSGW<Date>-<Timestamp>.log and XOT<Date>-<Timestamp>.log. Compressed log files are saved as FSGW<Date>-<Timestamp>.log.gz and XOT<Date>-<Timestamp>.log.gz. The active run-time log files are called FSGW.log and XOT.log – these file shouldn't be deleted.

If logging to a remote syslog server has been selected then there may be very little information logged locally.

If the Standby Gateway feature is enabled, the traces of failover operations will be displayed in the SYSTEM.log file.

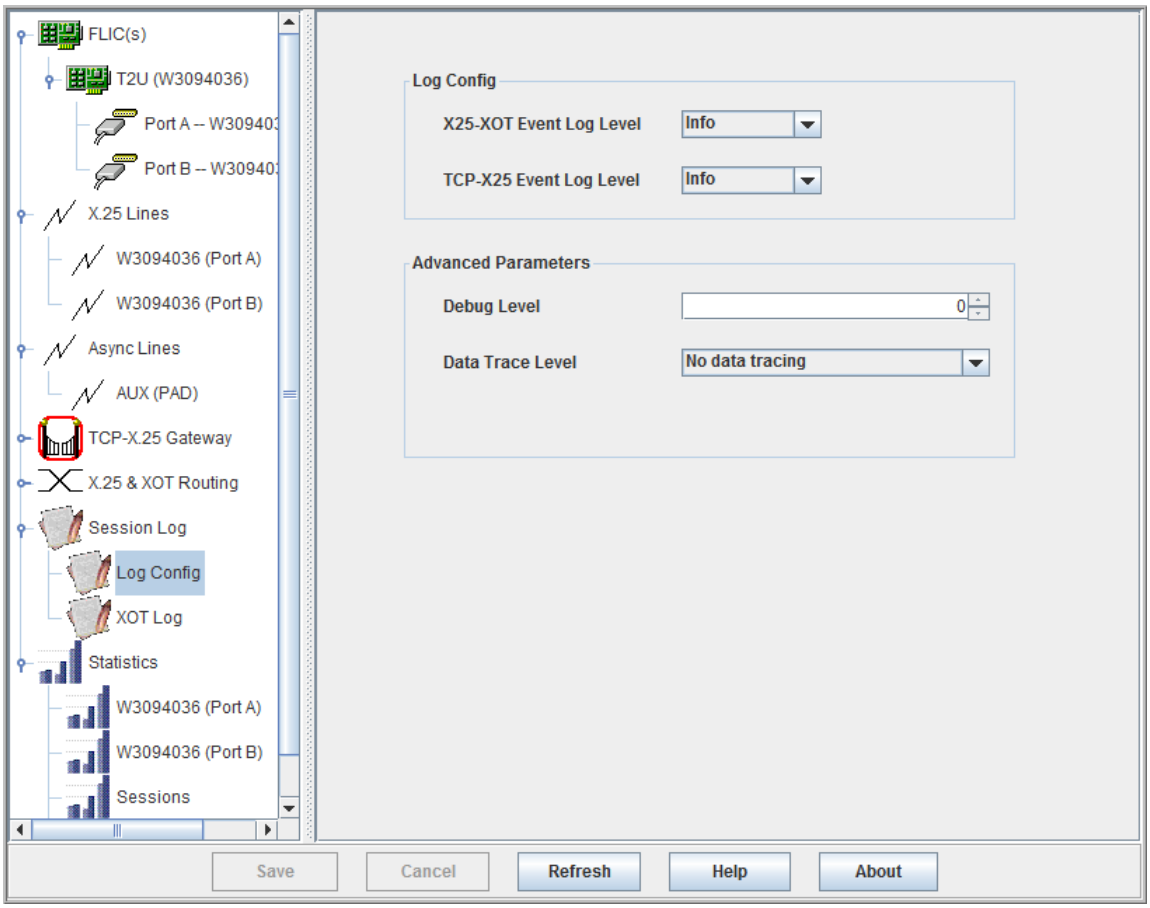
11.3 Transaction Log

As well as the Event Log, the Gateway can also produce logs for individual established Gateway sessions. Click 'Transaction Logs' under **Administration** on the navigation bar. All the transaction log files on the Gateway will be listed. Click 'Save' to save a transaction log to the Configuring PC. Click 'View' to view it on the page directly. You can also select the transaction log files and click 'Delete' to remove them.

If the option to trace data has been selected (see Section 11.4.4 for details) then transaction log data will be generated when a session is created. The data is logged locally or remotely to the configured syslog server. Transaction logs can be large, so if logging locally the Gateway will compress transaction log data internally. If the file is very large the compression operation may take some time and may adversely impact the web configuration and/or data communication response times. It is therefore important to always disable the Transaction log option when the Gateway is in normal production mode (unless you have a specific need to run the Gateway in debug mode for a particular period of time). Once the compression operation is completed the older log files may be viewed or deleted. Log file compression is not performed if logging to a remote syslog server.

11.4 Log Configuration – Detailed

The detailed configuration of the logging operation is made in the Log Config Section of the **X.25/Gateway Management** configuration applet. Click X.25/Gateway Management under **Configuration** in the navigation area. Once the applet has been launched, select the Session Log→Log Config icon. The options available are shown in the following sections.



11.4.1 X25-XOT Event Log Level

This sets the level of event logging related to X25-XOT operation.

The X25-XOT event log level has four settings:

None:	No X.25-XOT events are recorded
Error:	X25-XOT operation errors are recorded
Warning:	Both errors and warnings related to X.25-XOT operation are recorded
Info:	Record all information regarding X.25-XOT operation

Depending on the traffic rate though the Gateway, setting this level to **Info** can adversely affect the Gateway's performance so it is recommended that a value of **Warning** (or less) be configured for normal use.

11.4.2 TCP-X25 Event Log Level

This sets the level of event logging for TCP-X.25 operation.

The TCP-X25 event log level has four settings:

None:	No TCP-X25 events are recorded
Error:	TCP-X.25 operation errors are recorded
Warning:	Both errors and warnings related to TCP-X.25 operation are recorded
Info:	Record all information regarding TCP-X.25 operation

Depending on the traffic rate though the gateway, setting this level to **Info** can adversely affect the Gateway's performance so it is recommended that a value of **Warning** (or less) be configured for normal use.

11.4.3 Debug Level

When the Debug Level is set non-zero, the Gateway will generate additional messages within the Event Log. The higher the level is set, the greater the amount of detail is logged. Normally there will be no need to set this parameter to a non-zero value unless directed to do so by a support engineer. A value greater than 0 may adversely affect Gateway performance and it should always be reset to zero once troubleshooting is complete.

11.4.4 Data Trace Level

The actual data passing through the Gateway can be logged at the X.25 interface and at the TCP/IP interface. This parameter selects the amount of tracing to be enabled. When enabled, the trace gets written to the Transaction Log (see Section 11.3). This can generate a very large volume of output so it should only be used when diagnosing problems. Setting a value other than **No data tracing** can adversely affect the Gateway's performance.

11.5 Logging to a syslog server

If you have an existing Linux system you can configure the syslog server daemon to receive remote data as follows:

First, edit your configuration file /etc/sysconfig/syslog file and set the options to accept remote log input:

vi /etc/sysconfig/syslog *and create or change the options:*
sysLOGD_OPTIONS = "-r –m 0" ***## -r means accept the remote log***

Then restart the syslog service
/etc/rc.d/init.d/syslog restart

Or you can use other third party syslog tools, for example Kiwi Syslog Server for Windows.

The IP address or hostname for the syslog server is configured in the Log Configuration Page shown in section 11.1.

Note that even with remote logging some information may continue to be logged locally to Event Log files. In case of troubleshooting connectivity problems always check local Event Log files as well as remotely logged events.

11.5.1 Structure of Messages Sent to syslog

A remote syslog server treats all logging as if it is a single log of messages, but the Gateway has 3 possible types of log information: Event Log, Transaction Log and Debug Log. The contents of all these logs will become merged on the remote logging system but filtering can be used to separate the different types of message.

The Event log is for notable events, for example Line Up, Line down, session established or cleared.

The Transaction Log optionally provides information about individual sessions (transactions) through the Gateway.

The Debug Log can provide extra information about the handling of messages by the Gateway that may be helpful when connecting to a Host for the first time or when a problem occurs.

To assist with analysis of the received log information, the structure of Event Log and Debug Log messages:

```
DateTime + Address(ip) + Program Name(fsgatewayd) + variable content

For example:
    Jan 21 17:28:05 192.168.1.86 fsgatewayd: id=6: 17:24:47:GW2:Outgoing connection: 6
    successful; from: 192.168.1.99 to: 0234756333
```

For Transaction Log messages, the structure is:

```
DateTime + Address(ip) + Program Name(fsgatewayd) + data-type + variable content
```

The content of the Transaction Log messages can be of 3 types:

- Data-header provides a description of the data that follows
- Data-partial first part of the message contents
- Data-complete complete message contents or final part of a message

Examples of each of these messages are:

```
Jan 21 17:28:06 192.168.1.86 fsgatewayd: Data-header, id=6: len=61, Rx TCP Stream

Jan 21 17:28:06 192.168.1.86 fsgatewayd: Data-partial, id=6:
7474747474747474-7474747474747474-7474747474747474-7474747474373737

Jan 21 17:28:06 192.168.1.86 fsgatewayd: Data-complete, id=6:
3737373737373737-3737373737373737-3737373737373737-3737373737
```

If the message contains the text “id=**nn**”, then **nn** is a session identifier that can be used to filter the session contents.

12 X.25 MONITOR

The Gateway supports monitoring of the traffic on X.25 ports. The monitor support can capture all of the data on a given X.25 line and forward it to a remote Windows system running the FarSync Line Monitor software package.

12.1 Installing and Configuring the FarSync Line Monitor

The traffic monitored by the Gateway is forwarded to a remote Windows machine running the FarSync Line Monitor software package. This Windows system may be running Windows XP, Windows Server 2003/2008, Windows Vista or Windows 7.

To install the FarSync Line Monitor software package, just run the **setup.exe** program on the Windows system. The installation package is located in the /apps/fsmon folder of the FarLinX Gateway Resources download. Once installed, select “FarSync Line Monitor” from the Windows Start menu to run the program.

Once running, from the File menu, select File→Recording Mode… In the subsequent dialog box (see screenshot below), set Monitoring Mode to **Remote;** then set **Local TCP Port Number** so that it matches the **remote port** in the Gateway configuration (default is 5001). Click on the **Save** button to finish the configuration.

Ensure that the Windows system is configured to accept the TCP traffic sent to it from the Gateway. This includes configuring any installed firewall to permit incoming traffic for the TCP port configured.



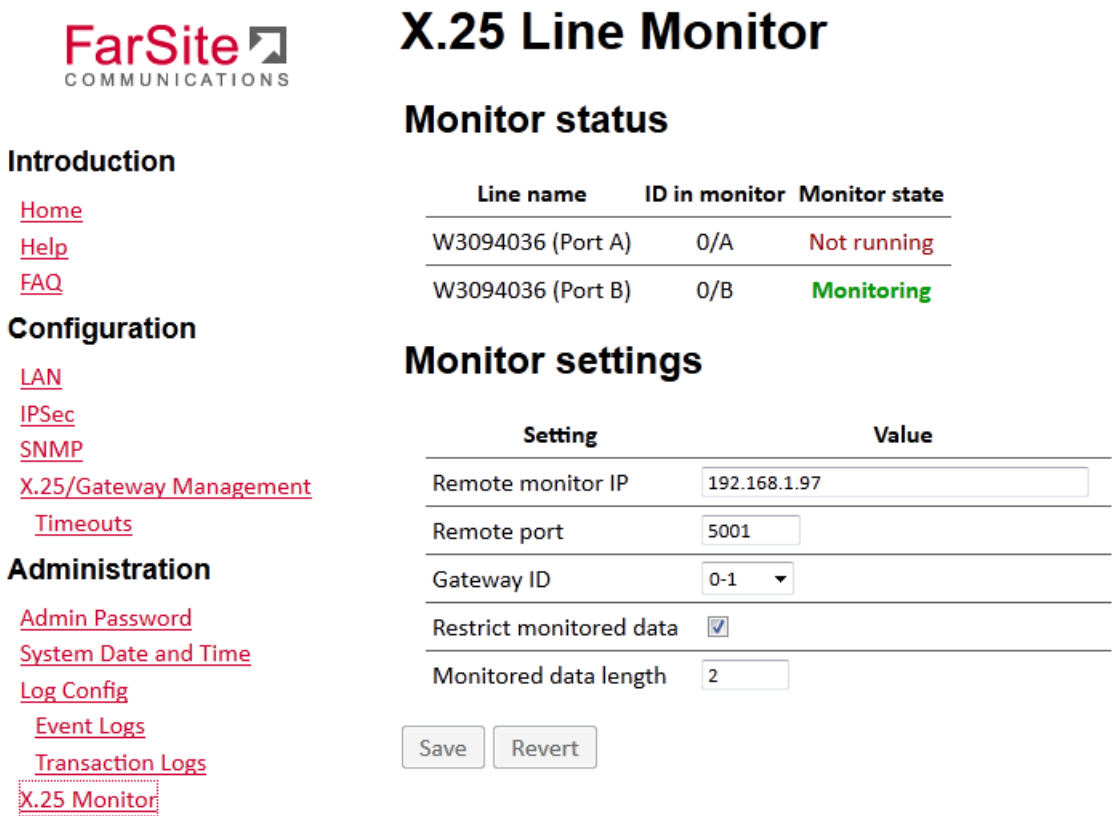
After configuring the TCP port, restart the line monitoring by pressing the “Record/Stop” button twice. You will then be able to view the monitored data from the Gateway, in real-time. See Section 12.4 for details of how to monitor and analyse the data using the popular Wireshark network monitor application.

Refer to the online help included with the FarSync Line Monitor for information on additional features such as the ability to obscure sections of the data before saving it, for additional security with sensitive communications data.

12.2 Configure the Gateway Settings

On the Gateway, you need to specify the IP address of the Windows system that is running the FarSync Line Monitor software. You must also select the TCP port number on which the FarSync Line Monitor is listening (default 5001).

To configure the monitor settings, click on “X.25 Monitor” in the navigation bar, then click the **Modify** button. You may then set the remote IP address, remote TCP port and a Gateway ID. The Gateway ID is used to distinguish the source of monitored traffic when multiple Gateways are sending data to the same Windows system. Ensure that the Gateway ID you choose is unique for the monitoring system. If you are only using one Gateway then this can be left as the default value of 0-1. Click Save to complete the configuration of the Monitor settings.



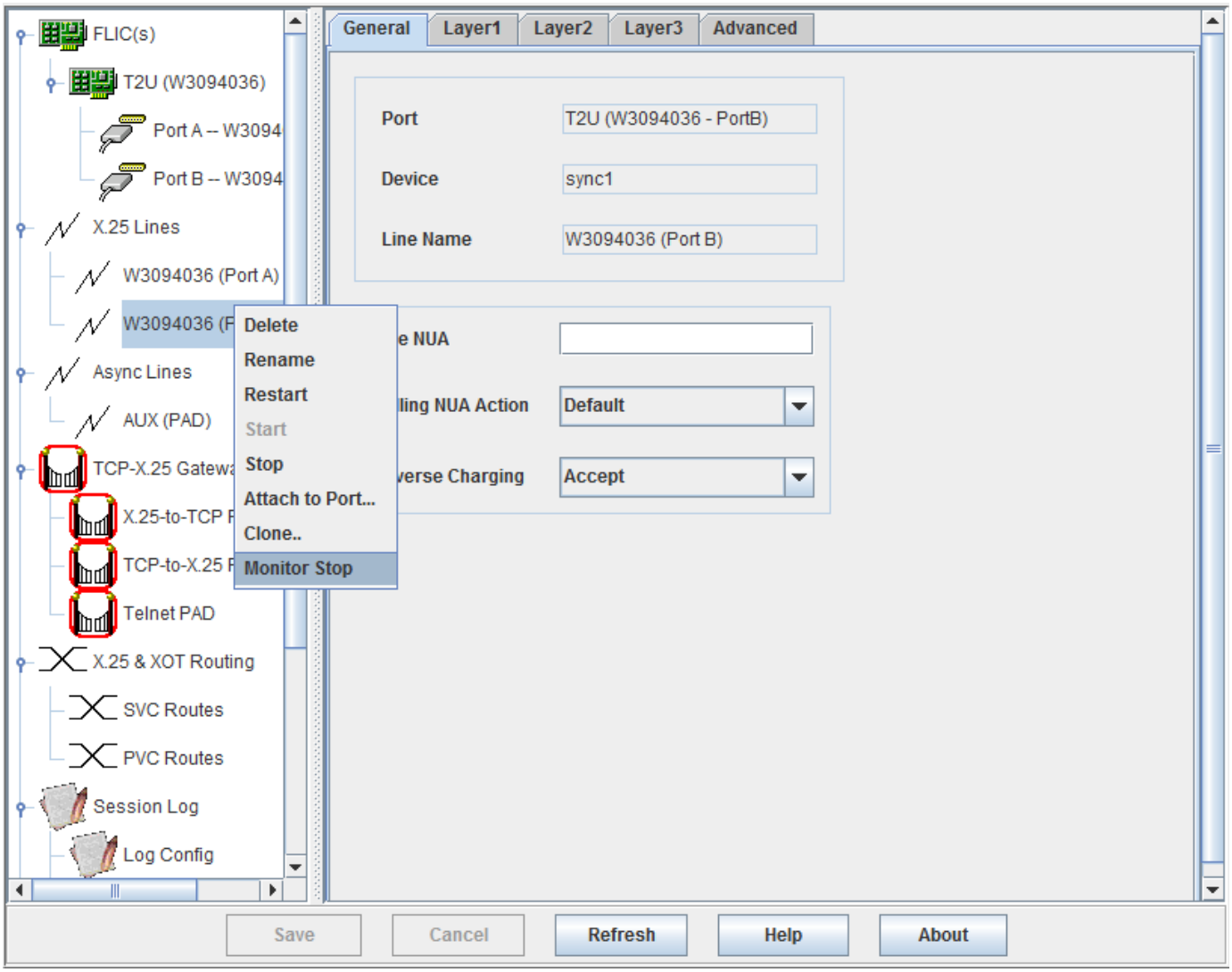
A Gateway ID of **0**-1 means that data from the X.25 line on this FarLinX X.25 Gateway will be displayed in the FarSync Line Monitor application with a line number prefixed by **0.** A value of **2**-3 will cause the prefix to be **2** etc.

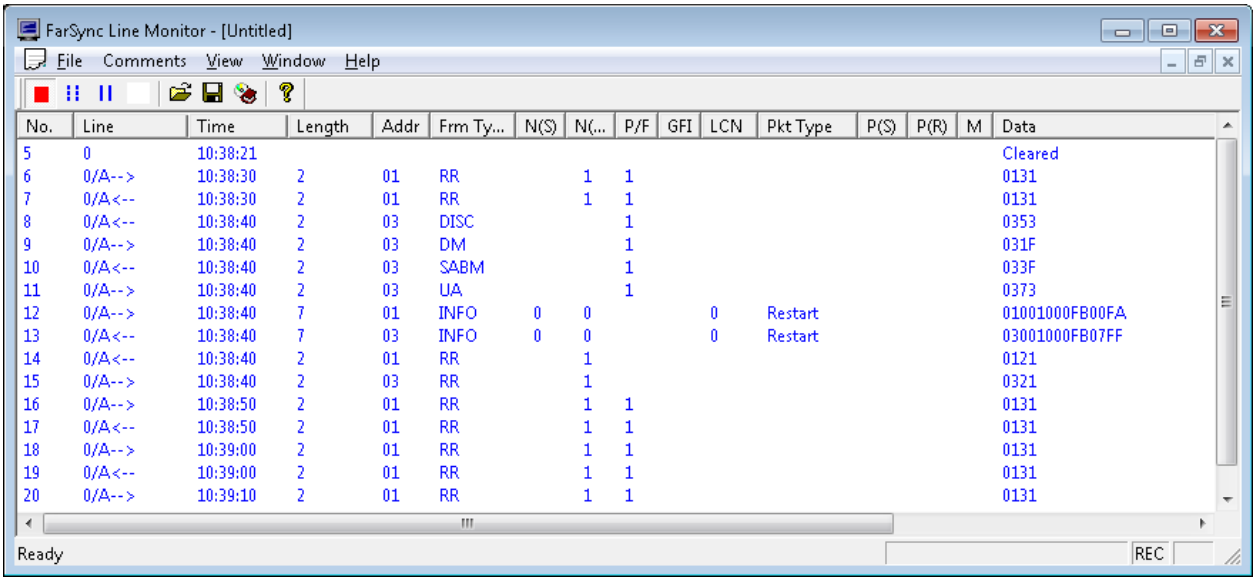
Note: After the monitor is running, if you modify the Gateway configuration again (e.g. IP, Port etc), then the X.25 line monitor operation must be stopped and started again for the new configuration to take effect.

12.3 Start/Stop Monitor

To instruct the Gateway to Start or Stop monitoring, enter the X.25/Gateway Management configuration applet, right-click on the required X.25 line and select **Monitor Start/Monitor Stop** in the popup menu. Once monitoring is started, all traffic on the corresponding line will be captured and forwarded to the remote Windows machine running the FarSync Line Monitor software package.

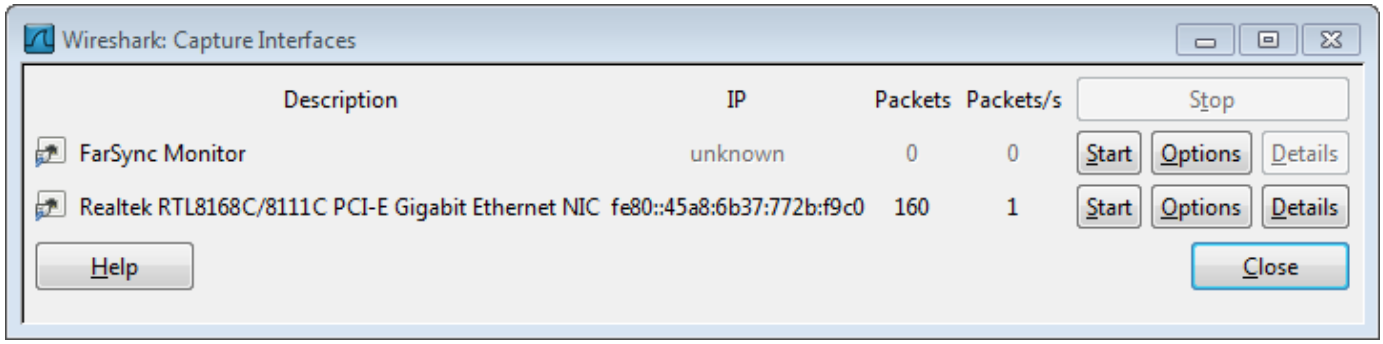
Remember that the Gateway(s) will continue to send monitored data to the Windows PC even if the FarSync Line Monitor application is terminated on the Windows system. This can affect network and Gateway performance so always stop the monitoring of the X.25 line traffic via the Gateway configuration when it is no longer required.





12.4 Monitoring the X.25 lines using Wireshark

The FarSync Line Monitor also allows data to be routed to the Wireshark network monitor application running on the same PC. To set this up, ensure that Wireshark is not running, then in the File→Recording Mode… dialog of the FarSync Line Monitor, select the **Wireshark...** option to setup the FarSync real-time monitoring support. The next time Wireshark is started it will offer an interface called **FarSync Monitor** in the list of available interfaces:



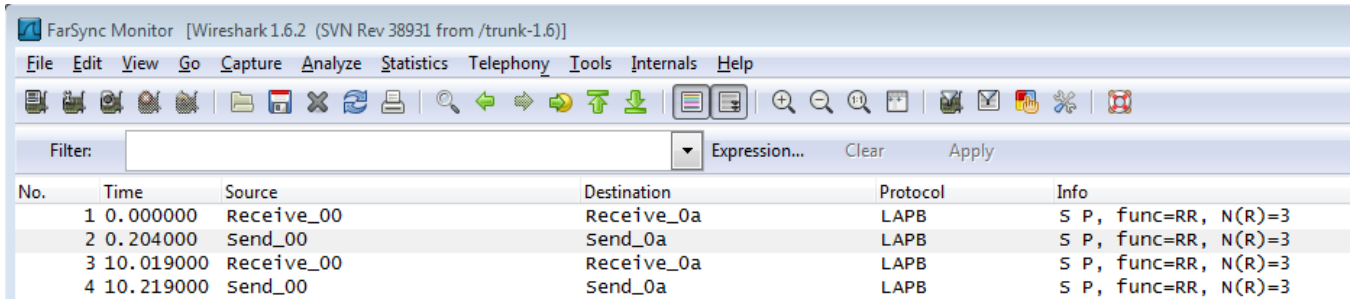
Note that only the 32-bit version of Wireshark is currently supported by the FarSync Line Monitor, although this can be used on a 64-bit version of Windows.

The Wireshark product itself is available for free download – currently from http://www.wireshark.org/download.html.

Note that although both the FarSync Line Monitor and Wireshark can be used for real-time monitoring, only one of these options can be used at any one time.

Data is provided to Wireshark in the form of Ethernet frames. A simulated MAC header is added to the data to indicate the data direction (SEND or RECV) and also the identity of the Gateway supplying the trace data. The Ethernet addresses are used as follows:

20:53:45:4e:44:xx (53:45:4e:44 is “SEND”, displayed as **Send_xx** by Wireshark)
20:52:45:43:56:xx (52:45:43:56 is “RECV” , displayed as **Receive_xx** by Wireshark)



The final 2 digits of the addresses in the Source Ethernet address fields are used to indicate the Gateway ID as shown in the following table:

Ethernet Source address **Send_xx** or **Receive_xx** **(xx** represents Gateway ID**)**		
Value of xx	**ID**	**Meaning**
00	0	Gateway ID 0-1
02	2	Gateway ID 2-3
04	4	Gateway ID 4-5
06	6	Gateway ID 6-7
08	8	Gateway ID 8-9
0a	10	Gateway ID 10-11
0c	12	Gateway ID 12-13
0e	14	Gateway ID 14-15

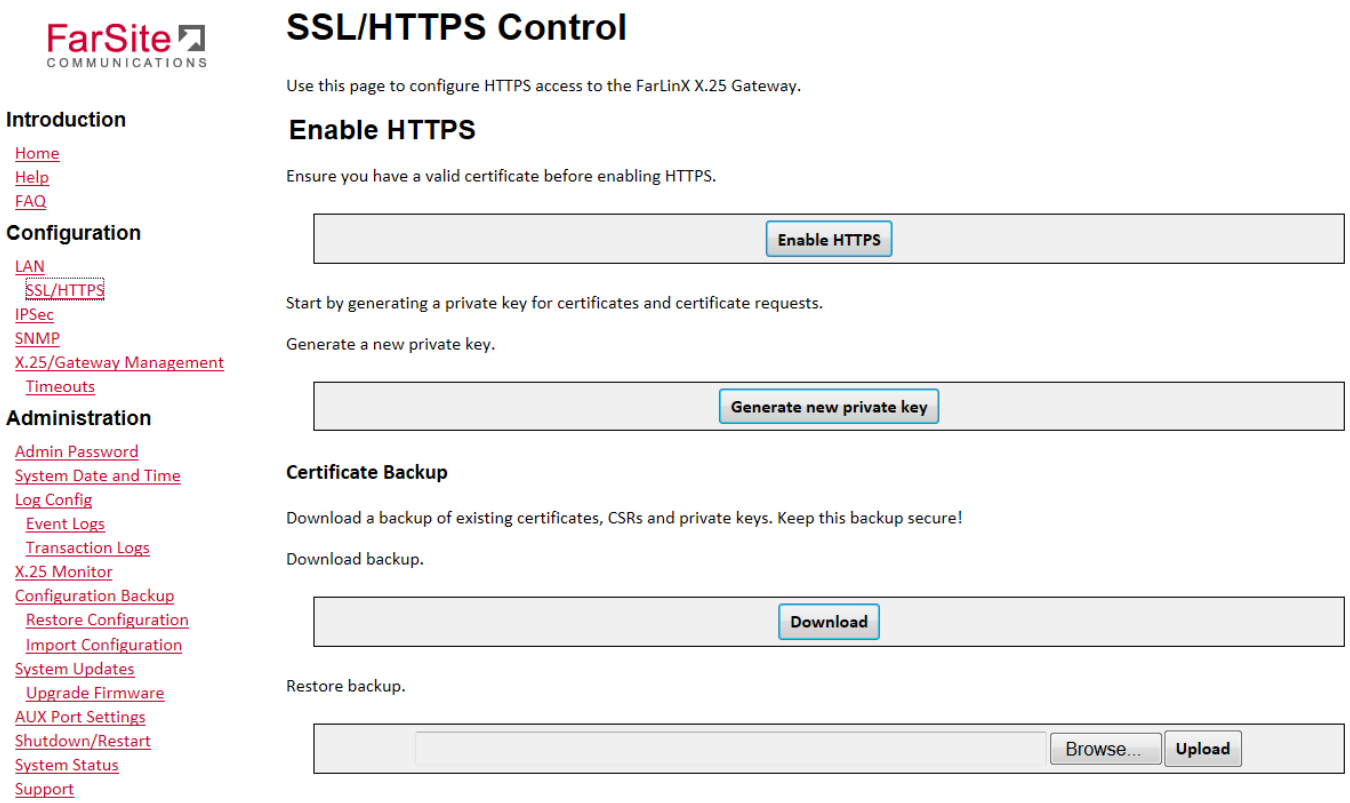
The final 2 digits of the addresses in the Destination Ethernet address fields (**Send_yy**/**Receive_yy**) should always be **0a**.

Refer to Section 12.2 for details of how to configure the Gateway ID of each Gateway you may be monitoring.

For further details regarding the use of Wireshark itself please refer to the help that is installed with the Wireshark product.

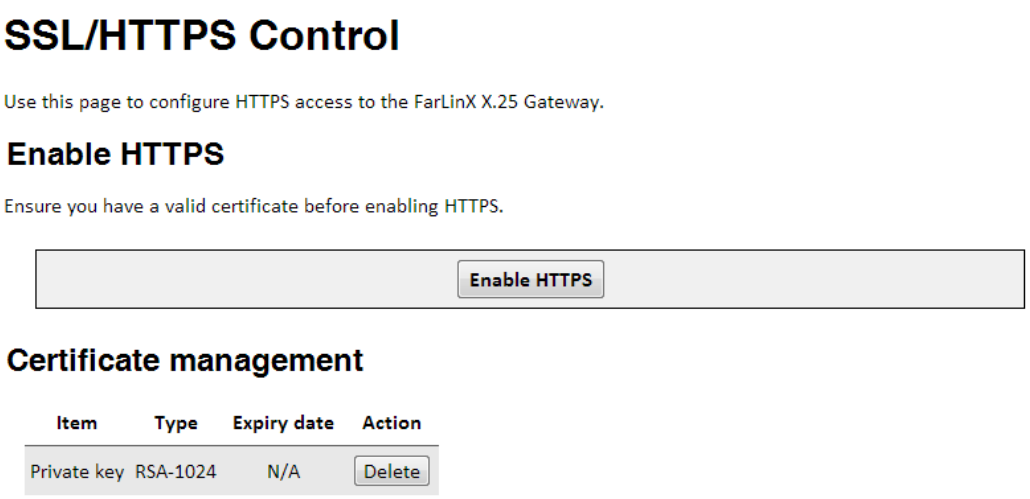
13 HTTPS SUPPORT

The FarLinX X.25 Gateway supports web access using the secure HTTPS protocol. HTTPS (SSL) usage is configured under the menu item LAN → SSL/HTTPS. This page has a different format depending on whether HTTPS has been configured and which options have been chosen. Before HTTPS has been configured and enabled it will look like this:



13.1 Private key

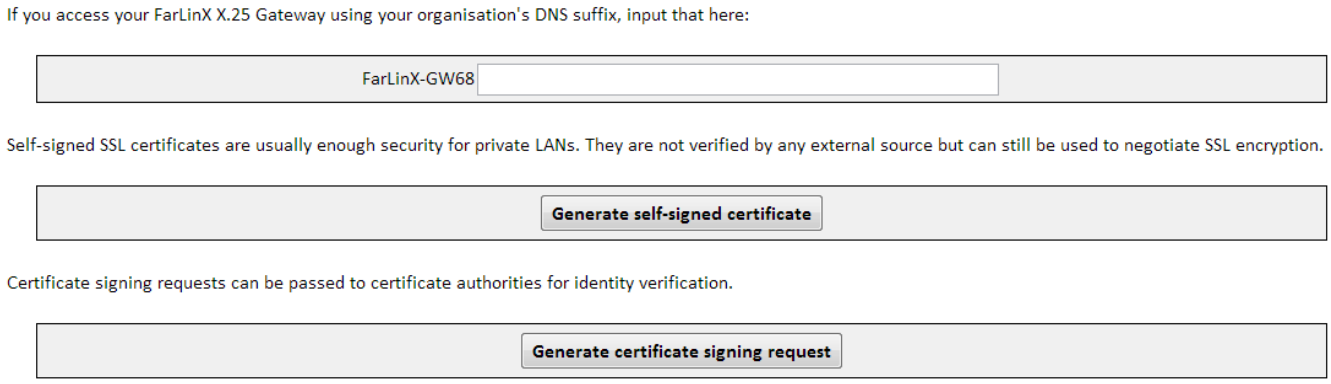
You cannot enable HTTPS until you have established a private key and generated or imported a certificate. Click the **Generate new private key** button to generate a new private key for the Gateway. If a private key already exists, this action will delete that key and invalidate any existing keys, certificates or pending certificate signing requests. This must be done as the first step to enabling HTTPS access. The private key will now be shown (and can be deleted) on this page:



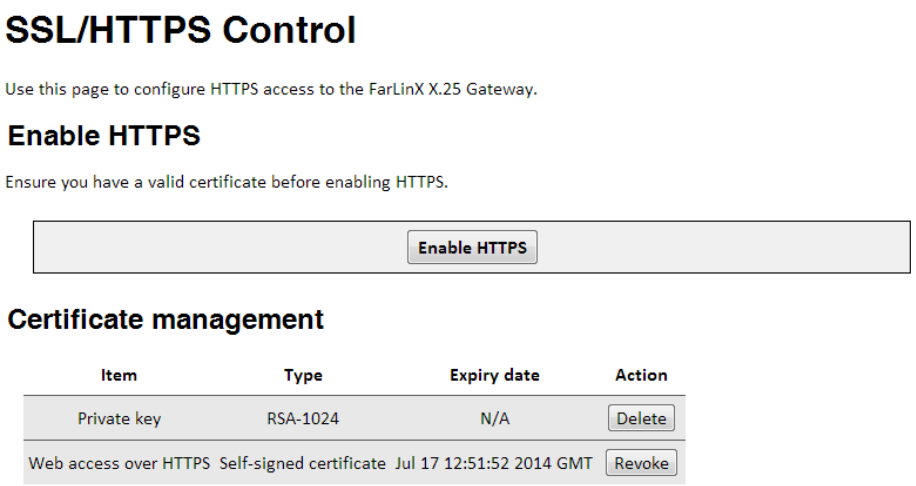
Once a private key exists, there are two possible methods of installing a certificate into the gateway.

13.2 Generate certificates

The first option is to use a self-signed certificate for access. This is usually sufficient security for small, private LANs. Click the **Generate Self-signed certificate** button to create and use a self-signed certificate. This will not be verified by any external source, but can still be used to negotiate SSL encryption. This will usually generate a warning in your web browser, unless the self-signed certificate is installed in your computer's trusted certificate store.



If you access your FarLinX X.25 Gateway using your organisation's DNS suffix, then you need to first input this in the space provided. The Gateway's current host name will be displayed; you can change this on the main LAN configuration page if necessary. Once you generate the self-signed certificate it will be shown on this page.

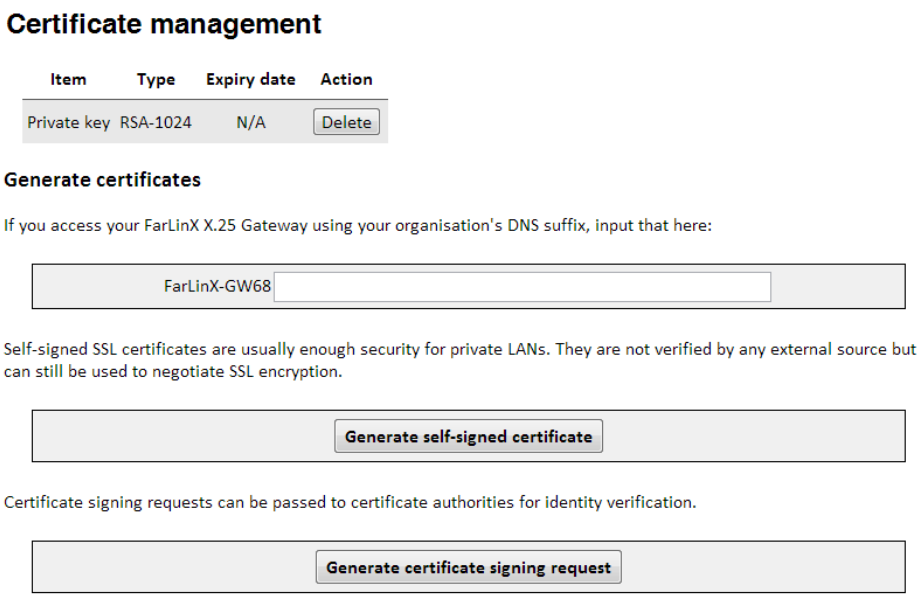


Click **Enable HTTPS** once you have a certificate in place.

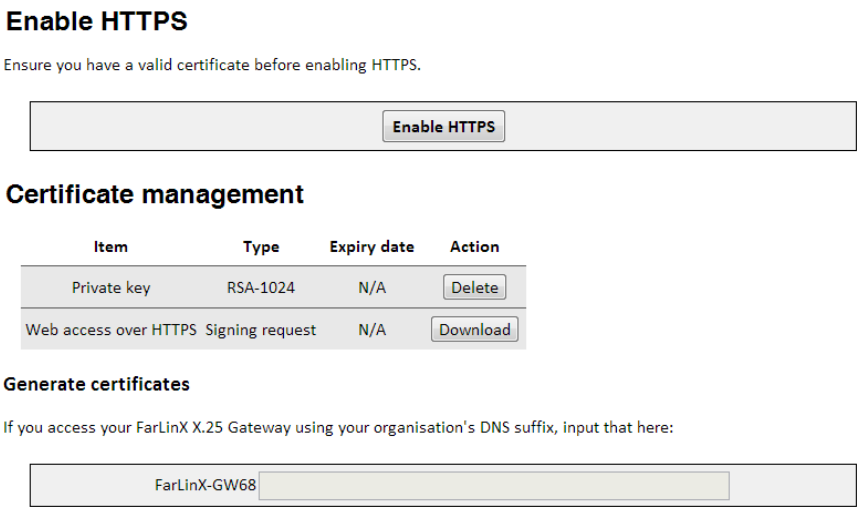
Once enabled, the Gateway should redirect all HTTP requests to the same target page using the more secure HTTPS protocol.

13.3 Upload certificates

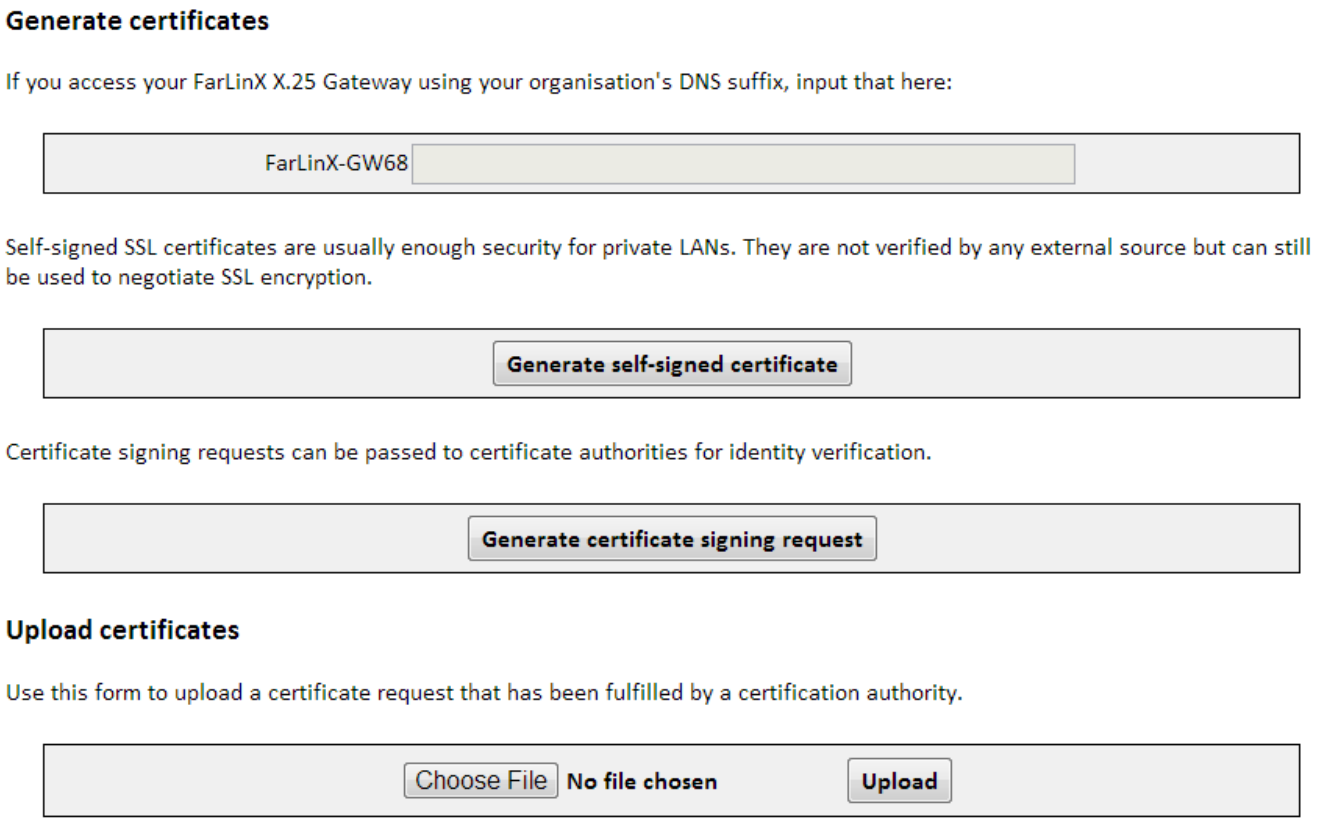
The second and more secure option is to have a certificate generated by a trusted certification authority. This is recommended for large LANs or if you access your Gateway over the internet. Click **Generate certificate signing request** to create a downloadable Certificate Signing Request (CSR). Make sure your organisation's DNS suffix is correctly entered before generating the Certificate Signing Request.



The signing request will now be shown with the private key and can be downloaded using the **Download** button.



Save this file and provide it to a Certification Authority (CA) for signing. After the signing request has been fulfilled by the CA, you can upload the certificate to the Gateway using the form further down this page.



Use **Choose File** and **Upload** to load the new certificate to the Gateway.

In order to validate the certificate when you connect to the GUI of the Gateway, the Certification Authority's root certificate must be installed on the computer you are using for configuration.

Once a signed certificate has been uploaded, click **Enable HTTPS** at the top of the page to start browsing the interface over HTTPS. Once enabled, the Gateway should redirect all HTTP requests to the same target page using the more secure HTTPS protocol.

13.4 Certificate Backup

You can download a backup of existing certificates, CSRs and private keys in one go. This file should be stored securely, as it may contain private keys for trusted certificates.

You can restore backed up certificates by uploading the previously saved **certificatebackup.tar.gz**. Please **Disable HTTPS** before attempting to restore a certificate backup.

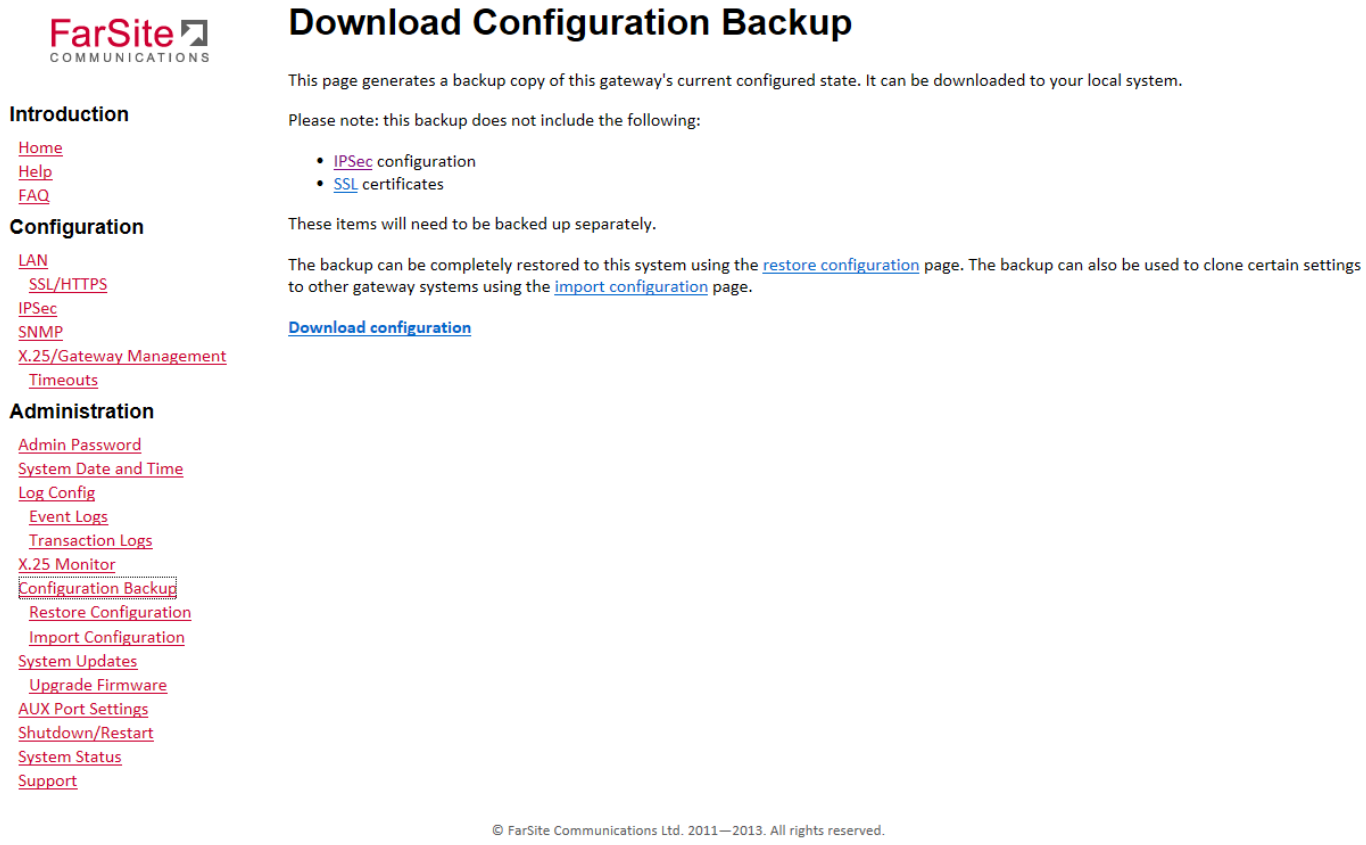
14 MAINTENANCE

14.1 Configuration Backup and Restore

14.1.1 Backup

Click Configuration Backup under **Administration** to obtain a backup file of the Gateway's current configuration (including the LAN, X.25 line, routing rules). The user may then download the file by clicking the Download configuration link.

This package can subsequently be used to Restore Configuration to the same Gateway which generated this package. Alternatively it could be used to Import Configuration to another Gateway in order to clone the configuration.



14.1.2 Restore

Click Restore Configuration under **Administration**, then click 'Browse' button to specify the backup file. After the backup file is specified, click 'Restore' button to restore the configuration.

Note: The complete current configuration will be replaced. All configured X.25 lines will be restarted and all the calls in progress will be terminated.

14.1.3 Import

Import Configuration is used to clone the configuration from another Gateway.

Note: Only the configuration under X.25/Gateway Management will be cloned. The other configuration will remain unchanged e.g. the configuration of LAN, SNMP etc.

14.2 System Update

From time to time FarSite may make available updates to the Gateway software, for example to fix known problems. Please check the Support section of www.farsite.com to see if any updates already exist for your Gateway. To apply an update, click System Update under **Administration**, then click 'Browse' button to select the update file. After the update file is selected, click 'Upload File' button to apply the update to the system.

Use the Support Info page to check that the System Version Number indicates the new patch level. For example it may have changed from 2.6.0 to 2.6.0-01.

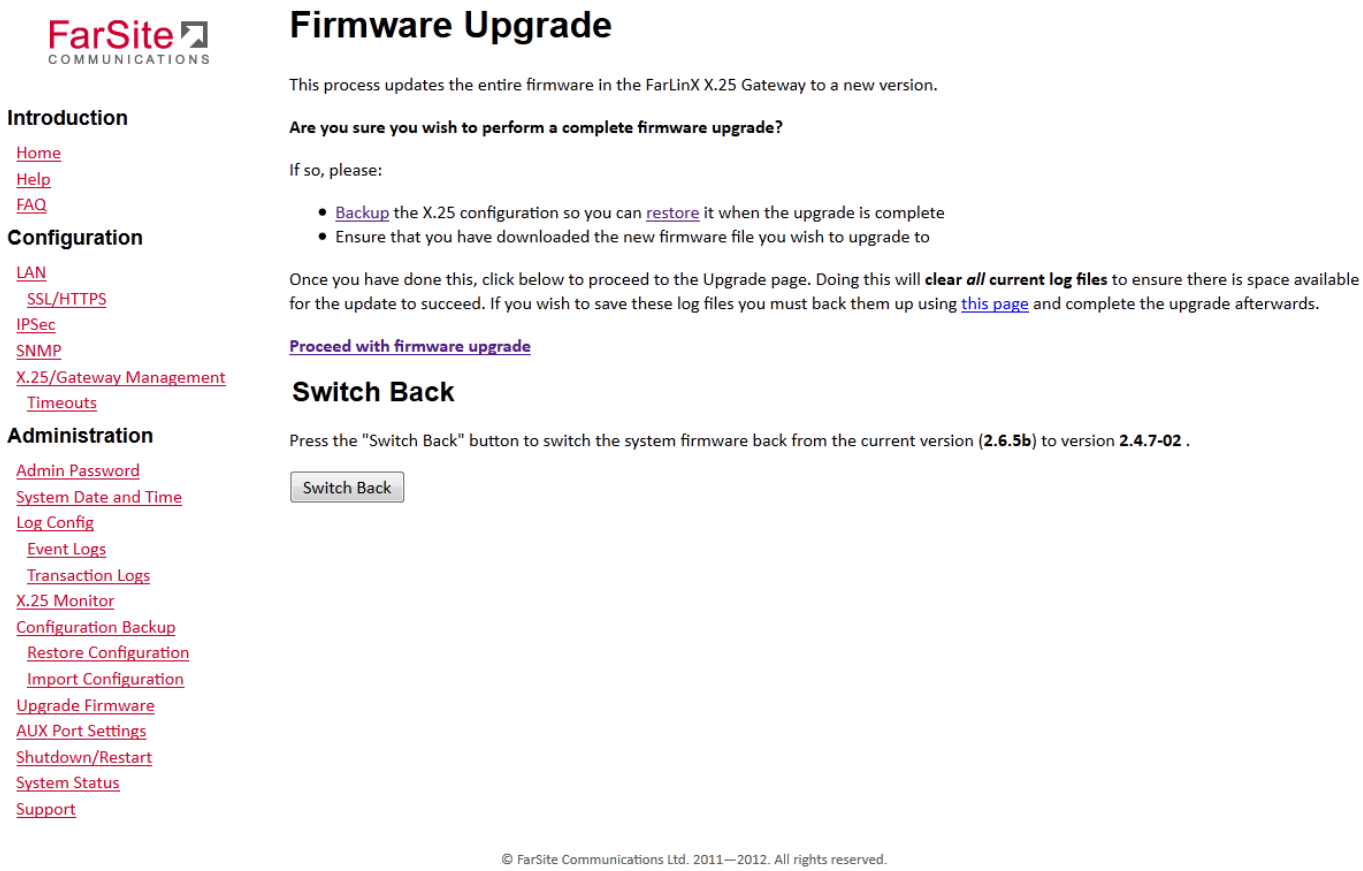
14.3 Firmware Upgrade

The Gateway's firmware can be updated to a new version in the field. Before applying a firmware upgrade you should stop directing all traffic/transactions through the Gateway.

You should backup the Gateway configuration so that you can restore it when the upgrade is complete. An unreliable network should not be used for upgrading systems. It is recommended that upgrades only be performed from a system on the same LAN as the Gateway. **Important note:** when upgrading from an earlier version to v2.6.x, there are a number of configurable options that cannot be restored from an earlier configuration backup. These configured items will need to be manually re-entered after upgrading to v2.6.x. Please see the relevant release note on www.farsite.com for more details.

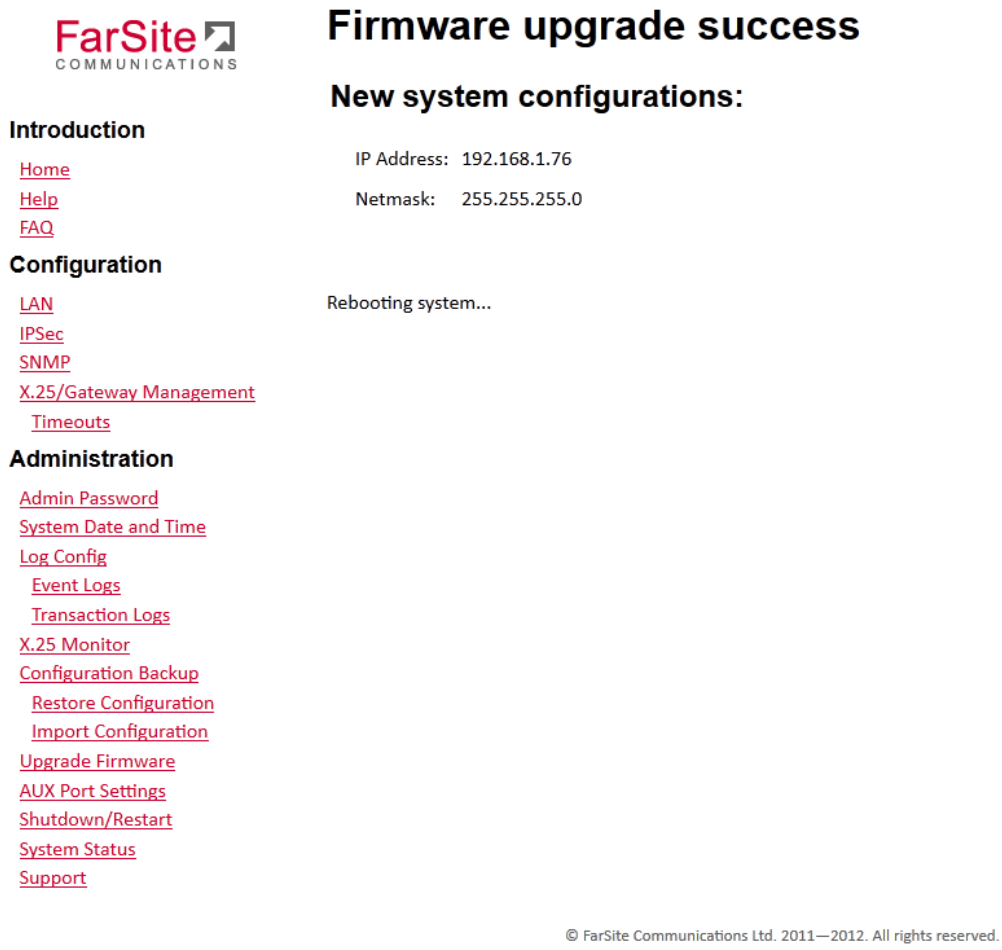
14.3.1 Upgrade

To apply a firmware upgrade, click Upgrade Firmware under **Administration**.



This page reminds you to back up the configuration and any log files you wish to keep. Once done, click "Proceed with Firmware Upgrade". On the next page, click the 'Browse' button to select the upgrade file, then click 'Upload…' button to begin the upgrade procedure. A progress display will be shown to indicate the upgrade process and the three LEDs on the front of the device will flash red while the upgrade is in progress. When the upgrade is finished, a

page will be shown to report that the upgrade is complete and that the Gateway is rebooting into the new software version.



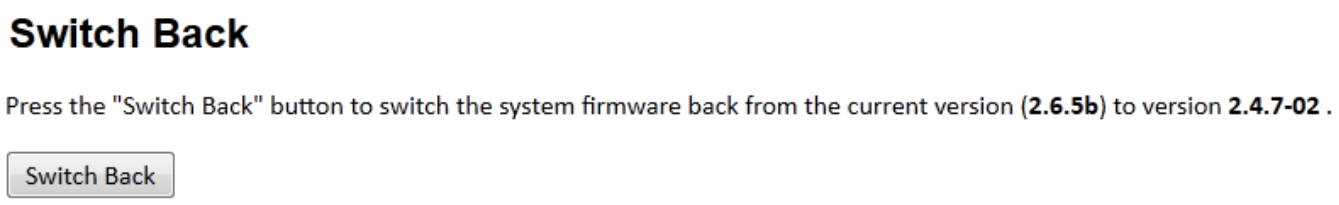
You should ensure that a reliable network connection is available to the Gateway before starting an upgrade. If the upgrade process fails due to a poor network connection then the process will be aborted with the following message:

Error: Upload aborted – please ensure that you are using a high quality network connection

Note: it is possible to use the upgrade procedure to apply the same system version twice. This is not recommended, as it will leave the Gateway unable to switch back to the previous version (see Switch Back, below).

14.3.2 Switch Back

After a successful upgrade, a “Switch Back” button will be shown at the bottom of the “Firmware Upgrade” page. This can be used to switch back to the previous version of the Gateway software, i.e. the version that was running before the upgrade was applied. Before switching firmware versions, you should stop directing all traffic/transactions through the Gateway.

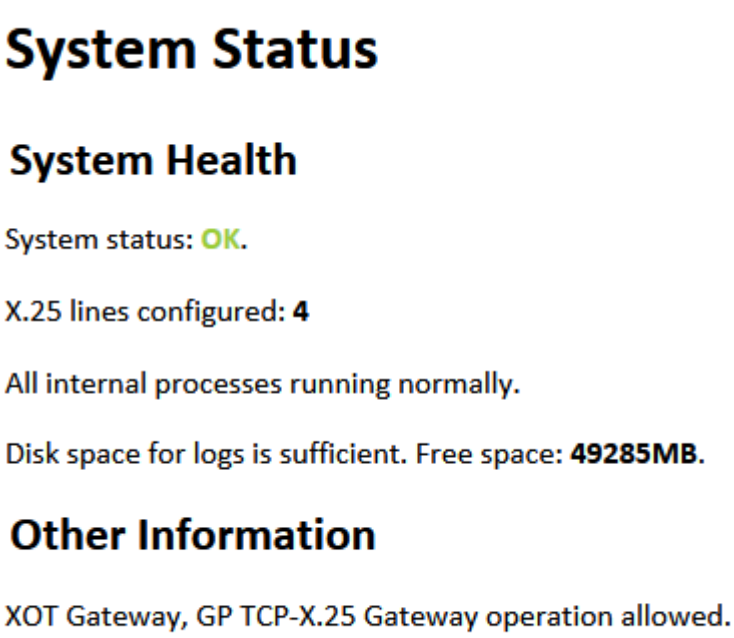


14.4 Restart and Shutdown

The Gateway can be restarted or shutdown from the Administration Menu. Click Shutdown/Restart under **Administration** on navigation bar then click Shutdown or Restart. If you choose Shutdown, the LED will go off once the shutdown is complete. Subsequently manual intervention will be required to start the Gateway by removing/reapplying power.

14.5 System Status

The System has 3 status states: Critical, Warning and OK



Critical status:
Disk free space is less than 3MB – the Gateway will automatically restart upon detecting this
One or more internal processes have failed – the Gateway will automatically restart upon detecting this
If you suspect that the Gateway has restarted due to a critical condition being detected, check the event log for a "fsgatewayd: Gateway Initialised" log entry and check what events led up to the restart

Warning status:
X.25 lines are defined in the Gateway and some are reporting that they are down.

OK status:
All defined X.25 lines are up, log space is over the threshold and all internal processes are running

14.6 Factory Default

The factory default button is located on the back of the Gateway near to the LAN connection. Press this in for a few seconds until the LEDs on the front glow red and the system restarts. **All configuration information will be reset** to factory defaults – this includes:

- Setting the IPv4 address to 10.0.0.1
- Using the default X.25 line configuration
- Deleting the SVC/PVC routing tables
- Removing all log files
- Resetting username/password to admin/farlinx

14.7 Support Information

The Support Information page provides the **Product Information** and **Module Versions** information. In Product Information, hardware features are listed to help identify the model of Gateway. In Module Versions, the version numbers of internal components are listed along with the **System version**. The **Serial Number** of the Gateway can also be found on this page.

FarSite COMMUNICATIONS

Introduction

Home
Help
FAQ
User Manual

Configuration

LAN
SSL/HTTPS
IPSec
SNMP
X.25/Gateway Management
Timeouts

Administration

Admin Password
System Date and Time
Log Config
Event Logs
Transaction Logs
X.25 Monitor
Configuration Backup
Restore Configuration
Import Configuration
System Updates
Upgrade Firmware
AUX Port Settings
Shutdown/Restart
System Status
Support Information

Support Information

Product Information

DRAM size: 1915564 kB

Flash size: 120.0 GB

Mainboard type: LN

Sync ports: 8

Async ports: 1

Module Versions

Module	Version
System version	2.7.6a
Java applet	2.8.7.1
Web interface	2.5.2.30
FarSync driver	2.3.2a-00
FarSync X.25 driver	4.07a-00
X.25 Tap	1.8.0
X.25 daemon	3.09.07.5
Management Server daemon	3.07.12.4
XOT daemon	3.02.06.0
Gateway daemon	2.6.6.15
LCD daemon	2.0.0.22
System manager daemon	2.0.0.20
SNMP daemon	2.0.1.5
Heartbeat	1.0.1.25
PAD TTY	1.0.3.37
PAD redirect daemon	2.0.0.16
IPSec tools	0.7.1
System monitor	3.11

15 LIVE STANDBY GATEWAY

The FarLinX X.25 Gateway supports two gateways running in parallel for resilience. It allows one Gateway to take over from the other in event of failure, without user intervention. Each Gateway has its own unique IP address (physical IP address), but they also share another IP address. The shared IP address is the published address to allow clients to connect – the physical IP addresses are only used to configure the gateways and for the gateways to communicate with each other.

One Gateway is designated as the primary Gateway. The other is designated as the secondary Gateway. The primary Gateway has priority in the event that both gateways are started simultaneously. Otherwise the live Gateway will be the first Gateway to become operational. Usually requests made to the shared IP address will be acted on by the current live Gateway. If the current live Gateway fails, then the shared IP address will be acquired by the standby Gateway and requests will be acted on by the standby Gateway. The following conditions are considered failure situations:

- The Gateway loses connectivity completely e.g. power failure;
- Any of the critical processes in the Gateway fail;
- Log space full;
- Ethernet link disconnected;
- All X.25 lines down (by default).

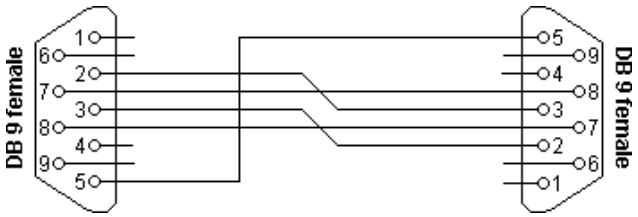
The user is responsible for the independent configuration of both of the gateways when operating in the Live/Standby configuration. Note the X.25 line failure condition will be ignored if no X.25 lines are configured and started.

15.1 Hardware setup

Step 1: Configure each Gateway independently and with unique LAN IP addresses.

Step 2: For additional resilience, connect the 2 Gateways using a serial line crossover cable (contact FarSite Communications or your reseller for suitable cables). This connection can be used in the event that LAN communications between the two Gateways is interrupted. The null modem cable connecting the two gateways is shown in the following figure:

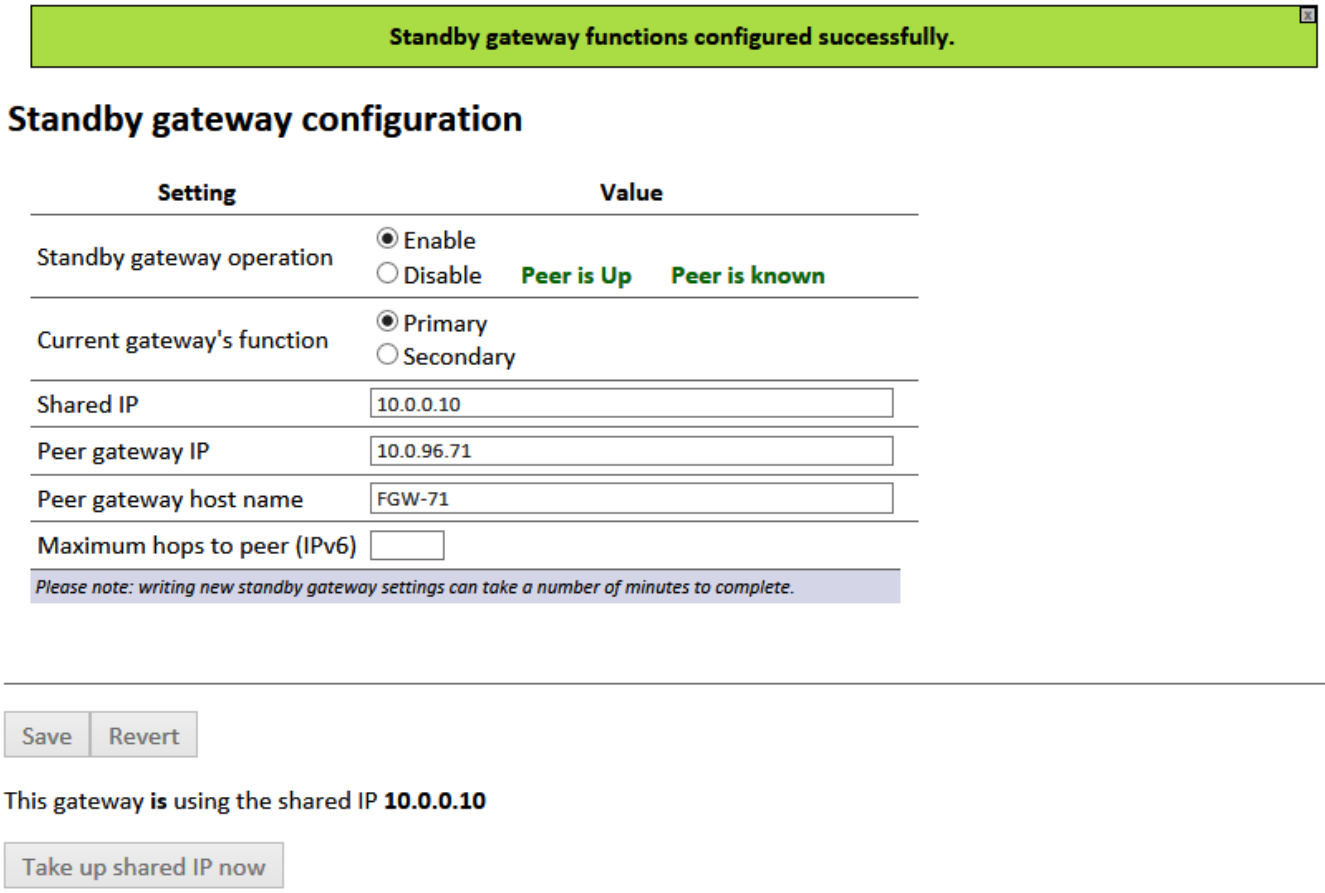
Rx(2) ← Tx(3)
Tx(3) → Rx(2)
GND(5) ←→ GND(5)
RTS(7) → CTS(8)
CTS(8) ← RTS(7)



15.2 Configuration

After finishing the hardware setup and LAN configuration, you can configure the two gateways for the Live/Standby Gateway function.

Click LAN under **Configuration** on the navigation bar. The Live Standby Operation configuration will appear in the main area below LAN configuration:



Enable/Disable: Enable or disable this feature.

Primary/Secondary: Set this Gateway to be the primary or secondary Gateway. This determines which Gateway should acquire the shared IP address when the two Gateways start at the same time. The primary has the higher priority when acquiring the shared IP in this situation, but after the two Gateways have started, the primary and secondary have no functional difference. This means if the secondary Gateway is running OK and holds the shared IP address already, the primary will not try to acquire the shared IP automatically.

If the secondary system comes up before the primary then the secondary will take the shared IP and will keep it. This is to avoid the IP ownership repeatedly switching between the gateways in an error condition case and disrupting possible normal operation. The primary system can be forced to take up ownership of the shared IP address using the “take up shared IP now” button in the configuration.

Note that there must be only one Gateway designated primary and one designated secondary. Setting both gateways to primary or secondary will result in failure to operate correctly.

Peer Status: This is a colour coded field which conveys information about the state of the connection between the Primary and the Secondary.

Peer is Up means that there are successful heartbeat exchanges occurring
Peer is Down means that there is no connection between the Primary and the Secondary
Peer is Known means that the configured peer is responding
Peer is Unknown means that the configured peer is not responding responding, probably due to a configuration error

Shared IP: This is the IP address shared by both gateways. The TCP/IP clients use this IP address to connect to the current live Gateway. The shared IP address must be identical in both gateways.

Peer Gateway IP: The unique physical IP address of the other Gateway in the cluster.

Peer Gateway Host Name: The host name of the other Gateway in the cluster.

Note that this field is required even if the 'Peer Gateway IP' has been configured. It is very important that this name matches the host name of the other Gateway. The host names of the two gateways must be set to different values. This value can be set in the 'HOSTNAME' parameter in the LAN configuration page.

Maximum hops to peer (IPv6): This is for IPv6 only. The IPv6 hop count for outbound heartbeats. This affects how far the multicast packet will propagate. (0-255). Set to 1 for the current subnet. Must be greater than zero.

Then click 'Save' to save your configuration, it will take effect at once.

Example configuration

	"Gateway 1"	"Gateway 2"
LAN Configuration		
Host Name	alpha.gw.mycorp.com	beta.gw.mycorp.com
IP address	10.0.21.8	10.0.21.9
Netmask	255.255.255.0	255.255.255.0
Broadcast	10.0.21.255	10.0.21.255
Gateway	10.0.21.1	10.0.21.1
DNS Server1	10.0.21.1	10.0.21.1
Standby Gateway Configuration		
Live/Standby Operation	Enable	Enable
This Gateway	Primary	Secondary
Shared IP	10.0.21.10	10.0.21.10
Peer Gateway IP	10.0.21.9	10.0.21.8
Peer Gateway Host Name	beta.gw.mycorp.com	alpha.gw.mycorp.com

15.3 Shared IP Transfer

The shared IP transfer event will occur in these situations:

- When only one Gateway starts up, this Gateway will acquire the shared IP whatever the operational status of the Gateway.
- If both gateways start up and one is operational and the other has a failure condition then the non-failing Gateway will acquire the shared IP.

Note that in the following situations, the shared IP transfer event will not occur:

When the standby gateway has failure conditions, the shared IP will stay in the active Gateway which previously acquired it. Note that the failed X.25 line failure condition does not apply to Gateways with no configured and started X.25 lines.

The first Gateway to start up will acquire the shared IP address automatically. To manually acquire the shared IP address for a Gateway, connect to that Gateway (using the unique physical IP address) and click the button 'Take up shared IP now'. The shared IP will then transfer to this Gateway being configured.

Always use the unique physical IP address of the Gateway when configuring the gateways from the web interface.
In the gateway currently using the shared IP address, all TCP or XOT calls made by the Gateway will use the shared IP address as the source.

When using the Live Standby Gateway with IPSec, the shared IP transfer will be delayed by no more than 2 minutes while the IPSec key is refreshed.

A split-brain condition is the result of a Cluster Partition, where each side believes the other is dead and then proceeds to take over resources as though the other side no longer owned any resources.
See http://linux-ha.org/wiki/Split_Brain for more details.

16 X.25 CAUSE AND DIAGNOSTIC CODES

This Section lists the call clearing, call rejecting, restart and reset X.25 Cause and Diagnostic codes that may returned by the X.25 network, a remote X.25 system or by the FarLinX X.25 Gateway itself. An understanding of the meaning of the cause and diagnostic codes will, in many cases, help determine why a reset or restart occurred or why a call was cleared or rejected.

16.1 Gateway Initiated Clearing and Resetting Reasons

This is a list of the X.25 Clearing Cause and Diagnostic codes used by the Gateway itself when clearing incoming X.25 calls.

Note that the cause codes are dependent on whether the X.25 line is configured as a DCE at layer 3 and, if a DTE, whether 1984 functionality is enabled.

16.1.1 DCE X.25 Line Clearing Reasons

Cause	Diag	Explanation
0x00	0x00	No additional information
0x05	0x86	Congestion – Internal Memory Allocation failure
0x0B	0x4c	Access barred – Closed User Group facility absent from Call and required by the route
0x0B	0x94	Access barred – Call cannot be routed back to XOT
0x0B	0x99	Access barred – Closed User Group in Call does not match the configured value for the route
0x0D	0xed	No route configured for Called Address
0x80	0x00	Gateway initiated clear – No licence for this operation

16.1.2 DTE/1984 X.25 Line Clearing Reasons

Cause	Diag	Explanation
0x00	0x00	No additional information
0x80	0x00	Gateway initiated clear – No licence for this operation
0x85	0x86	Congestion – Internal Memory Allocation failure
0x8B	0x4c	Access barred – Closed User Group facility absent from Call and required by the route
0x8B	0x94	Access barred – Call cannot be routed back to XOT
0x8B	0x99	Access barred – Closed User Group in Call does not match the configured value for the route
0x8D	0xed	No route configured for Called Address

16.1.3 DTE/1980 X.25 Line Clearing Reasons

Cause	Diag	Explanation
0x00	0x00	No additional information
0x00	0x86	Congestion – Internal Memory Allocation failure
0x00	0x4c	Access barred – Closed User Group facility absent from Call and required by the route
0x00	0x94	Access barred – Call cannot be routed back to XOT
0x00	0x99	Access barred – Closed User Group in Call does not match the configured value for the route
0x00	0xed	No route configured for Called Address

16.1.4 XOT (X.25 over TCP/IP) Specific Clearing Reasons

Cause	Diag	Explanation
0x05	0x48	Call collision
0x05	0x86	Unable to allocate internal session context
0x05	0xea	No LCN for Incoming SVC
0x05	0xed	Unable to make outgoing TCP connection
0x09	0x70	TCP connection failed

0x09	0x7a	TCP connection closed
0x0d	0xf0	No destination IP address configured

16.1.5 XOT (X.25 over TCP/IP) Specific Resetting Reasons

Cause	Diag	Explanation
0x01	0x70	TCP connection failed
0x01	0x7a	TCP connection closed
0x07	0xed	Unable to make outgoing TCP connection
0x07	0x86	Unable to allocate internal session context
0x11	any	Incompatible destination; the Diagnostic code is the Status code from the XOT PVC Setup response packet – see RFC1613
0x1d	0x70	TCP connection failed
0x1d	0x77	Invalid XOT PVC Setup Response
0x1d	0xf0	No route configured for PVC

16.2 X.25 Diagnostic Code Explanations

16.2.1 Standard X.25 Diagnostic Codes

This is a list of diagnostic codes as specified in the X.25 Standard. Whereas FarSite equipment conforms to this list, there is no guarantee that the meaning of a diagnostic code in a Clear initiated by Third Party equipment will do so,

0x00	No additional information
0x01	Invalid P(S)
0x02	Invalid P(R)
0x10	Packet type invalid channel state
0x11	Packet type invalid for state R1
0x12	Packet type invalid for state R2
0x13	Packet type invalid for state R3
0x14	Packet type invalid for state P1
0x15	Packet type invalid for state P2
0x16	Packet type invalid for state P3
0x17	Packet type invalid for state P4
0x18	Packet type invalid for state P5
0x19	Packet type invalid for state P6
0x1A	Packet type invalid for state P7
0x1B	Packet type invalid for state D1
0x1C	Packet type invalid for state D2
0x1D	Packet type invalid for state D3
0x20	Packet not allowed
0x21	Unidentifiable packet
0x22	Call on one way logical channel
0x24	Packet on unassigned logical channel
0x25	Reject not subscribed to
0x26	Packet too short
0x27	Packet too long
0x28	Invalid GFI
0x29	Restart on non zero logical channel
0x2A	Packet type not compatible with facility
0x2B	Unauthorised interrupt confirmation
0x2C	Unauthorised interrupt
0x2D	Unauthorised reject
0x30	Timer expired
0x31	Timer expired for Call Request
0x32	Timer expired for Clear Request
0x33	Timer expired for Reset Request

0x34 Timer expired for Restart Request

0x40 Call establishment problem
0x41 Facility code not allowed
0x42 Facility parameter not allowed
0x43 Invalid called address
0x44 Invalid calling address or Calling address does not match that specified by Listen
0x45 Invalid facility length
0x47 No logical channel available
0x4C Closed user group is absent

16.2.2 Non-Standard but Common X.25 Diagnostic Codes

These explanations apply to FarSite X.25 equipment and can also apply to some Third Party equipment; codes in Clears initiated by other equipment might have a different meaning entirely.

0x80 No local Name's NUA matches the called NUA
0x82 PVC out of order
0x84 No listens for calls on this Name
0x86 Temporary internal buffer shortage
0x87 Send timeout expired
0x8B Not enough internal resources for call accept
0x8C Session cleared due to name deletion
0x8E Call cancelled
0x8F Internally originated clear due to abnormal condition

0x94 Routing loop detected
0x99 Invalid closed user group
0x9A X.29 invitation to clear
0x9C Internal Accept error
0x9D Call cleared due to X.25 Reset
0x9E Transient level 2 failure causing data loss
0x9F Call cancelled

0xA1 Data packet too long
0xA2 Interrupt packet too long
0xA3 Interrupt packet too short
0xA4 Interrupt Confirmation packet too long
0xA5 Receive Ready packet too long
0xA6 Receive Not Ready packet too long
0xA7 Reset packet too long
0xA8 Reset Confirmation packet too long
0xAA Packet window range exceeded
0xAF Reset packet too short

0xB0 Reject packet too long
0xB2 Internal channel still in use
0xB5 Compression error
0xB6 Compression error
0xB7 Facility byte count too large
0xB9 Facility byte count greater than maximum
0xBB Calls not allowed from unknown callers
0xBD Called address too long
0xBF Calling address too long

0xC1 BCD error in called address
0xC2 BCD error in calling address
0xC3 User data field too long
0xC6 Facility negotiation invalid
0xCA PVC Operational
0xCC No internal session available for call
0xCE Reverse charging call not authorised

0xCF Session limit exceeded

0xD6 Internal X.25 call buffer error
0xD7 Link restarted due to X.25 Handler reset

0xE3 Clear confirmation packet too long
0xEA No free internal channel records
0xED No path to route call

0xFA Restart due to physical line failure
0xFD Restart due to link level FRMR
0xFE Restart due to link closedown
0xFF Restart due to link startup

16.3 X.25 Cause Code Explanations

This Section lists standard cause codes that may be returned by the X.25 network and those codes used by the FarLinX X.25 Gateway. Note that for DTE operation, cause code 00H will be used unless 1984 functionality is selected for the link, in which case the X.25 Handler may output cause codes of 80H or above.

16.3.1 Clearing Causes

0x00 DTE Clear

0x01 Number busy
0x03 Invalid call
0x05 Network congestion
0x09 Out of order
0x0B Access barred
0x0D No route
0x11 Remote procedure error
0x13 Local procedure error
0x15 RPOA (Recognised Private Operating Agency) out of order
0x19 Reverse charging not subscribed to
0x21 DTE incompatible call
0x29 Fast select not subscribed to
0x39 Ship absent

0x80 X.25 Handler internally originated clear
0x81 Number busy
0x83 Invalid call
0x85 Congestion
0x89 Out of order
0x8B Access barred
0x8D No Route
0x93 Local procedure error

16.3.2 Reset Causes

0x00 DTE reset
0x01 Out of order
0x03 Remote procedure error
0x05 Local procedure error
0x07 Network congestion
0x09 Remote DTE operational
0x0F Network operational
0x11 Incompatible destination
0x1D Network out of order
0x81 Out of order

0x83 Remote procedure error
0x85 Local procedure error

16.3.2.1 Reset-Specific Diagnostic Codes

0xA1 PVC attached at peer (cause=0x09)
0xA2 PVC detached at peer (cause=0x01)

16.3.3 Restarting Causes

0x00 DTE restarting
0x01 Local procedure error
0x03 Network congestion
0x07 Network operational
0x7F Registration confirmed
0x81 Local procedure error
0x87 X.25 Handler operational