





Extended Features Reference



Purpose This manual provides a complete description of all the complex, separately licensable features available for the BinTec products. The information included in this manual is compatible with software version 5.1.1.

Liability While every effort has been made to ensure the accuracy of all information in this manual, BinTec Communications AG assumes no liability to any party for any loss or damage caused by errors or omissions or by statements of any kind in this document. BinTec Communications AG is only liable within the scope of its terms of sales and delivery.

The information in this manual is subject to change without notice. Additional information can be retrieved from BinTec's WWW site at www.bintec.de.

Trademark BinTec and the BinTec logo are registered trademarks of BinTec Communications AG.

All other product names and trademarks are the property of their respective companies.

Copyright All rights are reserved. No part of this publication may be reproduced or transmitted in any form or by any means – graphic, electronic, or mechanical – including photocopying, recording in any medium, taping, or storage in information retrieval systems, without the prior written permission of the copyright owner. Also, an adaptation, especially a translation, of the document is inadmissible without the prior consent of BinTec Communications AG.



How to reach BinTec

By ...	At the telephone number or address
Telephone	+49 911 96 73 0
Fax	+49 911 688 07 25
Mail	BinTec Communications AG Südwestpark 94 D-90449 Nürnberg
Internet	www.bintec.de

Copyright © 1999 BinTec Communications AG, all rights reserved.

Version 1.3

Document #71050a

August 1999



1	About this Manual	11
1.1	About your User Documentation	12
1.1.1	How to Get the Latest Software and Documentation	12
1.1.2	Contents	13
1.1.3	Conventions Used in this Guide	13
2	Open Shortest Path First (OSPF)	17
2.1	Setup Tool Menus	18
2.1.1	OSPF	18
2.1.2	Static Settings	19
2.1.3	Interfaces	20
2.1.4	Areas	25
2.1.5	Monitoring and Debugging	26
2.2	Overview of the OSPF Protocol	32
2.2.1	Shortest Path Routing	32
2.2.2	OSPF Routers and Link State Advertisement	34
2.2.3	OSPF Virtual Links	35
2.2.4	Router Types	35
2.2.5	Link State Advertisement Types	36
2.2.6	Router Identification	38
2.2.7	Initialization	38
2.2.8	Neighbor Identification	39
2.2.9	Designated / Backup Designated Router Election	39
2.2.10	Building up the LSD and the SPT	40
2.2.11	Authentication	41
2.2.12	OSPF over Demand Circuits	41
2.3	Example OSPF Installation	43
2.3.1	Configuration Overview	45
2.3.2	Configuration Steps for BRICK-XL2	46
2.3.3	Configuration Steps for BRICK-XM	48
2.3.4	Configuration Steps for BRICK-XS	50



2.3.5	Configuring OSPF Virtual Links	52
2.4	Controlling Link State Database Overflow	54
2.5	Enabling Demand Circuit Support	56
2.6	Import / Export of Routing Information	57
3	RADIUS	61
3.1	Overview	62
3.2	Configuration on BRICK Side	64
3.2.1	Setup Tool	64
3.2.2	MIB	69
3.3	Configuration on the RADIUS Server	74
3.4	Authentication	76
3.4.1	List of Standard Attributes Supported	76
3.4.2	List of BinTec Attributes (Extensions)	81
3.4.3	Sample Modification for Merit RADIUS Servers	83
3.5	Accounting	84
3.5.1	List of Sent Attributes Supported	84
3.6	RADIUS for Dial-Out	87
3.6.1	Configuration on the BRICK	88
3.6.2	Configuration on the RADIUS Server	88
3.7	Examples	96
3.7.1	Typical Dial-In (Without BinTec Attributes)	96
3.7.2	Standard Dial-In with CLID	96
3.7.3	Callback PPP Negotiated	97
3.7.4	Callback (Windows Client)	97
3.7.5	Callback (CLID)	98
3.7.6	Working with one or more RADIUS Servers	99
3.7.7	Dial-Out	100

4	Token Authentication Firewall (TAF)	101
4.1	Overview	102
4.1.1	Requirements	103
4.1.2	Authentication	104
4.1.3	Encryption	104
4.2	Configuration of TAF	105
4.2.1	Configuring the ACE/Server	105
4.2.2	Configuring the BRICK (ACE/Agent)	107
4.2.3	Configuring the TAF Client PC	118
5	Virtual Private Networking (VPN)	123
5.1	Setup Tool Menus	124
5.2	Overview of Virtual Private Networking	133
5.2.1	Overview	133
5.2.2	Tunnelling and PPTP	134
5.2.3	Authentication – Encryption – Compression	135
5.3	VPN and NAT	137
5.3.1	Constellation	138
5.3.2	NAT on VPN Configuration	139
5.4	Virtual Private Networking Examples	144
5.4.1	Example Client-to-LAN Configuration	144
5.4.2	Example LAN-to-LAN Configuration	152
6	X.25	159
6.1	An Introduction to X.25	160
6.1.1	Call Setup	161
6.1.2	Data Links and Virtual Circuits	162
6.1.3	Point-to-Point and Point-to-Multipoint Interfaces	163
6.1.4	X.25 Addressing Schemes	164
6.1.5	X.25 Routing	167



6.2	Setup Tool Menus	169
6.3	X.25 Features	185
6.3.1	How do I Configure an X.31 Link (X.25 in the D-Channel)?	186
6.3.2	How do I Configure X.31 in the B-Channel (Case A/Case B)?	189
6.3.3	How do I Configure X.25 Access for a Host on my LAN?	194
6.3.4	How do I Configure ISDN Dialup Access for an X.25 Partner?	198
6.3.5	How do I Configure X.25 Dialout Without Configuration?	200
6.3.6	How do I Route IP Traffic over X.25 with MPX25?	205
6.3.7	How do I Use the Router as a TCP-X.25 Bridge?	207
6.3.8	How do I Configure the Routing for Using an X.25 PAD?	211
6.4	X.25 Utilities	215
6.4.1	X.25 PAD	215
6.5	X.25 Diagnostic Code	240
6.5.1	Clear Causes	241
6.5.2	Diagnostic Causes	242
6.5.3	Restart Causes	247
6.5.4	Reset Causes	247
6.6	X.25 Syslog Messages	248
6.7	X.21 Communications Module	256
6.7.1	CM-X21Adapter	256
7	Frame Relay	259
7.1	An Overview of Frame Relay Technology	261
7.2	Protocol Structure	264
7.2.1	Frame Relay Protocol Stack	264
7.2.2	Frame Relay Frame Format	265
7.2.3	Frame Relay Addressing	265
7.2.4	Congestion Notification	265
7.2.5	Virtual Circuits	266
7.2.6	Data Link Connection Identifier	266



7.3	Frame Relay Services	267
7.3.1	Committed Information Rate	267
7.3.2	Committed Burst Rate	267
7.3.3	Excess Burst Rate	267
7.4	The Frame Relay Subsystem	268
7.4.1	Overview: Frame Relay System Tables	268
7.4.2	Frame Relay Setup Tool Menus	269
7.4.3	Setup Tool Menus	271
7.5	Example Configuration using Setup Tool	280
7.5.1	Frame Relay over ISDN Lines	280



1 About this Manual

This manual provides a complete description of all the complex, separately licensable features available for the BinTec BIANCA/BRICK and BinGO! routers. These include Open Shortest Path First (OSPF), Remote Authentication Dial-In User Service (RADIUS), Token Authentication Firewall (TAF), Virtual Private Networking (VPN), X.25 and Frame Relay. A general description of the individual chapters and their contents will be given later in this chapter.

1.1 About your User Documentation

Your complete product documentation consists of the printed *User's Guide*, introductory *Getting Started* and *Los Geht's* manuals (optional) and the online references *BRICKware for Windows*, *Extended Features Reference*, *Software Reference*, and *The Management Information Base*.

This document describes extended features available on BinTec products that require a separate software license. Depending on your particular product some of the features described in this document may not be available on your system. For information regarding which supplemental features can be licensed for your product, consult your local BinTec product distributor.

1.1.1 How to Get the Latest Software and Documentation

Please visit our WWW server for current information on all BinTec products. Via our WWW server BinTec provides you free of charge with the most recent versions of:

- User documentation for your BinTec software/hardware
- System software for your product
- Release notes for upgrading your system software
- Windows software and UNIXtools applications


1.1.2 Contents





This manual is structured in the following way:

Chapter	Content
1: About this Manual	General Introduction.
2: OSPF	Describes using the Open Shortest Path First interior routing protocol on your BinTec router.
3: RADIUS	Describes using your BinTec router as a Remote Access Dial-In User Service Client.
4: TAF	Describes Token Authentication Firewall support on your BinTec router.
5: VPN	Describes using your BinTec router to implement Virtual Private Networking.
6: X.25	Describes operating your BinTec router in an X.25 environment.
7: Frame Relay	Describes using your BinTec router as Frame Relay router.




1.1.3 Conventions Used in this Guide

To help you locate and interpret information easily, this manual uses the following visual aids:

Symbol	Meaning
	Points out useful and relevant tips and tricks

Symbol	Meaning
	Predicts potential pitfalls and explains how to avoid them
	Brings to your attention general and important points
	Explains required fundamental information
	Brings your attention to important safety precautions. Levels of danger are in accordance with ANSI: <ul style="list-style-type: none"> ■ Caution (indicates possible danger that, if unheeded, could cause material damage) ■ Warning (indicates possible danger that, if unheeded, could cause bodily harm) ■ Danger (indicates danger that, if unheeded, could lead to serious bodily harm or death)

In order to help you find and interpret the information in this manual, the following typographical elements are used:

Typography	Meaning
	Here you are requested to do something
 —	Lists including two levels
MENU  SUBMENU	Indicates menus and submenus in the Setup Tool

Typography	Meaning
Non-proportional (Courier), e. g. ping 192.168.1.254	<ul style="list-style-type: none"> ■ Indicates commands (e. g. in the SNMP shell) that you must enter as shown ■ Used for drawings of the Setup Tool
<IP address>	Indicates commands (e. g. in the SNMP shell). Enter the value of the term in brackets. Do not enter the corner brackets.
<i>bold, italics, e.g. BigBoss</i>	Indicates example terms
bold, e. g. >> MIB	Indicates terms that you can find in the glossary. (For online texts, click the double arrow)
bold, e. g. biboAdmLoginTable, Windows Start menu	<ul style="list-style-type: none"> ■ Indicates fields in Setup Tool and MIB tables/variables ■ Indicates keys/key combinations and Windows terms
<i>italics, e. g. none</i>	Indicates values that can be entered or set in Setup Tool or MIB variables
<u>Online: underlined</u>	Indicates links

2 Open Shortest Path First (OSPF)

In this chapter we will describe the Setup Tool menus and settings you will see while using Setup Tool to configure the Open Shortest Path First (OSPF) protocol on your router.

After that, we have included an overview of the OSPF protocol as well as an example OSPF installation using different BinTec routers.

2.1 Setup Tool Menus

After entering `setup` from the shell prompt, Setup Tool's Main Menu is displayed as below. Depending on your hardware setup and software configuration your router's menu may differ slightly.

BRICK Setup Tool		BinTec Communications AG MyBRICK	
Licences	System		
Slot1:	CM-BNC/TP, Ethernet		
Slot2:	CM-2XBRI, ISDN S0, Unit 0 CM-2XBRI, ISDN S0, Unit 1		
Slot3:	CM-1BRI, ISDN S0		
WAN Partner			
IP	IPX	PPP	X.25 VPN
Configuration Management			
Monitoring and Debugging			
Exit			
Press <Ctrl-n>, <Ctrl-p> to scroll through menu items, <Return> to enter			

2.1.1 OSPF

The starting point for all OSPF settings:

➤ Go to *IP* ➤ *OSPF*.

OSPF on the router can be configured from Setup Tool using the three menus available here:

BRICK Setup Tool	BinTec Communications AG
[IP] [OSPF]: OSPF Configuration	MyBRICK
Static Settings Interfaces Areas Exit	
Press <Ctrl-n>, <Ctrl-p> to scroll through menu items, <Return> to enter	

Field	Meaning
Static Settings	Contains global OSPF parameters. This is where OSPF is enabled on the router.
Interfaces	Lists all OSPF capable router interfaces and is used for configuring interface-specific settings.
Areas	Lists all known OSPF areas and is used for adding/configuring area-specific settings.

Table 2-1: **OSPF CONFIGURATION**

2.1.2 Static Settings

To obtain the global settings for the OSPF protocol:

- Go to **STATIC SETTINGS**.

BRICK Setup Tool	BinTec Communications AG
[IP][OSPF][STATIC]: OSPF Static Settings	MyBRICK
OSPF	disabled
Generate Default Route for the AS	no
SAVE	CANCEL
Use <Space> to select	

Field	Meaning
OSPF	Is used to enable or disable OSPF. A valid license is also required before OSPF can be used on the router.
Generate Default Route for the AS	When set to <i>yes</i> the router advertises a default route over all active OSPF interfaces (see the Admin Status field in the <i>IP ► OSPF ► INTERFACES</i> menu).

Table 2-2: *OSPF STATIC SETTINGS*

Special consideration should be given to deciding which router is to provide a default route. This router should have the appropriate routes so that it can properly handle traffic for the AS.

2.1.3 Interfaces

To obtain a list of the router interfaces OSPF can be configured for:

► Go to *INTERFACES*.

By default, all IP compatible interfaces (present at the time OSPF was enabled) are added to this list and are placed in the passive state.

To configure an interface:

- Scroll to the appropriate entry and press **Enter**.

The fields shown in the resulting **EDIT** menu shown below can be configured separately for each interface.

Interface Configuration via Setup Tool

BRICK Setup Tool		BinTec Communications AG
[IP][OSPF][INTERFACE][EDIT]: Configure Interface en1		MyBRICK
Admin Status	passive (propagate routes)	
Area ID	0.0.0.0	
Metric Determination	auto (ifSpeed)	
Metric (direct routes)	10	
Authentication Type	none	
Authentication Key		
Export indirect static routes	no	
	SAVE	CANCEL
Use <Space> to select		



Once an interface is placed in the active state (and saved to memory), OSPF connections may be established over the interface resulting in appropriate costs for dial-up interfaces.



For dialup interfaces the Base Metric Value changes dynamically as ISDN channels are added/removed while the link is up. For leased line interfaces the base metric is equivalent to the result of the same function less 20 (i.e., 1542 for one leased B-Channel, 781 for two B-channels).

Field	Meaning
Admin Status	<p>The status of an OSPF interface defines whether routes and/or OSPF protocol packets are propagated over the interface.</p> <p>If OSPF has not been enabled yet, only the Admin Status field is displayed (in which case changes are irrelevant).</p> <p>OSPF routers propagate a Router Link (RL), one per Area, which identifies the router's interfaces in that Area. Both active and passive interfaces are identified in the RL. Status may be active, passive, or off with the following results:</p> <ul style="list-style-type: none"> ■ <i>Active</i>: OSPF is running over this interface ■ <i>Passive</i>: OSPF is not running over this interface OSPF protocol packets are neither sent nor received over the interface, however this interface may be included in other Router Links. ■ <i>Off</i>: OSPF is not running over this interface this interface is not included in Router Links.
Area ID	Identifies the Area this interface is assigned to.
Metric Determination	Determines how the metric for this interface is calculated. This is the cost of the link that is propagated via link state advertisements see table 2-4, page 24 .

Field	Meaning
Metric	<p>Identifies the base metric value, or cost of this interface. For <i>auto</i> determination values (see table 2-4, page 24) the actual metric used is adjusted starting a base metric value which is a simple function of the bandwidth of the physical medium (except leased line interfaces) use the function</p> $\text{Base Metric Value} = \frac{1000,000,000}{\text{bandwidth in bps}}$ <p>This results in 10 for ethernet, 6 for token ring, and 1562 for dialup ISDN interfaces (1 B-Channel).</p> <p>For fixed determination values (see previous field) the base metric value can be configured here.</p>
Authentication Type	<p>The type of authentication to use when sending (or verifying incoming) OSPF packets via this OSPF interface. This determines how the key in the Authentication Key field is used.</p> <p>By default this is set to <i>none</i>. With <i>simple</i>, Key is transmitted as a text string in each packet. With <i>md5</i>, Key is used to create (verify) an encrypted digest which is sent with each packet.</p>
Authentication Key	<p>A text string to use in connection with the Authentication Type set above.</p>

Field	Meaning
Import indirect static routes	If set to <i>no</i> (default) only direct routes for this interface are propagated over active OSPF interfaces (see the Admin Status field). When set to <i>yes</i> , indirect static routes are also propagated over active interfaces and are contained in external advertisements.

Table 2-3: *CONFIGURE INTERFACE EN1*

Although practical for sites using WAN interfaces without transfer networks, caution should be given to avoiding routing loops when importing indirect static routes.

Determination	Meaning
<i>auto</i>	The metric = the value of the base metric which is based on the bandwidth (<i>ifSpeed</i>) of the interface.
<i>fixed</i>	The metric defined (configurable) in the following field is always used (no adjustment).
<i>auto + adjust</i> (Only valid for Dial-up interfaces)	When the dial-up interface is in the up state, the metric = <base metric value> - 10. Otherwise metric = <base metric value>.
<i>fixed + adjust</i>	When the dial-up interface is in the up state the metric = <base metric value> - 10. Otherwise metric = <base metric value>.

Table 2-4: *Metric Determination*

2.1.4 Areas

To obtain a list of the OSPF Areas known to the router:

- Go to **AREAS**.
Before a router interface can be assigned to an Area, the **Area ID** must first be added here.

The exception is the backbone area which is automatically generated at boot time if no other area is configured and which all interface assignments default to if not explicitly assigned.

- To edit area-specific settings select the **Area ID** and press **Enter**.

BRICK Setup Tool	BinTec Communications AG
[IP][OSPF][AREA][ADD]: Area Configuration	MyBRICK
Area ID	0.0.0.0
Import external routes	no
Area Ranges>	
SAVE	CANCEL
Enter IP address (a.b.c.d or resolvable hostname)	

Field	Meaning
Area ID	Identifies the OSPF Area this entry corresponds to. The backbone area is <i>0.0.0.0</i> .
Import external routes	Specifies whether external routes should be imported for this area. When set to <i>no</i> , this Area is defined as an OSPF Stub Area.
Area Ranges	This submenu specifies IP Address ranges for route condensation among areas.

Table 2-5: **AREA CONFIGURATION**

2.1.5 Monitoring and Debugging

This menu consists of several submenus which allow you to monitor the router's operational status (and debug problems) in different ways:

➤ Go to **MONITORING AND DEBUGGING**.

BRICK Setup Tool	BinTec Communications AG
[MONITOR]: Monitoring and Debugging	MyBRICK
ISDN Monitor X.25 Monitor Interfaces Messages TCP/IP OSPF EXIT	

Field	Meaning
ISDN Monitor	lets you track incoming and outgoing ISDN calls
X.25 Monitor	lets you track incoming and outgoing X.25 calls
Interfaces	lets you monitor traffic by interface
Messages	displays system messages generated by the router's system logging and accounting mechanisms.
TCP/IP	menu lets you monitor IP traffic by protocol
OSPF	menu lets you monitor OSPF related information

Table 2-6: **MONITORING AND DEBUGGING**

- Go to **OSPF**.
The OSPF monitor is divided horizontally in three sections and displays information relating to OSPF Interfaces, Neighbors, and Areas.

BRICK Setup Tool		BinTec Communications AG			
[MONITOR][OSPF]: OSPF Monitor		MyBRICK			
Interface	DR	BDR	Admin Status	State	
en1	192.168.30.1	192.168.30.0	active	BDR	
brickxs	0.0.0.0	0.0.0.0	active	PTP	
Neighbor	Router ID	Interface	Retx Queue	State	
192.168.30.1	10.0.1.1	en1	0	full	
12.0.0.2	11.0.0.2	brickxs	0	full	
Area	Type	Link State ID	Router ID	Sequence	Age
0.0.0.0	Summary Net	10.0.0.0	10.0.1.1	0x800000003	1641=
0.0.0.0	Network Link	192.168.30.1	10.0.1.1	0x800000001	361 I
11.0.0.0	Router Link	11.0.0.2	11.0.0.2	0x800000009	1 I
11.0.0.0	Summary Net	0.0.0.0	192.168.40.3	0x800000001	2 V
EXIT					
Press <Ctrl-n>, <Ctrl-p> to scroll					

Interfaces Section The Interfaces section lists all enabled OSPF interfaces (interfaces that have NOT been turned “off” in the **IP ► OSPF ► INTERFACES** menu)

Field	Meaning
Interface	The router interface the entry corresponds to.
Designed Router (DR)	The Designated Router’s IP address on this interface (a DR is not shown for Point-To-Point interfaces).
Backup Designed Router (BDR)	The Backup Designated Router’s IP address on this interface (a BDR is not shown for Point-To-Point interface).
Admin Status	Only active and passive interfaces are shown here (see the IP ► OSPF ► INTERFACES menu in Interface Configuration via Setup Tool, page 21).

Field	Meaning
State	<p>The OSPF status (ospflfState) of the interface shown here may be</p> <ul style="list-style-type: none">■ <i>down</i>: OSPF is not running on this interface.■ <i>wait</i>: The initial phase of OSPF where DR and BDR are determined.■ <i>PTP</i>: The interface is a Point-To-Point interface. No DR or BDR is shown.■ <i>DR</i>: The router is the Designated Router for this interface.■ <i>BDR</i>: The router is the Backup Designated Router for this interface.■ <i>DRouter</i>: Another router is the DR/BDR for this interface.

Table 2-7: **OSPF MONITOR**

Neighbor Section The Neighbor section lists the OSPF neighbor routers that have been identified via the HELLO protocol.

Field	Meaning
Neighbor	The neighbor router's address on this interface.
Router ID	The neighbor router's system wide Router ID.
Interface	The router interface this router was identified over.
Retx Queue	The size of the retransmission queue for this neighbor. This is the number of advertisements that need to be sent to (and acknowledged from) this neighbor.
State	<p>The state of OSPF with this neighbor router may be</p> <ul style="list-style-type: none"> ■ <i>init</i>: The initial phase. A HELLO packet was received from this neighbor. ■ <i>twoWay</i>: Bidirectional communication with the neighbor. Transmitted HELLO packets have been accepted by the neighbor router (parameters are correct). ■ <i>EXstart</i>: The exchange of Database Description Packets between the router and neighbor has begun. ■ <i>exchange</i>: Actively exchanging Database Description Packets with the neighbor router. ■ <i>loading</i>: The router and the neighbor router are now exchanging Link State Advertisements. ■ <i>full</i>: The router and neighbor routers' Link State Database are now synchronized.

Table 2-8: **NEIGHBOR SECTION**

LSDB Section The Link State Database section lists the headers for all Link State Advertisements (LSA).

Field	Meaning
Area	The Area database to which this LSA belongs
Type	The type of LSA. Five types of LSAs exist: Router Link, Network Link, Summary Link, Summary ASBR, and AS External
Link State ID	The LSA's Link State ID. The Link State ID's meaning depends on the Type of advertisement
Router ID	Identifies the router that generated this LSA
Sequence	This advertisement's sequence number. Sequence numbers allow routers to determine if their database is current or if needs to request an update.
Age	The age (in seconds) of this LSA

Table 2-9: **LSDB SECTION**

2.2 Overview of the OSPF Protocol

OSPF (Open Shortest Path First), is an interior routing protocol that is often used by larger network installations as an alternative to RIP (Routing Information Protocol). It was originally designed to address some of the limitations of RIP (when used in larger networks). Here are some of the problems (with RIP) that OSPF addresses:

- **Faster Network Convergence**
Changes in routing information are propagated immediately when changes occur and not periodically as with RIP.
- **Reduced Network Load**
After a brief initialization phase, routing information does not need to be refreshed as in RIP where the entire routing table is broadcast every 30 seconds.
- **Routing Authentication**
Routers advertising OSPF routes can be authenticated.
- **Routing Traffic Control**
OSPF areas can be closed to limit the amount of traffic resulting from routing advertisements.
- **Link-Costs**
When calculating a route's cost OSPF can account for the different transport mediums such as LAN or WAN links.
- **No hop-count limitations**
In RIP, routes spanning more than 15 hops are unreachable. Although the OSPF protocol is more complex than RIP the basic concept is the same; the best interface must be calculated for forwarding packets to a particular station.

2.2.1 Shortest Path Routing

With RIP, routes are measured and selected according to the number of hops it takes for a packet to reach its destination. In the diagram below, each node

represents an IP router. According to RIP, the best route for a packet travelling from A to C will always be ABC.

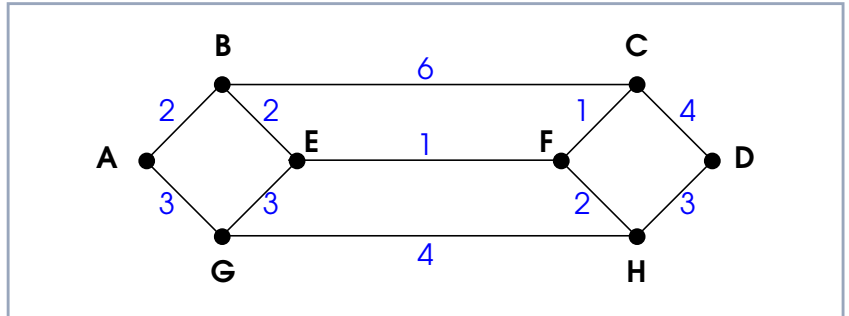


Figure 2-1: Shortest Path Routing

In OSPF each link has a cost associated with it (typically some fixed number divided by the bandwidth of the link). Routes are calculated and selected according to the least cost of the overall path a packet will travel. Thus in shortest-path routing the best path is also the fastest path (theoretically), regardless of the number of stations a packet travels through.

Assuming the relative costs of the links in the diagram above (shown in blue), according to OSPF the best route for a packet travelling from A to C is ABEFC (cost = 6). This route requires 4 hops as opposed to the 2 hop route (ABC) selected.

2.2.2 OSPF Routers and Link State Advertisement

OSPF is based on a concept of Areas. An Autonomous System (AS) consists of one or more Areas defined by network management. An Area may contain one or more IP networks.

If an AS does contain more than one area, one must be designated as the backbone, area: 0.0.0.0. All Area Border Routers (see [chapter 2.2.4, page 35](#)) in an AS must have a physical connection to the backbone.

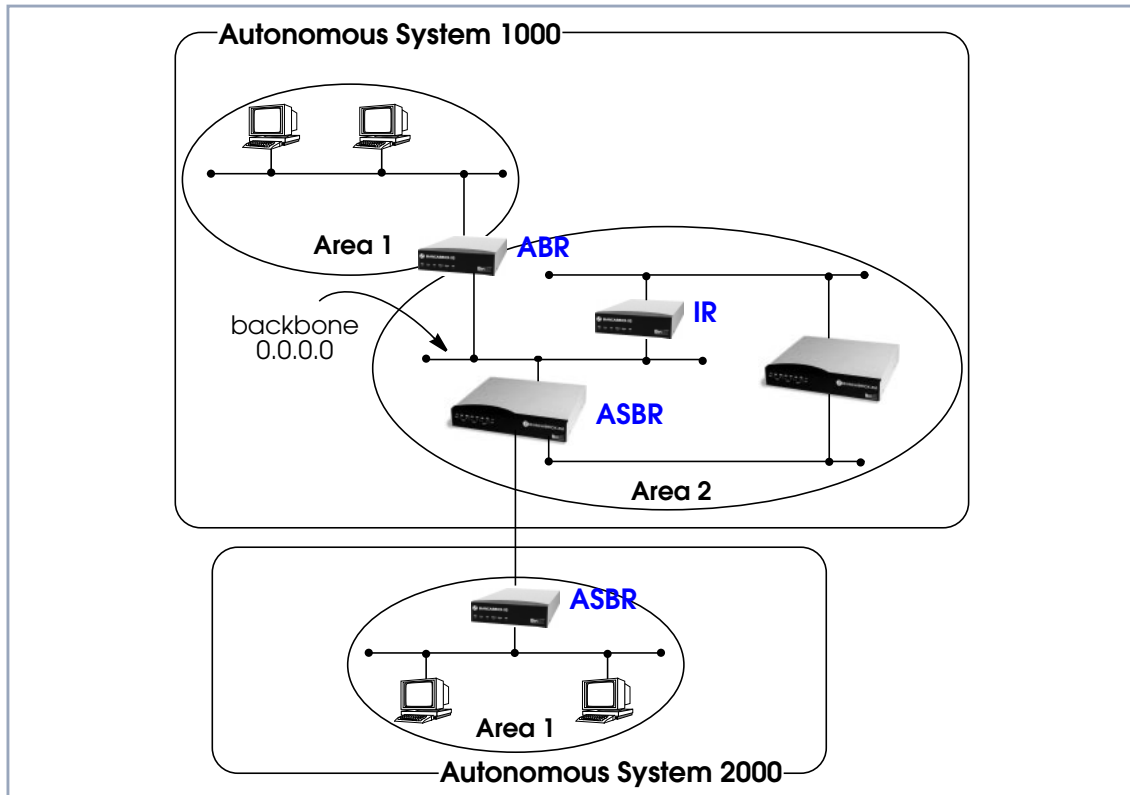


Figure 2-2: OSPF Routers and Link State Advertisement

Any of the routers shown above could additionally be the Designated Router or Backup Designated Router for its respective network.

2.2.3 OSPF Virtual Links

Note that in OSPF the backbone, Area 0.0.0.0, is the center for all areas in the Autonomous System. However, sometimes it is not possible to physically connect all areas to the backbone. By configuring a “Virtual Link” between two area border routers a remote area can still be assigned to the backbone.

As shown in the diagram below, a virtual link is established between two Area Border Routers that share a common area; called the “transit area”. Both routers must be physically connected to the backbone.

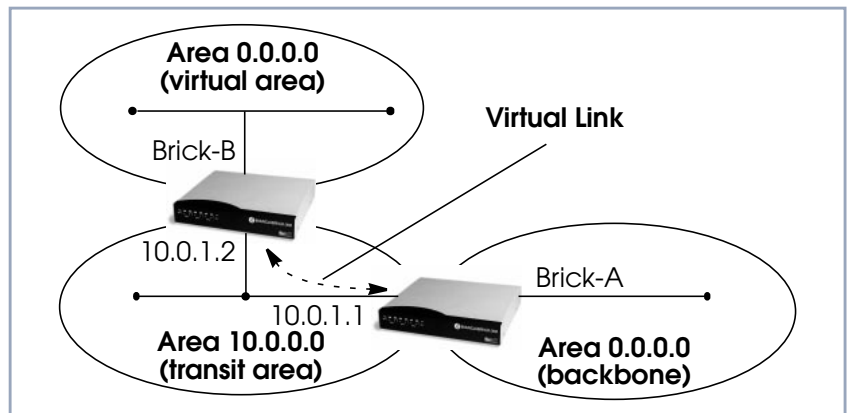


Figure 2-3: OSPF Virtual Links

2.2.4 Router Types

The location of a router’s interface with respect to an area determines the type of router it is and the types of Link State Advertisements it exchanges with other routers in that area.

■ Internal Routers (IR)

A router whose interfaces are within the same area. All Internal Routers compute the shortest path tree to all destinations within its area.

- **Area Border Router (ABR)**
A router with interfaces in different areas but within the same autonomous system. Topological information is gathered (and stored) for each attached area allowing the ABR to compute the shortest path tree for each area separately.
- **Autonomous System Border Router (ASBR)**
A router that acts as a gateway between OSPF and external routes (i.e., routes provided by other routing protocols, static indirect routes, etc.). These routers propagate routes to external networks.
- **Designated Router (DR)**
On broadcast networks (token ring and ethernet) where more than two routers are present only the DR needs to synchronize its link state database with other routers.
- **Backup Designated Router (BDR)**
A backup router assumes the responsibilities performed by the DR if that system goes down.

2.2.5 Link State Advertisement Types

OSPF routers exchange routing information via Link State Advertisements (LSAs) that contain information about the networks that can be reached over the router's interfaces.

Link State Advertisements are broken down into five different types shown in the table below. The example network shown on the previous page is redisplayed below and shows where the different types of LSAs would be found in an OSPF network.

LSA Type	Purpose
Router Links	Generated by: All OSPF Routers. Purpose: Contains information regarding the state of a router's interfaces within a particular area. Router Links are only flooded within a single area.
Network Links	Generated by: The Designated Router (or Backup Designated Router). Purpose: Identifies all OSPF routers present on the network segment and their state. These links are only flooded within a single area.
Summary Links	Generated by: Area Border Routers. Purpose: Identifies the presence of networks within an AS but outside the (local) area. Provides Inter-Area routes allowing routers to learn of networks in other Areas but within the AS.
ASBR Summary Links	Generated by: An Area Border Router. Purpose: A special type of summary link that provides routes to Autonomous System Border Routers allowing other routers in the AS to find their way out of the system.
External Links	Generated by: An Autonomous System Border Router. Purpose: Contains information about other Autonomous Systems and allows routers to learn about routes to networks there. External links are flooded into all areas except stub areas.

Table 2-10: Link State Advertisement Types

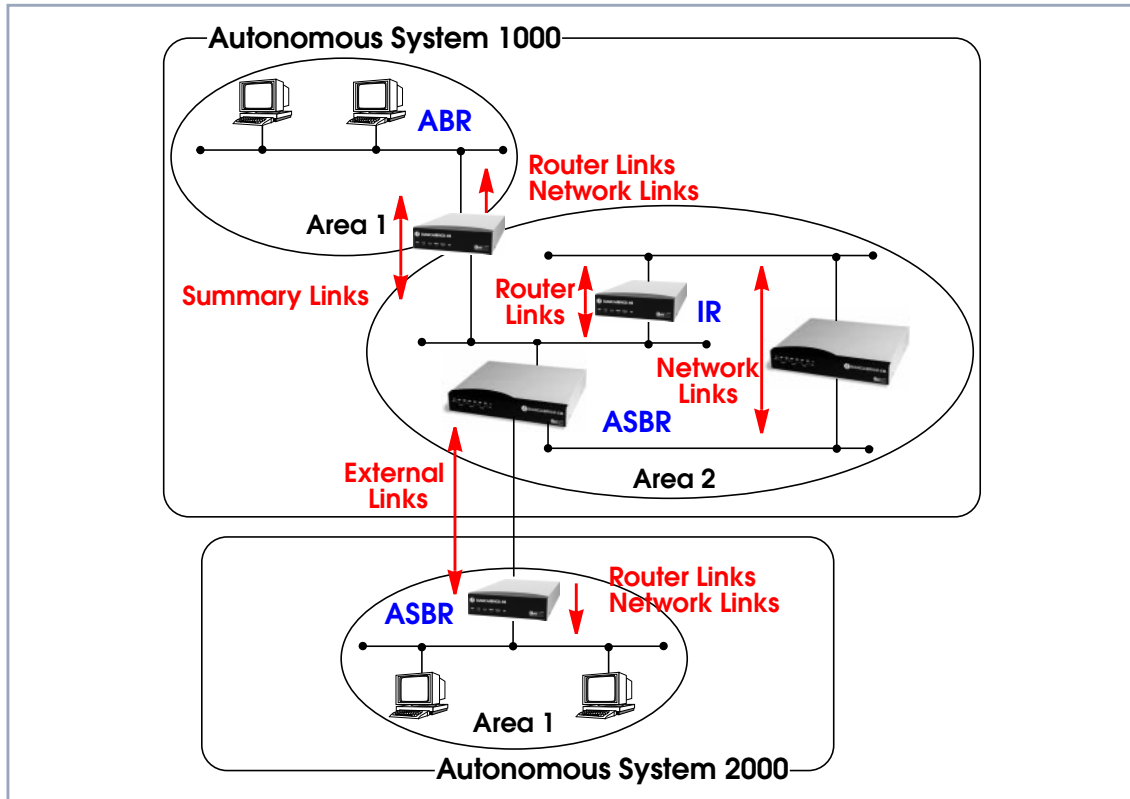


Figure 2-4: Different LSA Types in OSPF Network

2.2.6 Router Identification

All OSPF routers in an Autonomous System must have a unique Router ID that identifies the router with respect to the AS. Generally an OSPF router's Router ID is taken to be the highest IP address for its first LAN interface.

2.2.7 Initialization

OSPF networks are said to be much "quieter" in comparison to RIP based networks. This is because in OSPF once the initialization phase is complete routing

information is only exchanged when link state changes occur. This is much different than with RIP where every 30 seconds a router's complete routing table is broadcast and verified over the network.

The initialization phase of OSPF is completed once the Link State Database for the area has stabilized and generally occurs once:

- The OSPF Neighbors have been identified.
- The Designated and Backup Designated Routers have been established.

2.2.8 Neighbor Identification

When first coming into service an OSPF router attempts to identify its neighbor OSPF routers using the HELLO protocol. Two routers are neighbors if they:

- Share a common network.
- Are using the same Area Number for that segment.
- Are using the same Authentication for the segment.
- Are using the same parameters (HELLO interval, etc.).

Neighbor routers then decide whether to synchronize their Link State Database (LSDB) with one another. All routers on the segment synchronize their LSDBs with the Designated Router (DR) and the Backup Designated Router (BDR).

2.2.9 Designated / Backup Designated Router Election

When Neighbor routers are identified (via the HELLO protocol) the DR and BDR are also identified. This is sometimes called DR and BDR election and is achieved via IP multicast packets which a router broadcasts via each network segment. For each segment the router with the highest OSPF priority generally

becomes the DR. In case of a tie, the router with the higher Router ID becomes the DR.

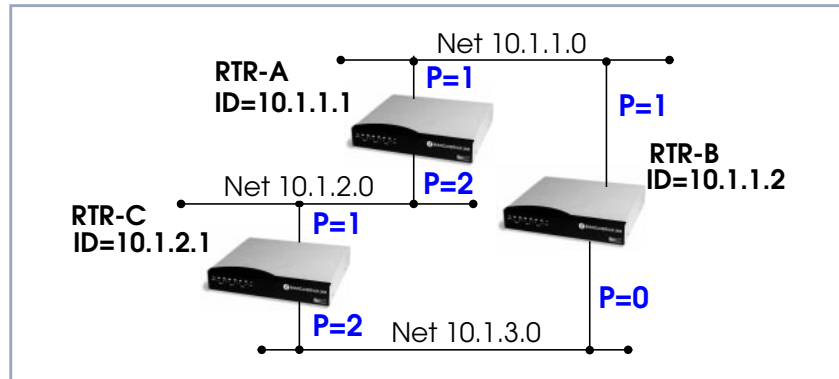


Figure 2-5: Designated/Backup Designated Router Election

The DR and BDRs for the three networks shown above would be elected as follows:

Network	DR	BDR
10.1.1.0	RTR-B	RTR-A
10.1.2.0	RTR-A	RTR-C
10.1.3.0	RTR-C	RTR-B

2.2.10 Building up the LSD and the SPT

Link State Database (LSD) Link State Advertisements contain information about a router's interfaces (i.e., link's IP address, mask, network type, networks reachable over the link, etc.).

All routers within an area receive all link state information for all routers in the area. Once synchronized each router has an identical image of the link state database that describes the topological structure of the area.

Shortest Path Tree (SPT) This database allows each router to separately calculate a shortest path tree (SPT), using itself as the root, to any destination in the area. The SPT is used to determine the best interface to route a packet. As in RIP the lowest cost route

is used however the cost to a destination is calculated differently. In OSPF the cost (or metric) of a link is a function of the bandwidth provided by the link. The higher the bandwidth, the lower the cost.

2.2.11 Authentication

OSPF allows packets containing OSPF routing information to be individually authenticated. Two authentication methods are available which must be configured separately for each network segment.

- Simple (password) authentication
A simple text string is sent with each packet. This method is less secure since packet contents can be “sniffed” off the wire using a link analyzer.
- MD5 (cryptographic) authentication
When MD5 (Message Digest) is used, each packet is appended with a 16 byte encrypted digest. The digest is a function of an authentication key and the contents of the packet. This method is more secure since the key is not sent with the packet.



With MD5 authentication, only the digest is encrypted and not the actual contents of the OSPF packet.

2.2.12 OSPF over Demand Circuits

Although OSPF generates less network traffic than RIP, the occasional exchange of routing information (HELLO packets, Link State Database updates or changes, etc.) can lead to increased costs for dial-up interfaces.

To help minimize these costs, OSPF on the **BRICK** has been implemented to include special extensions for Demand Circuits as defined in RFC 1793, OSPF

over Demand Circuits. These extensions allow for efficient use of dial-up interfaces with OSPF and avoiding excessive ISDN costs. In particular, this means:

- The exchange of HELLO packets between neighbors is suppressed once the **BRICK** has synchronized its LSDB with that neighbor (a dial-up connection is initially opened to synchronize the database.)



Link State advertisements are only flooded to neighbor routers when an actual change needs to be propagated.

Each LSA is marked with a special DoNotAge flag (identifiable by the DC-bit of the LSA or OSPF packet)



If a router without RFC 1793 support is removed from the domain in which this feature has been used it is recommended that all OSPF routers be briefly deactivated and re-activated to ensure that all LSAs generated by the removed router are actually flushed.

2.3 Example OSPF Installation

A typical network installation showing how OSPF could be put to use is shown in the diagram below. Highlights for this setup are shown below. Following the diagram is a Configuration Overview and following that a detailed listing of the configuration steps is provided for each router.

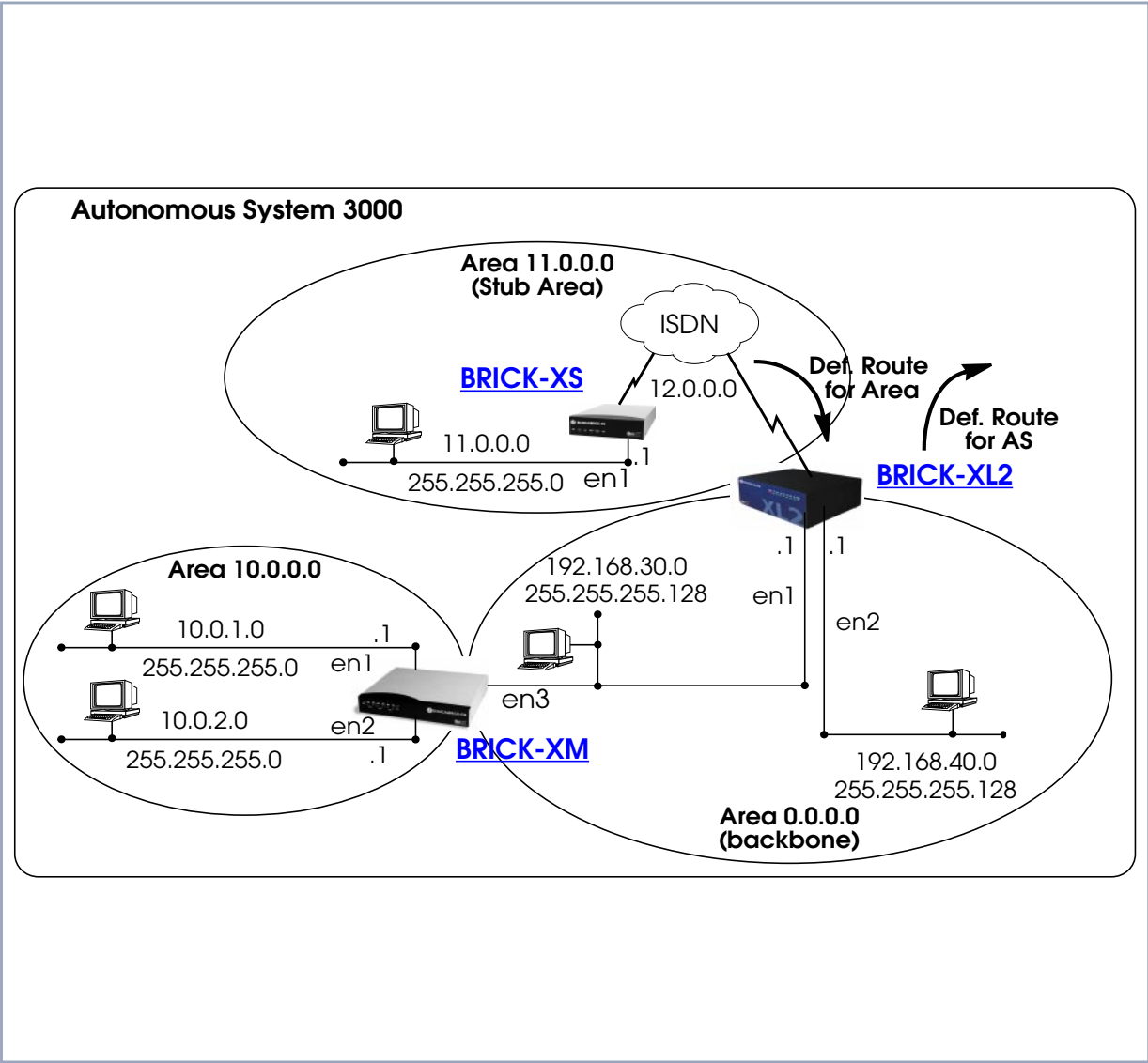


Figure 2-6: OSPF Installation

Area 11.0.0.0 (stub area)

- Since the remote LAN in Area 11.0.0.0 is linked to the backbone via an ISDN dialup link this area is configured as a stub area. This means that external routing information advertisements will not flow into this area. The default route for this area is provided by the router BRICK-XL2.
- Because OSPF on the **BRICK** includes support for Demand Circuits (RFC 1793) the dialup link is only opened when changes in routing information must be propagated.

Area 0.0.0.0 (backbone)

Area 0.0.0.0 is the backbone of the Autonomous System. The router at BRICK-XL2 will provide the default route for the entire AS and a default route for Area 11.0.0.0.

Area 10.0.0.0

Area 10.0.0.0 is connected to the backbone via the border router BRICK-XM. Since this is the only link between networks in this area and any external networks (such as the Internet) BRICK-XM will provide Summary Links to routers in other areas. This means that routing information about networks in Area 10.0.0.0 will be combined (or aggregated) into a single advertisement. This lessens the amount of traffic on the backbone and keeps the size of the link state database for area 0.0.0.0 small.

2.3.1 Configuration Overview

Prerequisite for all BinTec routers:

- A valid OSPF license must be installed. This can be added to the **biboAdmLicenseTable** or from Setup Tool's **LICENCES** menu.
- OSPF must be enabled by setting **ospfAdminStat** to enabled, or from Setup Tool's **IP** ➤ **OSPF** ➤ **STATIC SETTINGS** menu.

BRICK-XL2 Overview

- Create the dial-up partner interface to BRICK-XS.
- Have BRICK-XL2 advertise the default route for the AS.

- Create the Area entry for Area *11.0.0.0*.
- Assign the new dialup partner interface to Area *11.0.0.0* and set the interface to *active*.

BRICK-XM Overview

- Create the Area entry for Area *10.0.0.0*.
- Assign ethernet interfaces en1 and en2 to Area *10.0.0.0* and set both interfaces to *active*.
- Verify ethernet interface en3 is assigned to Area *11.0.0.0* and set the interface to *active*.
- Create the OSPF aggregate for the LANs attached to en1 and en2 to reduce the routing traffic sent over en3.

BRICK-XS Overview

- Create the dial-up partner interface to BRICK-XL2.
- Create the Area entry for Area *11.0.0.0*.
- Assign the ethernet interface (en1) to Area *11.0.0.0* and set the interface to *active*.
- Assign the new dial-up interface to Area *0.0.0.0* and set the interface to *active*.

2.3.2 Configuration Steps for BRICK-XL2

- Enable OSPF and create the partner interface to BRICK-XS. Note that our example uses a transfer network (network 12.0.0.0).
- Since BRICK-XL2 should advertise the default route for the AS go to **IP** ➤ **OSPF** ➤ **STATIC SETTINGS** and set the **Generate Default Route for the AS** field to yes.

BRICK Setup Tool	BinTec Communications AG
[IP][OSPF][STATIC]: OSPF Static Settings	MyBRICK
OSPF	enabled
Generate Default Route for the AS	yes
SAVE	CANCEL
Enter IP address (a.b.c.d or resolvable hostname)	

- In the **IP** ➤ **OSPF** ➤ **AREAS** menu create an entry for Area **11.0.0.0**. Define this area as a Stub Area and have BRICK-XL2 generate the default route for this area.

BRICKSetup Tool	BinTec Communications AG
[IP][OSPF][AREA][ADD]: Area Configuration	MyBRICK
Area ID	11.0.0.0
Import external routes	no
Import summary routes	no
Create area default route (only ABR)	yes
Area Ranges>	
SAVE	CANCEL
Enter IP address (a.b.c.d or resolvable hostname)	

- In the **IP** ➤ **OSPF** ➤ **INTERFACES** menu locate the dialup interface entry created before and press **Enter** to edit the settings.
- Set the **Admin Status** to **active** and assign it to Area **11.0.0.0** (or the area created before) and select **SAVE**.

BRICK Setup Tool		BinTec Communications AG
[IP][OSPF][INTERFACE]: Configure Interface BRICK		MyBRICK
Admin Status	active (propagate routes + run OSPF)	
Area ID	11.0.0.0	
Metric Determination	auto (ifSpeed)	
Metric (direct routes)	1562	
Authentication Type	none	
Authentication Key		
Import indirect static routes	no	
	SAVE	CANCEL
Use (Space) to select		

By default, dial-up interfaces are set to *passive* in the **Admin Status** field.

- In **IP** ➤ **OSPF** ➤ **INTERFACES** menu verify the ethernet interfaces en1 and en2 are assigned to the backbone, (Area *0.0.0.0* which is the default area).
- Set the **Admin Status** to active and assign it to Area *11.0.0.0* (or the value from the step before) and select **SAVE**.

2.3.3 Configuration Steps for BRICK-XM

- Enable OSPF in **IP** ➤ **OSPF** ➤ **STATIC SETTINGS**.
- Then create an area entry for Area *10.0.0.0* in the **IP** ➤ **OSPF** ➤ **AREAS** menu.

BRICK Setup Tool		BinTec Communications AG
[IP][OSPF][AREA][ADD]: Area Configuration		MyBRICK
Area ID		10.0.0.0
Import external routes		yes
Area Ranges>		
	SAVE	CANCEL
Enter IP address (a.b.c.d or resolvable hostname)		

- In the **IP** ➤ **OSPF** ➤ **INTERFACES** menu assign ethernet interfaces en1 and en2 to Area **10.0.0.0** (or the value from the previous step) and set the **Admin Status** for each interface to active.

BRICK Setup Tool		BinTec Communications AG
[IP][OSPF][AREA][ADD]: Area Configuration		MyBRICK
Admin Status		active (propagate routes + run OSPF)
Area ID		10.0.0.0
Metric Determination		auto (ifSpeed)
Metric (direct routes)		10
Authentication Type		none
Authentication Key		
Import indirect routes		no
	SAVE	CANCEL
Use (Space) to select		

- Ethernet interface en3 should already be assigned to the backbone, Area **0.0.0.0** which is the default.
- In the **IP** ➤ **OSPF** ➤ **INTERFACES** menu verify this setting and change the **Admin Status** to **active**.
- Return to the **IP** ➤ **OSPF** ➤ **AREAS** menu and scroll to the **Area ID** entry for the backbone and press **Enter**.
- Move to the **AREA RANGES** submenu to add an OSPF aggregate for the LANs attached to en1 and en2. The Address and Mask entries shown below will match any routes with a destinations starting with 10, or 10.*.*.*.

BRICK Setup Tool		BinTec Communications AG	
[IP][OSPF][AREA][RANGE][ADD]: Configure Address range for AreaMyBRICK			
Address		10.0.0.0	
Mask		255.0.0.0	
Advertise Matching		yes	
	SAVE		CANCEL
Enter IP address (a.b.c.d or resolvable hostname)			

This entry means that BRICK-XM will consolidate multiple routes (routes for destinations in Area *10.0.0.0*) into a single link state advertisement.

This will effectively reduce the amount of traffic sent over the backbone as will help keep the size of the link state database and routing tables for routers in other areas to a minimum.

2.3.4 Configuration Steps for BRICK-XS

- Enable OSPF and create the dial-up partner interface to BRICK-XL2. In our example a transfer network (12.0.0.0) is used.
- In the **IP** ➤ **OSPF** ➤ **AREAS** menu create Area *11.0.0.0* and define it as a Stub Area.

BRICK Setup Tool		BinTec Communications AG
[IP][OSPF][AREA][ADD]: Area Configuration		MyBRICK
Area ID	11.0.0.0	
Import external routes	no	
Import summary routes	no	
Create area default route (only ABR)	no	
Area Ranges>		
	SAVE	CANCEL
Enter IP address (a.b.c.d or resolvable hostname)		

- In the **IP** ➤ **OSPF** ➤ **INTERFACES** menu assign the ethernet interface (en1) to Area **11.0.0.0** and make sure the **Admin Status** is set to *active*.

BRICK Setup Tool		BinTec Communications AG
[IP][OSPF][INTERFACES] Configure Interface en1		MyBRICK
Admin Status	active (propagate routes + run OSPF)	
Area ID	11.0.0.0	
Metric Determination	auto (ifSpeed)	
Metric (direct routes)	10	
Authentication Type	none	
Authentication Key		
Import indirect routes	no	
	SAVE	CANCEL
Use (Space) to select		

- In **IP** ➤ **OSPF** ➤ **INTERFACES** menu locate the dialup interface (created in step 1) and assign the interface to Area **11.0.0.0** (or the value used in the step before).
- Set the **Admin Status** for the dialup interface to active and select **SAVE**.

BRICK Setup Tool		BinTec Communications AG
[IP][OSPF][INTERFACES] Configure Interface dialup		MyBRICK
Admin Status	active (propagate routes + run OSPF)	
Area ID	11.0.0.0	
Metric Determination	auto(ifSpeed)	
Metric (direct routes)	1562	
Authentication Type	none	
Authentication Key		
	SAVE	CANCEL
Use (Space) to select		

2.3.5 Configuring OSPF Virtual Links

A virtual interface must be defined on each of the ABRs by creating an entry in the **ospfVirtIfTable**. This is done by setting the **ospfVirtIfNeighbor** and **ospfVirtIfAreaID** objects.

- **ospfVirtIfNeighbor** should be set to the Router ID of the Area Border Router at the other end of the virtual link.
- **ospfVirtIfAreaID** should be set to the **Area ID** of the transit area.

The virtual link in the diagram here would be configured on BRICK-A as follows:

```
BRICK-A:system> ospfVirtIfTable
inx  AreaID(rw*)      Neighbor(rw*)      TransitDelay(rw)
      Retrasitinterval(rw) Hellointerval(rw) RtrDeadInterval(rw)
      State(ro)       Events(ro)         AuthKey(rw)
      Status(-rw)     AuthType(rw)
```

```
BRICK-A:ospdVirtIfTable> AreaID=10.0.0.0 Neighbor=10.0.1.2
```

This creates a new OSPF virtual interface (on BRICK-A) that links two parts of the backbone via the transit area *10.0.0.0*. The respective interface would be created on BRICK-B using almost the same command (**ospfVirtIfAreaID=10.0.0.0 ospfVirtIfNeighbor=10.0.1.1**).



Remember that the area being used as the transit area must already be defined in the **ospfAreaTable**.

2.4 Controlling Link State Database Overflow

Sites with large (or complicated) network installations that are running OSPF may notice the Link State Database (LSDB) becoming large. Most often this is the case where external routes are being imported as external advertisements.

One way to minimize the size of the LSDB (on the **BRICK**) is to use the **ospfExtLsdbLimit** variable. This object defines the maximum number of external LSAs to store in the database (the local copy).

Once the limit is reached the **BRICK** goes into Overflow State. In Overflow State two things happen:

- The **BRICK** begins to flush all external advertisements generated locally.
- The **BRICK** ignores all new external advertisements.



The maximum size of the LSDB must be the same for all OSPF routers in the domain for this feature to perform efficiently.

By default the **BRICK** remains in overflow state but can optionally be configured to leave overflow state (and continue to process new external LSAs) automatically after a time period. The **ospfExtOverflowInterval** variable defines the number of seconds to wait before leaving overflow state automatically. The default is 0 seconds (i.e., stay in overflow state). After waiting **ospfExtOverflowInterval** seconds the number of external LSAs in the LSDB is compared to the **ospfExtLsdbLimit**. If there is room in the database for new LSAs, the **BRICK** leaves overflow state; otherwise another time interval is waited.

The diagram shown below attempts to illustrate the behavior of database overflow control using the `ospfExtLsdbLimit` and `ospfExtOverflowInterval` variables.

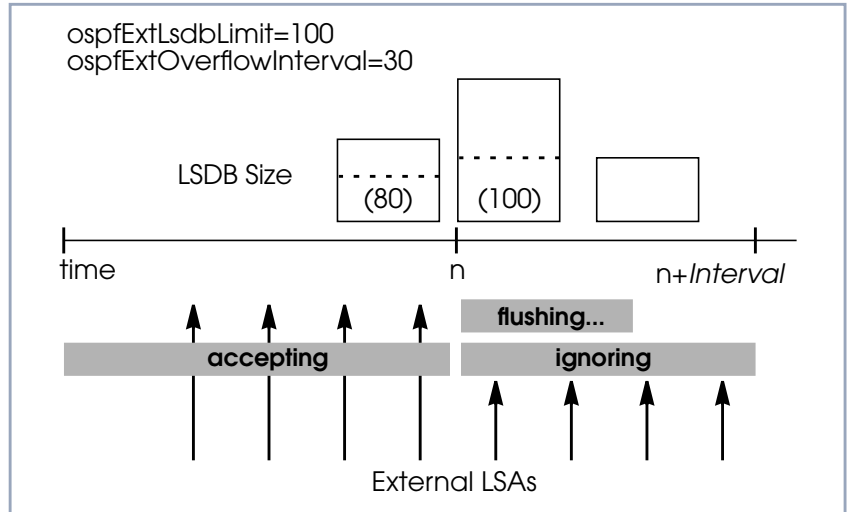


Figure 2-7: Database Overflow Control

2.5 Enabling Demand Circuit Support

Demand Circuit support for dial-up partner interfaces is enabled by default when an existing interface is enabled for OSPF (**AdminStatus** is set to *active*). Support can be manually controlled by setting the interface's **IfDemand** object (**ospfIfTable**) to *true* or *false*. When set to *false*, the state of this interface is always up.

Setting this variable to *true* for one side of the connection is sufficient (that is, as long as OSPF has been enabled on both sides, i.e., **ipExtIfOspf=active**) if both sides support RFC 1793.



Until a neighbor router has been identified HELLO packets are periodically transmitted (default, **ospfIfPollInterval** = 120 seconds) over the interface. This results in the link being opened. Once the LSDB has been synchronized, the HELLO protocol is then suppressed.

2.6 Import / Export of Routing Information

When different routing protocols are used within the same domain it is sometimes useful to be able to exchange (import or export) routing information between these protocols.

Using the **ipImportTable** routing information generated by one protocol (**ipImportSrcProto**) can be imported or exported to another protocol (**ipImportDstProto**).

Currently the following SrcProto ↔ DstProto combinations are possible:

		ipImportDstProto	
		rip	ospf
ipImportSrcProto	default route		✓(1) (see table 2-11)
	direct		
	static		✓(2) (see table 2-11)
	rip		
	ospf	✓(3) (see table 2-11)	

For further control, the fields of the **ipImportTable** allow how (and what) routing information is imported.

Variable	Meaning
ipImportSrcProto (1)	default_route ipImportDstProto =ospf This entry forces an external Link State Advertisement to be generated that defines a default route for the Autonomous System.
ipImportSrcProto (2)	static ipImportDstProto =ospf With this entry statically configured indirect routes will be propagated via OSPF as external LSAs.
ipImportSrcProto (3)	ospf ipImportDstProto =rip With this entry, all routes learned via OSPF are imported to RIP. If an OSPF route changes, the import to RIP will trigger an immediate broadcast of the entire routing table.

Variable	Meaning
ipImportMetric1	The metric in the context of the destination protocol the imported routes should get. If set to <i>-1</i> these routes get a protocol-specific default metric.
ipImportType	This object might define protocol specific properties of the imported routes in the context of the destination protocol.
ipImportAddr	Specifies (together with ipImportMask) the range of IP addresses for which the table entry should be valid. The entry is valid if the destination IP address of the route lies in the range specified by both objects. If both objects are set to <i>0.0.0.0</i> , the table entry will be valid for destination.
ipImportMask	Together with ipImportAddr specifies the range of IP addresses for which the table entry should be valid. For example, if Addr= <i>X.X.0.0</i> and Mask= <i>255.255.0.0</i> then addresses <i>X.X.0.0</i> through <i>X.X.255.255</i> are valid.
ipImportEffect	Defines the effect of this entry. If set to <i>import</i> importation from SrcProto to DstProto takes place. If set to <i>doNotImport</i> importation is prevented.
ipImportIfIndex	Specifies the interface index of the interface for which the entry should be valid. If set to <i>0</i> the entry is valid for all interfaces.

Table 2-11: **ipImportTable**

3 RADIUS

This chapter gives you information on the RADIUS implementation of BinTec Communications AG. You will learn how to configure a **BRICK** as a RADIUS client and what is necessary to know about configuring a RADIUS server. Useful examples are given.

The following items are covered:

- Overview (see [chapter 3.1, page 62](#))
- Configuration on **BRICK** side (see [chapter 3.2, page 64](#))
- Configuration on the RADIUS server (see [chapter 3.3, page 74](#))
- RADIUS attributes for Authentication (see [chapter 3.4, page 76](#))
- RADIUS attributes for Accounting (see [chapter 3.5, page 84](#))
- RADIUS for Dial-Out (see [chapter 3.6, page 87](#))
- Examples (see [chapter 3.7, page 96](#))
 - Typical dial-in (without BinTec attributes) (see [chapter 3.7.1, page 96](#))
 - Standard dial-in with CLID (see [chapter 3.7.2, page 96](#))
 - Callback PPP negotiated (see [chapter 3.7.3, page 97](#))
 - Callback (Windows client) (see [chapter 3.7.4, page 97](#))
 - Callback (CLID) (see [chapter 3.7.5, page 98](#))
 - Working with one or more RADIUS servers (see [chapter 3.7.6, page 99](#))
 - Dial-out (see [chapter 3.7.7, page 100](#))

3.1 Overview

Client / Server RADIUS (Remote Authentication Dial In User Service) is a client/server protocol originally developed by Livingston Enterprises. RADIUS provides a security system that allows you to exchange authentication and configuration information between a Network Access Server, such as the BRICK, and a RADIUS Server, a PC or UNIX machine running a RADIUS daemon process. The RADIUS server maintains a database of user authentication data and configuration information.

RADIUS can be used for:

- Authentication
- Accounting

The **BRICK** sends a request with username and password to the RADIUS server, the server examines its database. If the user is found and may connect, the RADIUS server returns an accept message to the **BRICK**. The message contains parameters (RADIUS attributes) that the **BRICK** uses for the configuration and further negotiation of the related WAN connection.

When using a RADIUS server for accounting, the **BRICK** sends an accounting start record at the beginning and a stop record at the end of every connection. These start and stop records also contain RADIUS attributes describing the connection (IP address, username, throughput, charges).

RADIUS packets The following types of packets are sent between RADIUS server and RADIUS client:

Types	Sent from → to	Purpose
ACCESS_REQUEST	Client → Server	When a connection request is received on the BRICK the RADIUS server is polled if a locally defined PPP partner could not be found (i.e., upon receiving the calling partner's PPP_ID and no local record exists for the PPP partner.).

Types	Sent from → to	Purpose
ACCESS_ACCEPT	Server → Client	If the RADIUS server authenticates the information contained in the ACCESS_REQUEST packet, it sends an ACCESS_ACCEPT packet to the RADIUS client that contains the link setup parameters to use.
ACCESS_REJECT	Server → Client	If the information contained in the ACCESS_REQUEST packet doesn't match the information in the RADIUS Server's user database (usually /etc/raddb/users) the server may deny access to the network.
ACCOUNTING_START	Client → Server	When using a RADIUS server for accounting, the BRICK sends an accounting start record at the beginning of every connection.
ACCOUNTING_STOP	Client → Server	When using a RADIUS server for accounting, the BRICK sends an accounting stop record at the end of every connection.

Table 3-1: RADIUS packets

Configuration steps The required configuration steps have to be done on:

- BRICK side (see [chapter 3.2, page 64](#))
- RADIUS server side (see [chapter 3.3, page 74](#))

RADIUS table entries The ifIndexes of RADIUS PPP entries start at 15001. They are not stored when saving your configuration.

Further information on RADIUS For further information on RADIUS, here are some useful links:

- A nice little introduction to RADIUS:
<http://www.squashduck.com/~roundman/radius/>
- A useful site with lots of information on RADIUS and sources of supply for RADIUS servers:
<http://www.dnt.ro/~vsv/radius.html>
- A BinTec FAQ concerning the Steel-Belted RADIUS server and configuring callback for Windows clients:
<http://www.bintec.de/gb/service/index.html>

3.2 Configuration on BRICK Side

The **BRICK** can be configured via

- Setup Tool (see [chapter 3.2.1, page 64](#))
- MIB variables (see [chapter 3.2.2, page 69](#))

3.2.1 Setup Tool

The menu **IP ► RADIUS SERVER** lists all RADIUS servers currently configured on the router.

BRICK		BinTec Communications AG MyBRICK	
Proto	Prio	IP Address	State
auth	0	111.11.11.11	active
Press <Ctrl-n>, <Ctrl-p> to scroll through menu items, <Return> to enter			

You can add, edit, or delete list entries in the usual fashion.

The configuration of a RADIUS server is made in **IP** ► **RADIUS SERVER** ► **ADD**:

BRICK	BinTec Communications AG
[IP][RADIUS][ADD]:Configure Radius Server	MyBRICK
Protocol	auth
IP Address	44.55.66.77
Password	blubb
Priority	0
Policy	authoritative
Port	1812
Timeout	1000
Retries	1
State	active
SAVE	
Use <Space> to select	

The menu contains the following entries:

Field	Meaning
Protocol	<p>Defines whether the RADIUS server is used for authentication purposes or for accounting ISDN connections.</p> <p>Possible values:</p> <ul style="list-style-type: none"> ■ <i>auth</i> (default value): Authentication. ■ <i>acct</i>: Accounting.
IP Address	The IP address of the RADIUS server.
Password	This is a shared secret between RADIUS server and BRICK .

Field	Meaning
Priority	<p>Priority of the RADIUS server. When there are several RADIUS server entries, the server with the highest priority is used first. If there is no reply from this server, the server with the next highest priority is used and so forth.</p> <p>Possible values: Integers from 0 (highest priority) to 7 (lowest priority). Default value: 0.</p>
Policy	<p>Defines how the BRICK reacts when receiving a negative answer to a request.</p> <p>Possible values:</p> <ul style="list-style-type: none"> ■ <i>authoritative</i> (default value): A negative answer to a request will be accepted. ■ <i>non-authoritative</i>: A negative request will not be accepted, but the next RADIUS server will be asked until there is finally an authoritative server configured.
Port	<p>Number of TCP port to use for RADIUS data.</p> <p>According to RFC 2138 the default ports are 1812 for authentication (was 1645 in older RFCs) and 1813 for accounting (was 1645 in older RFCs). Many RADIUS servers, including Merit, still use 1645 and 1646. As RADIUS servers use different port numbers, you should refer to the documentation for the RADIUS server you are using.</p> <p>Default value: 1812.</p>
Timeout	<p>Number of milliseconds to wait for an answer to a request.</p> <p>Possible values: Integers from 50 to 50000.</p> <p>Default value: 1000 (1 second).</p>

Field	Meaning
Retries	<p>Number of retries if a request is not answered. If, after these attempts, still no answer has been received, the server State is set to <i>inactive</i>. The BRICK then tries to contact the server every 20 seconds, and once the server replies, State is changed to <i>active</i> again.</p> <p>Possible values: Integers from 0 to 10. Default value: 1.</p> <p>To prevent the State switching to <i>inactive</i>, set this value to 0.</p>
State	<p>The state of the RADIUS server.</p> <p>Possible values:</p> <ul style="list-style-type: none"> ■ <i>active</i> (default value): Server answers requests. ■ <i>inactive</i>: Server does not answer (see Retries above). ■ <i>disabled</i>: Requests to a certain RADIUS server are temporarily disabled.

Table 3-2: **IP** ➤ **RADIUS SERVER** ➤ **ADD**

Menu PPP For incoming calls there are some options that can not be set user specific. They have an effect on the PPP negotiation and RADIUS server usage before the caller can be identified by username and password. These settings are entered in the menu **PPP**:

BRICK Setup Tool		BinTec Communications AG	
[PPP]:PPP Profile Configuration		MyBRICK	
Authentication Protocol	Radius Server Authentication	CHAP + PAP + MS-CHAP inband	
PPP Link Quality Monitoring		no	
SAVE		CANCEL	
Use <Space> to select			

PPP contains the following items:

Field	Meaning
Authentication Protocol	Defines the PPP authentication protocol offered to the caller first.
Radius Server Authentication	<p>Is used to configure possible RADIUS authentication on incoming calls.</p> <p>Possible values:</p> <ul style="list-style-type: none"> ■ <i>inband</i> (default value): Only inband RADIUS requests (PAP,CHAP) are sent to the specified RADIUS server. ■ <i>Calling Line Identification (CLID)</i>: Only outband requests are sent to the RADIUS server. ■ <i>CLID + inband</i>: Both requests are sent to the RADIUS server (first outband request, then inband request if necessary). ■ <i>none</i>: No requests are sent.
PPP Link Quality Monitoring	Defines whether Link Quality Monitoring is executed for PPP connections.

Table 3-3: **PPP**

3.2.2 MIB

RadiusServerTable Configuration is made over **RadiusServerTable**, it contains the following variables:

Variable	Meaning
RadiusSrvProtocol	<p>Defines whether the RADIUS server is used for authentication purposes or for accounting ISDN connections.</p> <p>Possible values:</p> <ul style="list-style-type: none"> ■ <i>authentication</i> (default value). ■ <i>accounting</i>.
RadiusSrvAddress	The IP address of the RADIUS server.
RadiusSrvPort	<p>Number of TCP port to use for RADIUS data.</p> <p>According to RFC 2138 the default ports are 1812 for authentication (was 1645 in older RFCs) and 1813 for accounting (was 1645 in older RFCs). Many RADIUS servers, including Merit, still use 1645 and 1646. As RADIUS servers use different port numbers, you should refer to the documentation for the RADIUS server you are using.</p> <p>Default value: <i>1812</i>.</p>
RadiusSrvSecret	This is a shared secret between RADIUS server and BRICK .
RadiusSrvPriority	<p>Priority of the RADIUS server. When there are several RADIUS server entries, the server with the highest priority is used first. If there is no reply from this server, the server with the next highest priority is used and so forth.</p> <p>Possible values: Integers from <i>0</i> (highest priority) to <i>7</i> (lowest priority). Default value: <i>0</i>.</p>

Variable	Meaning
RadiusSrvTimeout	<p>Number of milliseconds to wait for an answer to a request.</p> <p>Possible values: Integers from <i>50</i> to <i>50000</i>. Default value: <i>1000</i> (1 second).</p>
RadiusSrvRetries	<p>Number of retries if a request is not answered. If after these attempts still no answer was received, the RadiusSrvState is set to <i>inactive</i>. The BRICK then tries to contact the server every 20 seconds, and once the server replies, RadiusSrvState is changed to <i>active</i> again.</p> <p>Possible values: Integers from <i>0</i> to <i>10</i>. Default value: <i>1</i>.</p> <p>To prevent the RadiusSrvState switching to <i>inactive</i>, set this value to <i>0</i>.</p>
RadiusSrvState	<p>The state of the RADIUS server.</p> <p>Possible values:</p> <ul style="list-style-type: none"> ■ <i>active</i> (default value): Server answers requests. ■ <i>inactive</i>: Server does not answer (see RadiusSrvRetries above). ■ <i>disabled</i>: Requests to a certain RADIUS server are temporarily disabled. ■ <i>delete</i>: Deletes the entry.

Variable	Meaning
RadiusSrvPolicy	<p>Defines how the BRICK reacts when receiving a negative answer to a request.</p> <p>Possible values:</p> <ul style="list-style-type: none"> ■ <i>authoritative</i> (default value): A negative answer to a request will be accepted. ■ <i>non_authoritative</i>: A negative request will not be accepted, but the next RADIUS server will be asked until there is finally an authoritative server configured.
RadiusSrvValidate	<p>This additional option is only used for Bogus RADIUS servers, which send response messages with a miscalculated MD5 checksum. All messages generated by the BRICK, however, will always use the proper authentication scheme.</p> <p>Possible values:</p> <ul style="list-style-type: none"> ■ <i>enabled</i> (default value). ■ <i>disabled</i>. <p>For security reasons this option should always be set to enabled.</p>
RadiusSrvDialout	<p>This option provides the means for RADIUS dial-out configuration.</p> <p>Possible entries:</p> <ul style="list-style-type: none"> ■ <i>disabled</i> (default value). ■ <i>enabled</i>: Enables initial loading of dial-out routes after reboot. ■ <i>reload</i>: Reload of dial-out routes. <p>For further information about RADIUS for Dial-Out, see chapter 3.6, page 87).</p>

Variable	Meaning
RadiusSrvDefaultPW	<p>Is not required with certain RADIUS implementations, such as Merit. Here you should consult the documentation for your RADIUS server.</p> <p>This is the default USER-PASSWORD the BRICK sends where no password is available (for example, in requests for the calling number or boot requests). Some RADIUS servers rely on a configured USER or CHAP-PASSWORD for any RADIUS request. The default value is an empty string.</p>

Table 3-4: RadiusServerTable

biboPPPPProfileTable For incoming calls there are some options that can not be set user specific. They have an effect on the PPP negotiation and RADIUS server usage before

the caller can be identified by username and password. These settings are entered in the **biboPPPProfileTable**:

Variable	Meaning
biboPPPProfileName	The name of the PPP profile.
biboPPPProfileAuth-Protocol	The type of authentication used on the point-to-point link as described in RFC 1334.
biboPPPProfileAuth-Radius	<p>This entry is used to configure RADIUS authentication on incoming calls.</p> <p>Possible values:</p> <ul style="list-style-type: none"> ■ <i>none</i>: RADIUS requests are not sent to the specified RADIUS server. ■ <i>inband</i> (default value): Only inband RADIUS requests (PAP, CHAP) are sent to the specified RADIUS server. ■ <i>outband</i>: Only outband RADIUS requests are sent to the server. ■ <i>both</i>: Both requests are sent to the RADIUS server (first outband request, then inband request if necessary).
biboPPPProfileLQ-Monitoring	This parameter enables or disables PPP Link Quality Monitoring (LQM) according to RFC 1989. Only relevant for inband authentication.

Table 3-5: **biboPPPProfileTable**

3.3 Configuration on the RADIUS Server

RADIUS server files When configuring a RADIUS server, different files have to be edited:



The files described in the following table are available when using a RADIUS server under Unix.

Using a RADIUS server under Windows, the configuration takes place in another way, but it is the same principle.

File	default location	Remarks
radiusd	/etc/raddb/	The RADIUS daemon on UNIX systems.
dictionary	/etc/raddb/	The dictionary file lists the RADIUS attributes the daemon process supports and defines each attribute's default behavior.
clients	/etc/raddb/	The clients file defines the list of hosts that are allowed to request authentication information from the server. Each entry typically contains the RADIUS client's host name and password, (also called the Client-Key).
users	/etc/raddb/	The users file contains user-authentication information for (dial-in) hosts that will be establishing connections via the RADIUS clients. The file consists of user-profiles (also referred to as authentication-lines) that: <ol style="list-style-type: none"> 1. define requirements for authenticating callers (password, PPP ID, Calling Line) and, 2. define the type of connections to establish if the user has been successfully authenticated.
logfile	/etc/raddb/	The logfile contains error messages from the radiusd process on Unix hosts.
detail	/usr/adm/radacct/ <client>/	The detail file contains RADIUS accounting information records submitted by RADIUS clients. <client> in the pathname to this file is usually the host name of the RADIUS client.

Table 3-6: RADIUS server files

Configuration steps on the RADIUS server

The following steps have to be performed:

- The dictionary file has to be imported. It is available from BinTec's WWW server at <http://www.bintec.de> (Section: Download). To reach the section Download, click Solutions & Products.

A list of tested RADIUS servers (eventually with an adapted dictionary file) is also available from BinTec's WWW server at <http://www.bintec.de/de/prod/index.html> (Section: Lösungen für unterschiedliche Unternehmensgrößen).

For further information concerning the syntax of dictionary files of definite RADIUS servers, see [chapter 3.4.3, page 83](#).

- The **BRICK** has to be entered as NAS (Network Access Server) server and the shared secret has to be entered (Clients file under Unix).
- The correct port has to be entered (as RADIUS servers use different port numbers, you should refer to the documentation for the RADIUS server you are using).
- The users have to be entered (users file under Unix). Here you can define for each user:
 - authentication information (username, password)
 - configuration information which is transferred from the RADIUS server to the RADIUS client (e. g. IP address, callback, access lists, etc.)Therefore, you use the standard attributes supported by BinTec's RADIUS implementation and BinTec extensions (see [chapter 3.4, page 76](#)).



If you use a RADIUS server for accounting, be sure to have a strategy for packing, moving and accounting files!

3.4 Authentication

To use the RADIUS server for the purpose of authentication, you can define several attributes for each user. The RADIUS server transfers this configuration information to the RADIUS client when accepting the authentication.

In the tables below, all supported standard RADIUS attributes (see [chapter 3.4.1, page 76](#)) and BinTec extensions (see [chapter 3.4.2, page 81](#)) are listed.

The values of the attributes can have the following types:

Value	Meaning
string	0 - 253 octets
integer	32-bit value in big endian order (high byte first)
ipaddr	4 octets in network byte order

Table 3-7: Values for Type

3.4.1 List of Standard Attributes Supported

Your router supports the following standard RADIUS attributes. Also a couple of BinTec-specific options have been added to facilitate using your router in conjunction with RADIUS servers.



Note, however, that the BinTec-specific options are only available if you use the dictionary file (available from BinTec's WWW server).

RADIUS attribute	No.	Type	Corresponding MIB variable	R / A	Remarks
User-Name	1	string	biboPPPAuthIdent or biboDialNumber	REQ	User name, mandatory. Values: <ul style="list-style-type: none"> ■ inband: PPP partner name. ■ outband: PPP partner telephone number. If outband authentication (CLID) is requested, configuration in pppProfileTable has to be done (see chapter 3.7.2, page 96).
User-Password	2	string		REQ	Password for PAP authentication. In case of outband authentication, a password is not available. If your RADIUS server requires a password, set RadiusSrvDefaultPW in RadiusServerTable .
CHAP-Password	3	string		REQ	Password for CHAP authentication.
NAS-Port	5	integer		ANS	Corresponds to the ISDN stack used for the connection.
Service-Type	6	integer		ANS	Values: <ul style="list-style-type: none"> ■ for PPP: Framed (2). ■ for PPP callback (CBCP) or Microsoft callback: Call-back-Framed (4).
Framed-Protocol	7	integer		ANS	Modifications only take affect in case of outband authentication (CLID) or in case of RADIUS used for dial-out. Values: see table 3-9, page 80 .

RADIUS attribute	No.	Type	Corresponding MIB variable	R / A	Remarks
Framed-IP-Address	8	ipaddr	biboPPPIpAddress	ANS	Partner IP address. Note: With Framed-IP-Address = 255.255.255.254 an IP address from an IP address pool on the BRICK is assigned (dynamic server mode).
Framed-IP-Netmask	9	ipaddr		ANS	Partner IP netmask.
Framed-Routing	10	integer	ipExtIfRipSend	ANS	Defines which entries in the ipExtIfTable are set. Values: <ul style="list-style-type: none"> ■ None (0): No entry. ■ RIPv1-Broadcast(1): ipExtIfRipSend gets the value <i>ripV1</i>. ■ RIPv1-Listen(2): ipExtRipReceive gets the value <i>ripV1</i>. ■ RIPv1-Broadcast-Listen (3): ipExtIfRipSend gets the value <i>ripV1</i> and ipExtRipReceive gets the value <i>ripV1</i>.
Filter-Id	11	string	ipExtIfRuleIndex	ANS	ipExtIfRuleIndex is set to <Filter-Id>.
Framed-MTU	12	integer		ANS	Is replaced by MRU/MRRU.
Framed-Compression	13	integer		ANS	Compression. Values: <ul style="list-style-type: none"> ■ None (0) ■ Van-Jacobson-TCP-IP (1)

RADIUS attribute	No.	Type	Corresponding MIB variable	R / A	Remarks
Reply-Message	18	string	ifDescr	ANS	Outband: interface name (ifDescr) is set to this name, instead of using the telephone number.
Callback-Number	19	string	biboDialTable	ANS	Telephone number for callback. An entry in biboDialTable is created.
Framed-Route	22	string		ANS	You can create a routing entry (see table 3-10, page 81).
Framed-IPX-Network	23	integer		ANS	If not fffffffe: Where necessary, entries in ipxCircTable (ipxCircType can get the value <i>wanRIP</i> or <i>unnumberedRIP</i>), ripCircTable and sapCircTable are made.
Class	25	string		ANS	If returned by RADIUS server, this attribute is added to every RADIUS accounting record.
Vendor-Specific	26	string		ANS	Only for encapsulation.
Session-Timeout	27	integer		ANS	Not used!
Idle-Timeout	28	integer	biboPPPSHORTHold	ANS	Shorthand.
Called-Station-Id	30	string		REQ	Called phone number.
Calling-Station-Id	31	string		REQ	Calling phone number (is often empty for analog users).
NAS-Identifier	32	string	biboAdmSysName	REQ	System Name of the BRICK .
CHAP-Challenge	60			REQ	Necessary for CHAP.
Port-Limit	62	integer	biboPPPMaxConn	ANS	Number of B channels that are allowed for this user.

Table 3-8: Standard RADIUS attributes for authentication

Values for attribute Framed-Protocol

Value	Name	Remarks
1	PPP	inband
17825794	X25	outband
17825795	X25-PPP	
17825796	IP-LAPB	
17825798	IP-HDLC	
17825799	MPR-LAPB	
17825800	MPR-HDLC	
17825801	FRAME-RELAY	
17825802	X31-BCHAN	
17825803	X75-PPP	
17825804	X75BTX-PPP	
17825805	X25-NOSIG	
17825806	X25-PPP-OPT	

Table 3-9: Possible values for attribute Framed-Protocol

Values for attribute Framed-Route

With Framed-Route you can create a route of the format:

```
Framed-Route = <destaddr/mask> <gateway> <metric>
```

The following values are available:

Name	Remarks
destaddr/mask	Destination address with netmask (required for a dial-out request).
gateway	Gateway address (nexthop) (optional).
metric1 - 5	Sets the variables ipRouteMetric1 to ipRouteMetric5 in the ipRouteTable ; metric1 should always lie above the value for the dial-in case, i.e. the worse metric. If no metric is given, metric1 is set to 5, while metric2 to metric5 are set to 0 (optional).

Table 3-10: Possible values for attribute Framed-Route

3.4.2 List of BinTec Attributes (Extensions)

If you use the dictionary file mentioned above, you can directly access and configure specific MIB tables via RADIUS.



The syntax of these extensions can change depending on the used RADIUS server. So refer to the documentation of your RADIUS server. For an example see [chapter 3.4.3, page 83](#).

The following BinTec extensions are available at the moment:

Option	No.	Type	Corresponding MIB variable	Mode
BinTec-biboPPPTable	224	string	biboPPPTable	static
BinTec-biboDialTable	225	string	biboDialTable	dynamic
BinTec-ipExtIfTable	226	string	ipExtIfTable	static
BinTec-ipRouteTable	227	string	ipRouteTable	dynamic
BinTec-ipExtRtTable	228	string	ipExtRtTable	dynamic
BinTec-ipNatPresetTable	229	string	ipNatPresetTable	dynamic
BinTec-ipxCircTable	230	string	ipxCircTable	dynamic
BinTec-ripCircTable	231	string	ripCircTable	dynamic
BinTec-sapCircTable	232	string	sapCircTable	dynamic
BinTec-ipxStaticRoute-Table	233	string	ipxStaticRouteTable	static
BinTec-ipxStatic-ServTable	234	string	ipxStaticServTable	static

Table 3-11: BinTec RADIUS extensions

Syntax Each of these options corresponds to a MIB table. You can modify values inside the table by using a syntax similar to the SNMP client shell of your BRICK:

```
<BinTec-Option> = "<variable1>=<value1> ... <variablen>=<valuen>"
```

A few lines from a RADIUS users file might look like this:

```
Service-Type = Framed,
BinTec-biboPPPTable = "DynShorthold=50 IpAddress=static",
BinTec-ipNatPresetTable = "Protocol=tcp extport=1050
                          intport=100"
```

When using these options, please note:

- The **ifIndex** is automatically set for each table, you cannot influence it. There is, however, one exception to this rule: In the **IpExtRtTable** both the **DstIfIndex** and the **SrcIfIndex** are automatically set. You can set one of these to 0 if need be.

- The entries are not case-sensitive.
- You must not use blank spaces before or after »=« signs inside the double quotes.
- There are two different option modes, static, and dynamic.
Static options modify existing table entries while dynamic options add a new table entry. Therefore, all the variables you want to set in a dynamic option have to be included in one single line.

3.4.3 Sample Modification for Merit RADIUS Servers

Merit Here we will give you an example of what the dictionary file on a Merit RADIUS server can look like (Merit 3.6 and later).

The syntax is as follows:

```
<vendor-name>.<vendor-string> <vendor-specific-value> <attribute-  
number> <attribute-type> <expression>
```

The dictionary file looks like this:

Dictionary file	BinTec.attr BinTec-biboPPPTable	224	string(*, 0, NOENCAPS)
	BinTec.attr BinTec-biboDialTable	225	string(*, 0, NOENCAPS)
	BinTec.attr BinTec-ipExtIfTable	226	string(*, 0, NOENCAPS)
	BinTec.attr BinTec-ipRouteTable	227	string(*, 0, NOENCAPS)
	BinTec.attr BinTec-ipExtRtTable	228	string(*, 0, NOENCAPS)
	BinTec.attr BinTec-ipNatPresetTable	229	string(*, 0, NOENCAPS)
	BinTec.attr BinTec-ipxCircTable	230	string(*, 0, NOENCAPS)
	BinTec.attr BinTec-ripCircTable	231	string(*, 0, NOENCAPS)
	BinTec.attr BinTec-sapCircTable	232	string(*, 0, NOENCAPS)
	BinTec.attr BinTec-ipxStaticRouteTable	233	string(*, 0, NOENCAPS)
	BinTec.attr BinTec-ipxStaticServTable	234	string(*, 0, NOENCAPS)

3.5 Accounting

When you configure a RADIUS server for the purpose of accounting, the **BRICK** transmits Start and Stop RADIUS packets for each ISDN connection to this server.

The values of the attributes can have the following types:

Value	Meaning
string	0 - 253 octets
integer	32-bit value in big endian order (high byte first)
ipaddr	4 octets in network byte order

Table 3-12: Values for Type

3.5.1 List of Sent Attributes Supported

The following attributes are available for accounting:

RADIUS attribute	No.	Type	Corresponding MIB variable	Start	Stop	Remarks
User-Name	1	string	biboPPPAuthIdent or biboDialNumber	X	X	User name.
NAS-Port	5	integer	isdnCallStkNumber	X	X	Corresponds to the ISDN stack used for the connection.
Service-Type	6	integer		X	X	The value is always Framed (2).
Framed-Protocol	7	integer		X	X	The used encapsulation is always specified as PPP (1).
Framed-IP-Address	8	ipaddr		X	X	If the RADIUS server sent this attribute when authenticating.

RADIUS attribute	No.	Type	Corresponding MIB variable	Start	Stop	Remarks
Class	25	string		X	X	Allows the adjustment of accounting and authentication. If this attribute is sent back from the RADIUS server during authentication, it is inserted to accounting records (depends on the RADIUS server used).
NAS-Identifier	32	string	biboAdmSysName	X	X	Name of the BRICK .
Acct-Status-Type	40	string		X	X	Possible values: Start, Stop.
Acct-Delay-Time	41	integer		X	X	Time offset in seconds from establishing the connection and sending the accounting record.
Acct-Input-Octets	42	integer			X	Received bytes.
Acct-Output-Octets	43	integer			X	Sent bytes.
Acct-Session-Id	44	string		X	X	Common index for multi-link connections, e. g. a2000002.
Acct-Session-Time	46	integer			X	Duration of session in seconds.
Acct-Input-Packets	47	integer	isdnCallReceived-Packets		X	Received packets.
Acct-Output-Packets	48	integer	isdnCallTransmit-Packets		X	Sent packets.

RADIUS attribute	No.	Type	Corresponding MIB variable	Start	Stop	Remarks
Acct-Multi-Session-Id	50	string		X	X	Unambiguous name of the session, e. g. a2000002 (with every call the last 6 digits are incremented).
Acct-Link-Count	51	integer	biboPPPCConnActive	X	X	Number of B channels, that are established for the connection at the moment.
Acct-Charge	59	integer	isdnCallCharge		X	Charging units. If this information is not sent as units but as currency amounts (e. g. 0.12 DM), the values are converted to digits (e. g. 120).

Table 3-13: Standard RADIUS attributes for accounting

3.6 RADIUS for Dial-Out

As the name suggests (Remote Authentication Dial-In User Service), RADIUS was designed as a client-server system for authenticating dial-in connections. The **BRICK** can be configured to operate as a RADIUS client that consults the RADIUS server at connection time for the authentication and identification of specified dial-in partners.

With BinTec's RADIUS implementation it is possible, however, for the **BRICK** to request user data from the server in order to establish an PPP connection also for outgoing calls.

Why RADIUS for dial-out

The principal objectives that lay behind the implementation of RADIUS for dial-out are two-fold:

- Firstly, in view of the fact that at most 500 WAN partners can be configured on the **BRICK** and some installations can greatly exceed this figure, this feature provides an alternative to configuring WAN partners on the router. The entries for WAN partners are no longer made locally on the **BRICK** via Setup Tool, but now on the RADIUS server. There can thus be considerable savings in terms of Flash memory.
- Secondly, RADIUS for dial-out is easier to manage in terms of configuration. The many entries over Setup Tool are replaced by the more convenient administration of the RADIUS server over the usual editor tools.

How does it work?

The **BRICK** firstly requests all the routing information contained on the RADIUS server and stores it in the **ipRouteTable**. Loading of this initial information is driven over the **RadiusSrvDialout** variable. On the one hand, the variable can be set to *enabled* and then saved with the configuration so that initial loading occurs immediately after every reboot. Alternatively, by setting to *reload*, it is possible to load or reload the routing information at any time you choose.

When a dial-out call to a WAN partner is to be made on one of these loaded routes, another request is sent to the RADIUS server in order to receive the

necessary partner-specific information (e. g. data for the authentication, encapsulation, extension number etc.), each partner can have more than just one entry in the **ipRouteTable**. If the partner is configured on the RADIUS server, the necessary information entries are transferred to the **BRICK** and generated in the respective MIB tables for the duration of the call.

After the end of the call, all entries on the **BRICK**'s MIB tables are deleted with the exception of the routing information in the **ipRouteTable**, which is loaded initially.

Configuration Configuration of RADIUS for dial-out takes place on two levels (similar to configuration for RADIUS for dial-in):

- Configuration on the **BRICK** side: entries are made over the **RadiusServerTable** (see [chapter 3.6.1, page 88](#)).
- Configuration on the RADIUS server: configuration on the users file and dictionary file on the RADIUS server (see [chapter 3.6.2, page 88](#)).

3.6.1 Configuration on the **BRICK**

Configuration on the **BRICK** is made over the **RadiusServerTable**.

The required MIB variables are described in [chapter 3.2.2, page 69](#).

3.6.2 Configuration on the RADIUS Server

The following description is taken from a Unix RADIUS implementation, e. g. Merit.

The configuration of the RADIUS server deals with

- telling the users file on the RADIUS server
 - the routing information required for the WAN partner (see [IP routing information, page 90](#)).
 - the partner-specific information assigned to each routing entry (see [Partner-specific information, page 94](#)).

- making sure, that the BinTec-specific extensions are included in the dictionary file of the server.



A significant advantage of the implementation from BinTec Communications AG is that only one entry in the users file is sufficient to enable both dial-in as well as dial-out.

IP routing information

Syntax This part deals with defining the necessary IP routing information. Here the following syntax should be obeyed:

```
Framed-Route = <destaddr/mask> <gateway> <userid> <userpw>
<private> <metric>
```

Routing info	Meaning
destaddr/mask	Destination address with netmask (required for a dial-out request).
gateway	Gateway address (nexthop) (optional).
userid	For RADIUS for dial-out only. The partner's user ID, necessary for a dial-out request.
userpw	For RADIUS for dial-out only. This entry must match the password attribute in the users file, see chapter 3.7.7, page 100 . You need only make the one entry for both dial-in and dial-out. It may not consist only of digits.
private	For RADIUS for dial-out only. Selection of the routing protocols, RIP, OSPF, or the PROXYARP used for the propagation of this IP route.
metric1 - 5	Sets the variables ipRouteMetric1 to ipRouteMetric5 in the ipRouteTable ; metric1 should always lie above the value for the dial-in case, i.e. the worse metric. If no metric is given, metric1 is set to 5, while metric2 to metric5 are set to 0 (optional).

Table 3-14: Possible values for attribute Framed-Route

Minimum entries If only dial-out (without callback) is being configured, destaddr and userid are the minimum entries required.

User dialout-X Several of these routes are then compiled and arranged under a fictitious user "dialout-X", which begins with the number 1. The number of entries under one

of these dummy-users is restricted to the UDP limit of 4096 bytes. Whereby the optimum numbers in terms of loading times and system utilization lie at around 20-40 entries inside the "Framed-Route" record. On initial loading, the BRICK asks for a user by the name of dialout-1, then dialout-2 and so on. Here is an example of what a dummy user could look like:

```
dialout-1
  Framed-Route = "1.2.1.1 user1 secret1 3",
  Framed-Route = "1.2.1.2 user2 secret2 3",
  Framed-Route = "1.2.2.0/24 network1 secret3 3",
  Framed-Route = "1.2.1.3 user3 secret OSPF 5",
  Framed-Route = "1.2.1.4 user4 secret OSPF 5",
  Framed-Route = "1.2.1.5 user5 more_secret RIP 5",
  Framed-Route = "1.2.1.6 user6 secret6 RIP6",
  Framed-Route = "1.2.1.7 user7 secret7 RIP 7",
  Framed-Route = "1.2.1.8 user8 secret8 RIP8",
  Framed-Route = "1.2.1.9 user9 secret9 RIP9",
  Framed-Route = "1.3.1.0/24 network10 passwdnetwork10 10",
  Framed-Route = "1.3.2.0/24 network11 passwdnetwork11 11",
  Framed-Route = "1.3.3.0/24 network12 passwdnetwork12 12",
  Framed-Route = "1.3.4.0/24 network13 passwdnetwork13 13",
  Framed-Route = "1.3.5.0/24 network14 passwdnetwork14 14",
  Framed-Route = "1.4.6.0/24 network15 passwdnetwork15 OSPF 15",
  Framed-Route = "1.4.2.0/24 network16 passwdnetwork16 OSPF 16",
  Framed-Route = "1.4.2.0/24 network17 passwdnetwork17 OSPF 17",
  Framed-Route = "1.4.2.0/24 network18 passwdnetwork18 18",
  Framed-Route = "1.4.2.0/24 network19 passwdnetwork19 RIP 19",
  Framed-Route = "1.5.1.0/24 network20 passwdnetwork20 RIP 20",

dialout-2
  .....
  .....

dialout-3
  .....
  .....
```

Specifying the BRICK to which the IP routing information should go

In the event that you have more than just one BRICK to which your IP routing information is to be transferred, it is possible to differentiate between the routers using the following syntax for the aforementioned dummy user, the following **sysName** variable is from the MIB II table **system**:

```
dialout-[sysName]-x
dialout-brick1-1
.....
.....
dialout-brick1-2
.....
.....
dialout-brick1-3
.....
.....
dialout-brick2-1
.....
.....
dialout-brick2-1
.....
.....
dialout-brick2-3
.....
.....
```

When the IP routing information is loaded to the BRICK (usually on booting when **RadiusSrvDialout** is set to *enabled*), the information is stored in the **ipRouteTable**. The indices for the as yet unused interfaces for these route entries extend from 30000. The **ipRouteTable** could look something like this:

inx	Dest(*rw)	Ifindex(rw)	Metric1(rw)	Metric2(rw)
	Metric3(rw)	Metric4(rw)	NextHop(rw)	Type(-rw)
	Proto(ro)	Age(rw)	Mask(rw)	Metric5(rw)
	Info			
03	1.2.1.1	30001	3	0
	0	1	0.0.0.0	indirect
	other	1538	255.255.255.25	0
	.0.0			
04	1.2.1.2	30002	3	0
	0	3	0.0.0.0	indirect
	other	1540	255.255.255.255	0
	.0.0			
05	1.2.2.0	30003	3	0
	0	3	0.0.0.0	indirect
	other	1540	255.255.255.255	0
	.0.0			

Example: Propagating dial-out IP routes via RIP

Here an example for propagating dial-out IP routes via RIP:

```
dialout-1
    Framed-Route = destaddr/mask userid userpw RIP
```

For settings made in the **ipExtIfTable** on the BRICK and derived from the Bin-Tec-specific Radius attributes, the variable **RouteAnnounce** is rendered ineffectual.

In principle, this is possible but not advisable if there is a large number of IP routes.

Example: Propagating dial-out IP routes via OSPF

Here an example for propagating dial-out IP routes via OSPF:

```
dialout-1
    Framed-Route = destaddr/mask userid userpw OSPF
```

For settings made in the **ipExtIfTable** on the BRICK and derived from the Bin-Tec-specific Radius attributes, the variables **Ospf**, **RouteAnnounce** and **OspfMetric** are rendered ineffectual. The dial-out IP routes are thus propagated with an OSPF metric calculated as follows:

IpMetric + 20

Example: Dial-out IP routes and Proxy-ARP

Here an example for dial-out IP routes and Proxy-Arp:

```
dialout-1
```

```
Framed-Route = destaddr/mask userid userpw PROXYARP
```

For settings made in the **ipExtIfTable** on the BRICK and derived from the BinTec-specific Radius attributes, the variable **ProxyArp** is rendered ineffectual.

Partner-specific information

Now it is necessary to assign specific details about the partner to each IP route entry, again in the users file of the RADIUS server. Here it is possible that several routes refer to just the one user entry. The minimum configuration entries must include the following:

Attribute	Meaning
Service-Type = Framed	This is the default value for PPP connections.
Framed-Protocol = PPP	This is the type of encapsulation used. If not set PPP is used.
Framed-IP-Address = X.X.X.X	This is the IP address of the WAN partner and must correspond to the destination address in the routing entry described above in Framed-Route.
Framed-IP-Netmask = Y.Y.Y.Y	IP netmask, it could be something like <i>255.255.255.255</i> .
BinTec-biboDialTable = "direction=outgoing number=*****"	A temporary entry for dial-out, including the phone number of the WAN partner, is made in the biboDialTable .
Password	A user password that must match the password used in the Framed-Route (see userpw above).

Table 3-15: Attributes for partner-specific information (minimum configuration entries)



Caution!

It is important for security reasons to make sure that no incoming calls are authenticated over this entry.

➤ Set **direction** to *outgoing*.

The following attributes are optional:

Attribute	Meaning
Framed-MTU	Sets the ifMtu variable in the IfTable .
Framed-Compression	Sets the VJHeaderComp variable of the PPPTable if necessary.
Idle-Timeout	Sets the ShortHold variable in the PPPTable .
Port-Limit	Sets the MaxConn variable in the PPPTable .
BinTec-biboPPPTable = "biboPPPAuthentication=pap/chap/ms_chap"	The authentication protocol used for dial-out; if this is not explicitly specified, the authentication protocol CHAP is set.
BinTec-biboPPPTable = "biboPPPLocal-Ident=local_pppid"	This is the local ppp ID for authentication at the WAN partner's (optional) and the default setting.

Table 3-16: Optional attributes for partner-specific information

Example For an example of Framed-Route and corresponding partner-specific entries in the users file see [chapter 3.7.7, page 100](#).

3.7 Examples

3.7.1 Typical Dial-In (Without BinTec Attributes)

To enter a user for typical dial-in, there has to be an entry like the following in the RADIUS database (users file):

```
user Password = topsecret,  
      Service-Type = Framed,  
      Framed-Protocol = PPP,  
      Framed-IP-Address = 1.2.1.1,  
      Framed-IP-Netmask = 255.255.255.255,  
      Idle-Timeout = 25
```

3.7.2 Standard Dial-In with CLID

To identify RADIUS partners outband by their CLID (calling line identification, i.e. ISDN telephone number) there has to be an entry like the following in the RADIUS database (users file):

```
user Password = topsecret3,  
      Service-Type = Framed,  
      Framed-Protocol = PPP,  
      Framed-IP-Address = 1.2.1.1,  
      Framed-IP-Netmask = 255.255.255.255,  
      Idle-Timeout = 25,  
      BinTec-biboDialTable = "number=00815123456  
                             direction=outgoing"
```



Note that the phone number must be specified here exactly as it is signalled with the incoming call (you can see this in the **RemoteNumber** field of the **isdnCallTable**).

When a call from the number 00815123456 comes in, a new PPP entry is generated with **Encapsulation = PPP**.



Please also note that when using RADIUS inband authentication it can take up to 2 seconds to accept an incoming call if the RADIUS server is delayed inactive.

3.7.3 Callback PPP Negotiated

To configure callback PPP negotiated, there has to be an entry like the following in the RADIUS database (users file):

```
user Password = topsecret,  
    Service-Type = Callback-Framed,  
    Framed-Protocol = PPP,  
    Framed-IP-Address = 1.2.1.1,  
    Framed-IP-Netmask = 255.255.255.255,  
    Idle-Timeout = 25  
    Callback-Number = 12345
```

3.7.4 Callback (Windows Client)

There are several possibilities:

- Callback to the Windows client will take place in every case, the client has to enter the number to be called during the negotiation.

There has to be an entry like the following in the RADIUS database (users file):

```
msclient password = xy,  
    Service-Type = Callback-Framed,  
    Framed-Protocol = PPP,  
    Idle-Timeout = 600,  
    biboPPPTable = "MaxRetries=1"
```



You can handle the whole configuration even without using BinTec specific attributes at all. But we suggest using the entry `biboPPPTable = "MaxRetries=1"` for the case that the Windows user enters a wrong phone number for callback.

- Callback to the Windows client will take place in every case, the number is entered on the **BRICK**.

There are two possibilities:

There has to be an entry like the following in the RADIUS database (users file):

```
msclient password = xy,
      Service-Type = Callback-Framed,
      Framed-Protocol = PPP,
      Idle-Timeout = 600,
      Callback-Number = 12345
```

Or with BinTec attributes:

```
msclient password = xy,
      Framed-Protocol = PPP,
      Idle-Timeout = 600,
      biboPPPTable = "callback=ppp_offeredMaxRetries=1",
      biboPPPTable = "Authentication=ms_chap
                    AuthSecret=xx",
      biboDialTable = "number=12345 direction=outgoing"
```

- Callback to the Windows client is allowed but the user has the possibility to enter a number or to cancel the callback.

There has to be an entry like the following in the RADIUS database (users file):

```
msclient password = xy
      Framed-Protocol = PPP,
      Idle-Timeout = 600,
      biboPPPTable = "callback=callback_optional
                    MaxRetries=1",
      biboPPPTable = "Authentication=ms_chap
                    AuthSecret=geheim"
```

3.7.5 Callback (CLID)

RADIUS server To configure Callback (CLID), there has to be an entry like the following in the RADIUS database (users file):

```

9119732123 Service-Type = Framed,
           Framed-Protocol = PPP,
           Framed-IP-Address = 1.2.1.1,
           Framed-IP-Netmask = 255.255.255.255,
           Idle-Timeout = 25
           Reply-Message = username

```

BRICK In addition on the **BRICK**, the variable **biboPPPProfileAuthRadius** in the **biboPPPProfileTable** has to be set to *outband* (see [biboPPPProfileTable](#), page 72).

3.7.6 Working with one or more RADIUS Servers

In this example, you can see how to work with more than one RADIUS server.

Here are examples of two different entries in the **RadiusServerTable**:

inx	Protocol(*rw)	Address(rw)	Port(rw)	Secret(rw)
	Priority(rw)	Timeout(rw)	Retries(rw)	State(-rw)
	Policy(rw)	Validate(rw)	Dialout(rw)	DefaultPW(rw)
00	authentication	172.16.70.14	1645	secret
	0	1000	5	active
	authoritative	enabled	enabled	
01	authentication	172.16.70.93	1645	secret
	1	1000	5	active
	authoritative	enabled	enabled	

What happens on the BRICK? According to the example above, once a dial-out request is made that is to be sent on one of the routes loaded from the Radius server and thus occupying an **IfIndex** above 30000, the Radius server with the IP address 172.16.70.14 receives a request, as this entry has the lowest **RadiusServerPriority** setting.

Backup If this server does not reply after 5 attempts (**RadiusSrvRetries**), each after an interval (**RadiusSrvTimeout**) of 1000 seconds, the **RadiusSrvState** is set to *inactive*. The server with the next lowest priority setting, in this case the server with the IP address 172.16.70.93, then receives a request from the BRICK. If this server responds to the BRICK request, the partner-specific information for an outgoing call to a WAN partner can be loaded from the Radius server to the corresponding MIB tables on the BRICK.

3.7.7 Dial-Out

The following example shows a Framed-Route with the corresponding partner-specific entries in the users file:

```
dialout-1
  Framed-Route = "1.2.1.1 user1 topsecret3"

user1 Password = topsecret3,
  Service-Type = Framed,
  Framed-Protocol = PPP,
  Framed-IP-Address = 1.2.1.1,
  Framed-IP-Netmask = 255.255.255.255,
  Idle-Timeout = 25,
  BinTec-biboPPPTable = "biboPPPAuthentication=chap",
  BinTec-biboPPPTable = "biboPPPLocalIdent=mylocalid",
  BinTec-biboDialTable = "direction=outgoing
                        number=00815123456"
```

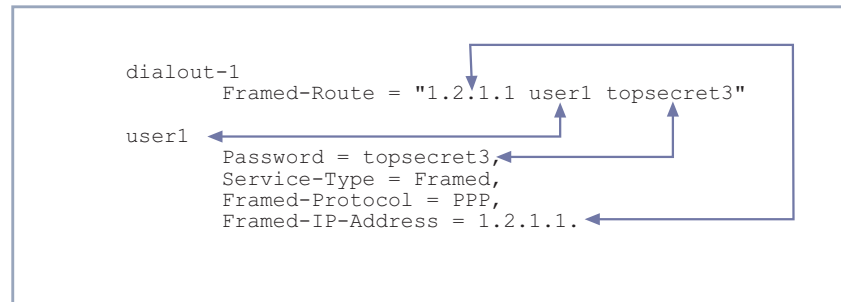


Figure 3-1: Matching Framed-Routes and partner-specific entries

For the purpose of clarification, this example places the Framed-Route together with the partner-specific information. As shown above, user name, IP address and password are identical in the routing and partner-specific information included in each example. This is essential for RADIUS for dialout to function properly.

4 Token Authentication Firewall (TAF)

In this chapter we will cover the configuration of TAF (Token Authentication Firewall).

We place emphasis on the configuration of the **BRICK** as ACE/Agent using the Setup Tool, describing the TAF client, PC configuration and all related steps in setting up TAF.

4.1 Overview

Token Authentication Firewall (TAF) is an advanced feature for controlling access to central site computing resources that goes beyond the theoretical limitations of existing security mechanisms. TAF is a user-oriented security system, which affords human interaction and by that grants that an authorized user is sitting in front of the remote host, which is connected to the central site. TAF can only be used to control IP traffic.

TAF login user verification is based on the established and well-respected Token-Card-ACE/Server solution provided by Security Dynamics.

You will need a special TAF license to use TAF on your **BRICK**. Along with this license you will get 10 TAF Login licenses for PCs you wish to use as TAF clients.

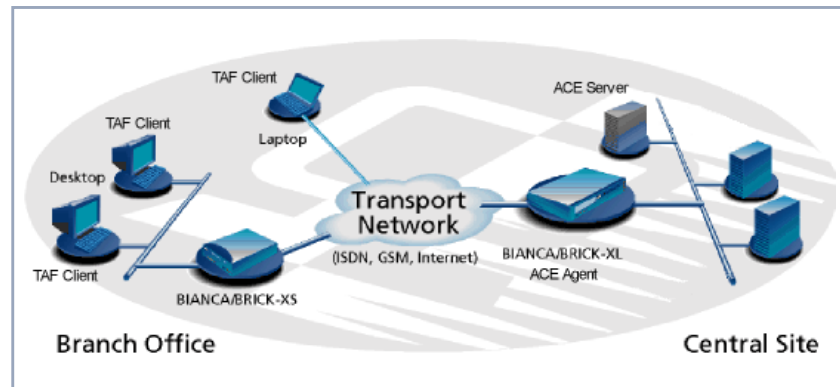


Figure 4-1: TAF Clients, ACE Agent and ACE Server

A security solution using TAF is made up of four components:

- an ACE/Agent by BinTec (BRICK-XL2, BRICK-XM with 2 MB flash or BRICK-XMP) in the central site
- an ACE/Server by Security Dynamics in the central site
- a Token Card by Security Dynamics for the user of the TAF client PC
- an application for the TAF client PC by BinTec (Windows 3.x, Windows 95/98 and Windows NT)

In this TAF security solution the **BRICK** as an ACE/Agent answers login attempts from a TAF client with a request for authentication. It then sends the user's response to the ACE/Server for verification. On the other hand, the **BRICK** verifies the authenticity of the ACE/Server so that no other server can masquerade as an ACE/Server with the intention to acquire security data. Above that the **BRICK** encrypts and decrypts messages between the TAF client and the ACE/Server.



You must bear in mind that TAF can only authenticate IP connections.

4.1.1 Requirements

As a requirement for the TAF authentication procedure, the four components (as mentioned above) must be established. Based on an existing WAN partner connection (Remote Client – LAN, LAN – LAN), the following conditions must be provided:

- In the central site LAN an ACE Server must be set up and the central site's **BRICK** must be configured as an ACE/Agent to serve as remote access server to the central site's LAN.
- The client side PC must have installed and configured the TAF login program and its user must be in possession of the Token Card, which generates one part of the password for the TAF login.



Figure 4-2: Token Card

4.1.2 Authentication

User authentication by the ACE/Server uses a “two factor” user authentication, i.e. the password consists of a static PIN, which is secret and memorized by the user and of a second part, which is generated by the user’s token card.

4.1.3 Encryption

Additionally two different encryption methods are used:

- For the communication between ACE/Server and ACE/Agent (the **BRICK** of the central site) Node Secret, a string of pseudorandom data known only to the client (ACE/Agent) and the ACE/Server, is combined with other data to encrypt client/server communications.
- For the communication between TAF client and ACE/Agent the **BRICK** generates a pair of keys (private key and public key), where the private key stays on the **BRICK** (ACE/Agent) and the public key is sent to the TAF client. By the help of these keys the transmission of authentication data is encrypted and the TAF client also uses them to check the identity of the central site.

4.2 Configuration of TAF

4.2.1 Configuring the ACE/Server

The following steps require that you have already installed an ACE/Server in your network. For instructions on how to install and configure the ACE/Server, please refer to its manuals.



Please note that the ACE/Server configuration described in this document refers to ACE/Server Version 3.01.

On the ACE/Server you first have to configure the **BRICK** to act as a gateway for the TAF-protected network, and then you have to configure each user who will be authenticated.

- Go to the **Client** menu of your Server administration tool and select **Add Client**.

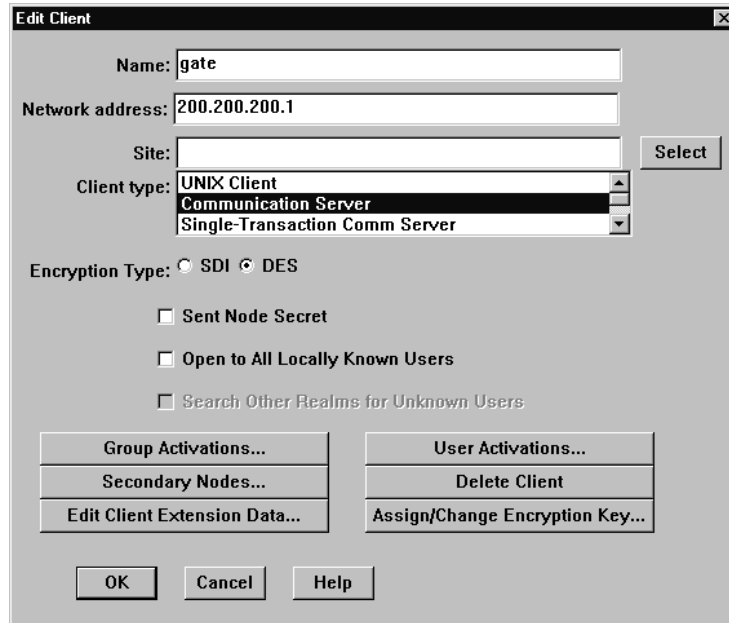


Figure 4-3: ACE/Server (Windows NT): Add Client

Now enter the name and network (IP) address of the **BRICK**, select Communication Server as the client type, and select the encryption type based on the client device configuration.



Please note that the same encryption type must also be configured on the **BRICK**.

If you want to modify ACE/Server system settings under Unix – e.g. the port to use for communication with the **BRICK** (default: 5500) – you can use the `sdsetup -config` command. In most cases no changes are necessary.

When the server receives the first authentication request from the **BRICK**, it will send a Node Secret, which is subsequently used to encode the messages exchanged between the ACE/Server and the **BRICK**.

The Sent Node Secret checkbox should not be selected. Once the Node Secret has been sent the corresponding checkbox in the dialog shown earlier will appear selected (for detailed information see “Node Secret” in [table 4-3, page 113](#)).



You can find a detailed description of this dialog box and related configuration steps in the ACE/Server Administration Manual.

- If you have not already done so, you now have to import the Token Card information into your ACE/Server (see ACE/Server Administration Manual).
- You should then enable the Token Cards, and synchronize them with the server.
- You can now start adding users (TAF clients). For each user you have to enter his first and last name, login name, whether he will be allowed or required to create his own PIN and some other items. The final step is to assign a Token Card to the user.

After you have entered all users the server configuration is complete (for TAF purposes).

As already mentioned earlier, we recommend referring to the ACE/Server’s manuals for detailed information on the configuration of the ACE/Server.

4.2.2 Configuring the **BRICK** (ACE/Agent)

In the following the TAF configuration of the **BRICK** is described in detail.

The first part introduces the Setup Tool menus dealing with TAF and in a second part the necessary configuration steps are listed.

Setup Tool Menus

- Go to **IP** ➤ **TOKEN AUTHENTICATION FIREWALL**
This menu consists of two submenus where Token Authentication Firewall relevant settings are configured.

BRICK Setup Tool	BinTec Communications AG
[IP][TAF]: Token Authentication Firewall	MyBRICK
Interfaces Server EXIT	
Press <Ctrl-n>, <Ctrl-p> to scroll through menu items, <Return> to enter	

Field	Meaning
Interfaces	used to enable/disable SecurID support separately for each BRICK interface.
Server	used for configuring SecurID Server relevant settings on the BRICK . These settings must correspond to the parameters configured on the ACE/Server.

Table 4-1: *TOKEN AUTHENTICATION FIREWALL*

Configuring Interfaces ➤ Go to *INTERFACES*.

This menu lists the **BRICK** interfaces that may be configured for Token Authentication Firewall support. TAF can only be used on interfaces which have been explicitly enabled for use with SecurID.



Typically, the SecurID Server (ACE/Server) is accessible via the **BRICK**'s LAN interface. Authentication for this interface should be set to *off*. Dial-Up interfaces used for accepting secure connections from TAF clients must be set to *SecurID*.

```

BRICK Setup Tool                               BinTec Communications AG
[IP][TAF][INTERFACES]: Interface Configuration      MyBRICK

Interfaces                Authentication
Datex-P                   off
en1                       off
en1-snap                  off
sales-ppp1                SecurID
salesppp2                 SecurID

EXIT

Press <Ctrl-n>, <Ctrl-p> to scroll, <Return> to edit/select

```

By default, Authentication is disabled (set to *off*) for existing **BRICK** interfaces.

- To enable TAF support for an interface, select the interface and press the **Enter** key. In the resulting menu ensure that Authentication is set to *SecurID* and select **SAVE**.

Configuring Interface-Specific Settings

- **EDIT**
To configure interface-specific settings for Token Authentication Firewall.

```

BRICK Setup Tool                               BinTec Communications AG
[IP][TAF][INTERFACES][EDIT]: Configure Interface sales-ppp2  MyBRICK

Authentication Type                SecurID
Life Time (seconds)                3600

Authentication Mode                 strict
Keepalives (seconds)               60

                                SAVE                CANCEL

Use <Space> to select

```

Field	Meaning
Authentication Type	This field is used to enable/disable TAF for the respective interface. By default Authentication Type is disabled (<i>off</i>). Setting to <i>SecurID</i> enables TAF for the interface.
Life Time (seconds)	The time in seconds allows data traffic on this connection. 180 seconds before the Life Time expires a new passcode is requested. Possible Values: 180 - 3600 Default Value: 3600
Authentication Mode	The authentication policy used by the ACE/Server. If set to <i>strict</i> each source IP address must be authenticated separately. If set to <i>loose</i> all source IP addresses are allowed if at least one IP address was successfully authenticated on this interface. Default value: <i>strict</i>
Keepalives (seconds)	The interval in seconds after which a new keep-alive request is sent to the BRICK by the ACE/Server. Keepalive packets will never cause a new connection to be set up, nor will they affect the shorthold mechanism.

Table 4-2: **CONFIGURE INTERFACE**

Configuring TAF Servers

■ Go to **SERVER**

This menu contains a list of the TAF servers currently configured. At the moment up to two active ACE/Servers (Master and Slave server) are supported.

By choosing **ADD** or **EDIT** you will get to the following menu, which contains the **BRICK** settings relevant to the configuration of the SecurID server (ACE/Server). The settings here must correspond to the values used by the ACE/Server.



Under Unix the parameters to use here can easily be retrieved from the ACE/Server with the included `sdinfo` program. Refer to your ACE/Server documentation for information.

BRICK Setup Tool		BinTec Communications AG
[IP][TAF][SERVER][ADD]: Configure TAF Server		MyBRICK
Type	ace	
IP Address		
Encryption	des	
Priority	0	
State	active	
Version	7	
Retries	5	
Timeout	5	
Server Port	5500	
Client Port	5656	
Node Secret	empty	
	SAVE	CANCEL
Use <Space> to select		

Field	Meaning
Type	The type of authentication server. Currently <i>ace</i> (ACE/Server) is the only type supported.
IP Address	The IP address of the authentication server.
Encryption	Specifies the type of encryption to use when communicating with the authentication server. For ACE/Servers this can currently be either <i>des</i> (Data Encryption Standard) or <i>sdi</i> (Security Dynamics proprietary) encryption. Default value is <i>des</i> .
Priority	The authentication server with the lowest priority value is the first used for requests. Use the value <i>0</i> for the master server and the value <i>1</i> for the slave server.
State	Either <i>active</i> or <i>disabled</i> .
Version	The file version number used by the authentication server. Default value is 7.
Retries	This is the number of times the BRICK will attempt to connect to the authentication server before reporting a connection failure. Valid range is 1 through 6.
Timeout	The time in seconds to wait for a reply from the authentication server before retrying. Valid range is 1 through 20. Default value is 5.
Server Port	The port number to use for communication between the BRICK and the authentication server. By default port <i>5500</i> is used.

Field	Meaning
Client Port	<p>The port number to use for communication with TAF Clients.</p> <p>Default port is 5656.</p>
Node Secret	<p>Indicates whether the Node Secret has already been received by the BRICK (<i>received</i>) or not (<i>empty</i>).</p> <p>The node secret is automatically generated by the ACE/Server and then transmitted to the BRICK. It is a password used to encode messages between the BRICK (ACE/Agent) and the ACE/Server. Usually the node secret is initially sent by the ACE/Server and after that the Sent Node Secret checkbox on the ACE/Server is automatically selected. See “Configuring the ACE/Server” in chapter 4.2.1, page 105.</p> <p>You can use RESET NODE SECRET to momentarily clear the Node Secret on the BRICK. When the Sent Node Secret checkbox on the ACE/Server is cleared, the ACE/Server will transmit a new Node Secret at the next communication.</p> <p>Whenever the BRICK receives a new Node Secret from the ACE/Server, the tafServerTable, where the Node Secret is stored, is saved to the flash ROM.</p>

TABLE 4-3: CONFIGURE TAF SERVER

TAF Commands on the BRICK

Kommando	Meaning/Tab
<code>makekey [-g]</code>	<p>The <code>makekey</code> command can be used to show the current public key (stored on the biboAdmPublicKey variable), or – when invoked with the <code>-g</code> option – to generate a new pair of keys (public and private).</p> <p>You will only need to use <code>makekey -g</code> once before configuring TAF for a WAN partner for the first time.</p>
<code>shtaf</code>	<p>The <code>shtaf</code> command can be used to test the TAF authentication procedure. The BRICK will prompt you for an ACE/Server user name and a passcode (the Token currently displayed on this user's Token Card).</p> <p>If the authentication was successful, it will give you a normal BRICK login prompt. After logging in to the BRICK you can terminate <code>shtaf</code> by typing <code>exit</code>.</p>

Table 4-4: TAF commands

Configuration of the BRICK (ACE/Agent) via Setup Tool

We will assume that your **BRICK** is up and running, and that a TAF license is available.

- Login to your **BRICK** as the admin user and start the Setup Tool (`setup`).
- Go to the **IP** ➤ **TAF** ➤ **SERVER** menu and **ADD** a new Server.
 - First you have to add a main ACE/Server.
 - Enter the ACE/Server's name or IP address and select the same encryption as configured on the Server. Make sure to use the correct (Config File) Version, Retries, and Timeout settings (you can obtain a list of the important

Server settings under Unix by issuing the `sdfinfo` command on your ACE/Server).

For normal applications it is advisable to use the default port setting (5500). The Node Secret field is filled in automatically (table 4-3, page 113).

- You can then, if necessary, add one slave server, which must be configured identically to the main server, only its Priority value must be set to 1 or higher (i.e. it gets a lower priority than the main server).

Exit the Setup Tool and execute the command `makekey -g` (table 4-4, page 114). This will generate a pair of keys (public and private) which will be used to encode the authentication messages exchanged between the **BRICK** and the user's PC.



These steps only have to be taken once.



At this point you should test your configuration by executing the `shtaf` (table 4-4, page 114) command on your **BRICK**. The **BRICK** will then contact the main ACE/Server and request you to enter a user name and passcode for authentication.

When the respective TAF client is part of a LAN, the remote **BRICK**, the gateway to the TAF client's LAN, must be configured as a WAN Partner. When you have TAF clients, which are single remote PCs (via modem or ISDN), then you have to create a WAN Partner entry for every PC that will be used to authenticate users.



For this WAN Partner only the IP protocol should be configured, because TAF can only authenticate IP packets. If you activate IPX or Bridging simultaneously, this traffic will not be verified by TAF.

- After you made sure the connection works, go to the **IP** ➤ **TAF** ➤ **INTERFACES** menu and select the interface you just created (interface name = WAN partner name).
Switch **Authentication Type** to *SecurID*. Adjust the other three parameters if necessary for your application (for an explanation of the parameters please refer to [table 4-2, page 110](#)).
- Repeat this procedure until all partners are configured.

System Logging Messages

Syslog messages are created during various events. TAF Syslog messages are reported on the **BRICK** under the INET subsystem. The following messages may be seen in connection with Token Authentication Firewall and SecurID.

Message	Meaning	Level
TAF: new session for <IP addr> ifc <ifindex>		Debug
TAF: delete session for <IP addr>		Debug
TAF: set Authlifetime to <seconds> for <IP addr> ifc <ifindex>		Debug
TAF: allow auth packet from if <ifindex> prot <protocol> <IP addr> :<port>-><IP addr> :<port>		Debug
TAF: early request for <IP addr.> ifc <ifindex>		Info
TAF: life timer expired for <IP addr.> ifc <ifindex>		Info
Taf: mibio: ACE server <IP addr.> ignored - wrong Configuration	The named server was deactivated, because its configuration was different to the configuration of the Master Server.	Err

Message	Meaning	Level
Taf: mibio: ACE server <IP addr.> ignored - too many masters	Two Master Servers have the same priority; one of them was deactivated.	Err
Taf: mibio: ACE server <IP addr.> ignored - too many slaves	Two Slave Servers have the same priority; one of them was deactivated.	Err
Taf: mibio: Saving tafServerTable to the flash ROM	The tafServerTable was automatically saved to flash ROM after the Node Secret had been transmitted. All changes, made to this table are still existent after the next reboot.	Notice
Taf: clienudp: Unable to create/bind ACE/Server socket - errno = ...		Err
Taf: clienudp: Unable to locate ACE/Server host - errno = ...	There are no servers configured in the tafServerTable .	Err
Taf: clienudp: Unable to send to the ACE/Server - errno = ...	Cannot send message to the ACE/Server; internal error.	Err
Tafd: PC Message corrupted	The message from the client was wrongly coded	Notice
Tafd: decryption error 0x<type>	The message from the client was wrongly coded	Err
Tafd: encryption error 0x<type>	The message from the client was wrongly coded	Err
Tafd: no key for encryption	You have to call <code>makekey -g</code> to generate a new key	Err
Tafd: Request for token authentication ignored - no key available	You have to call <code>makekey -g</code> to generate a new key.	Err
Tafd: TAF server unreachable	The ACE/Server is unreachable/does not answer/ is not working	Err
Tafd: No TAF License		Err

Message	Meaning	Level
Tafd: Authentication result for <IP addr> ifc <ifindex>: <result>		Info
Tafd: Tafd: received <message type> Message from <IP addr> ifc <ifindex>		Debug
Tafd: Tafd: sent <message type> Message to <IP addr> ifc <ifindex>		Debug

Table 4-5: biboAdmSyslogMessage

4.2.3 Configuring the TAF Client PC

The TAF client application is a component of BinTec's BRICKware, which can be found on the BinTec ISDN Companion CD or can be downloaded from BinTec's Web Server at <http://www.bintec.de> (Section: Download). To reach the section Download, click Solutions & Products. You can install it together with BRICKware on the TAF client PC.

- If you want to use TAF Login from a PC, you must select **TAF Login** in the Components list during the installation of BRICKware for Windows. In case you already have installed other components of BRICKware and want to add TAF Login, we recommend reinstalling all components of BRICKware (including TAF).
- The TAF Login program will automatically be installed in your **Startup** menu (you may have to select this during installation). When the TAF Login is not automatically started after the installation is complete, you must select TAF Login from the BRICKware group in the **Start** menu.
- In the Login dialog box, you must select Configuration to configure the Login program. In this dialog, you enter the **BRICK's** (ACE/Agent of the central site LAN) IP address and can modify the Listen Port if necessary (the listen port setting on the PC must be identical to the setting on the **BRICK**). Above that you must initially enter the program's license key for the TAF client, which is provided together with your **BRICK's** TAF license.

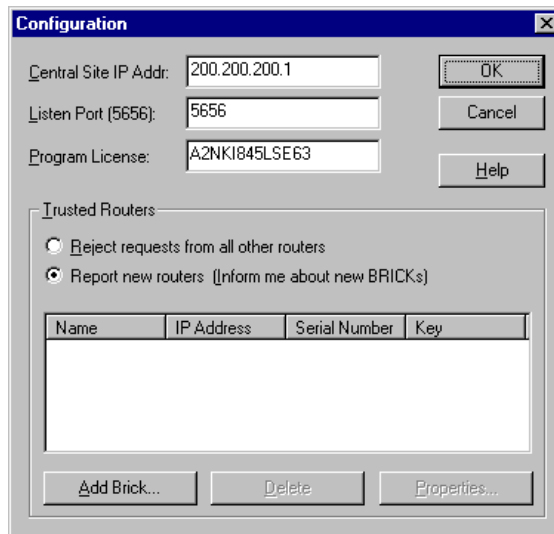


Figure 4-4: TAF Configuration

- Repeat this procedure on each PC you want to use for TAF authentication purposes. Each PC needs its own TAF client license.
- In the **Trusted Routers** group, you can select whether only to accept logins from trusted routers or also be notified when a router not contained in the trusted routers list below sends a login request. In the notification (shown below), you can then decide whether to trust the new router. Trusted routers are displayed in the list at the bottom of the Trusted Routers group.

Using TAF Login

The TAF Login program is added to the Autostart menu and will remain in the background until it receives an authentication request from the remote LAN.





Figure 4-5: Notification about the login request of a not-trusted router

- You can also activate the program by double-clicking on the TAF icon in the task bar or by starting it from the BRICKware program group to start the authentication procedure from your TAF client PC.

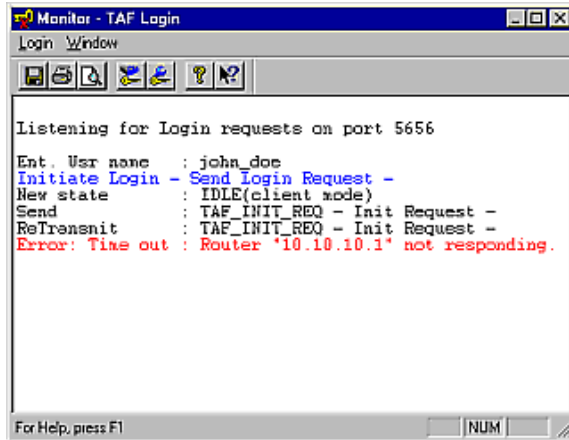


Figure 4-6: TAF Login

- Enter your login name for the ACE/Server and the passcode displayed on your Token Card. Click **OK**.

If the authentication was successful the TAF Login dialog will be closed and the TAF icon in the task bar will change to  , if the authentication failed an error message is displayed, and the icon will remain  .

- TAF Login also includes a monitoring function. If you right-click on the TAF icon, you will get a menu from which you can select **Show Monitor Window**.

The image shows a Windows-style window titled "Monitor - TAF Login". The window has a menu bar with "Login Window" and a toolbar with icons for file operations and help. The main area contains a text log with the following content:

```
Listening for Login requests on port 5656
Ent. Usr name   : john_doe
Initiate Login - Send Login Request -
New state      : IDLE(client mode)
Send           : TAF_INIT_REQ - Init Request -
ReTransmit    : TAF_INIT_REQ - Init Request -
Error! Time out : Router '10.10.10.1' not responding.
```

At the bottom of the window, there is a status bar with the text "For Help, press F1" and a "NUM" button.

Figure 4-7: TAF Monitor

All important activities concerning TAF are logged in this window. You can also initiate a login or configure the program from this window.

5 Virtual Private Networking (VPN)

In this chapter we will cover the Setup Tool menus and settings you will see while using configure the Virtual private networking support on your router.

Following that we will cover some background information relating to Virtual Private Networking technology.

Also we will introduce VPN and NAT configuration.

Then we will describe a few examples showing you how Virtual Private Networking can be used on your router.

5.1 Setup Tool Menu

After entering `setup` from the shell prompt Setup Tool's Main Menu is displayed as below. Depending on your hardware setup and software configuration your router's menu may differ slightly.

```

BRICK Setup Tool                                     BinTec Communications AG
                                                    MyBRICK

Licences      System

Slot1:        CM-BNC/TP, Ethernet
Slot2:        CM-2XBRI, ISDN S0, Unit 0
              CM-2XBRI, ISDN S0, Unit 1

Slot3:        CM-1BRI, ISDN S0

WAN Partner
IP  IPX  X.25  VPN

Configuration Management
Monitoring and Debugging
Exit

Press <Ctrl-n>, <Ctrl-p> to scroll through menu items, <Return> to
enter

```

➤ **Go to *VPN*.**

This is the point where our exploration of Setup Tool begins. The ***VPN*** menu lists the current Virtual Private Networking partner interfaces configured on the router.

BRICK Setup Tool	BinTec Communications AG	
[VPN]: Configure VPN Interfaces	MyBRICK	
Current VPN Interfaces		
Interface	Protocol	State
ADD	DELETE	EXIT

To CREATE VPN INTERFACES ➤ Go to **ADD**.
Use this menu to create Virtual Private Networking interfaces.

BRICK Setup Tool [VPN][ADD]: Configure VPN Interface	BinTec Communications AG MyBRICK
Partner Name	tunnel
Encapsulation	PPP
Compression	none
Encryption	none
PPP> Advanced Settings>	
IP > IPX >	
SAVE CANCEL	
Enter string, max length = 25 chars	

Field	Meaning
Partner Name	The partner name assigned to this virtual interface.
Encapsulation	The type of encapsulation to use; currently PPP must be used.
Encryption	Determines the type (if any) of encryption to use with this partner. Microsoft Point-to-Point Encryption (MPPE) using 40 bit or 128 bit keys are supported.

Table 5-1: **VPN** ➤ **ADD** ➤ **CONFIGURE VPN INTERFACE**

- PPP Settings** ➤ Go to **PPP**.
The VPN PPP submenu defines PPP settings for the VPN partner interface.

BRICK Setup Tool		BinTec Communications AG
[VPN][ADD][PPP]: PPP Settings ()		MyBRICK
Authentication	CHAP + PAP + MS-CHAP	
Partner PPP ID	tunnell-ppp-id	
Local PPP ID	brick	
PPP Password	tunnell-ppp-pwd	
Keepalives	off	
Link Quality Monitoring	off	
OK		CANCEL
Use <Space> to select		

Field	Meaning
Authentication	The authentication protocol to use when authenticating this partner.
Partner PPP ID	The PPP ID that the VPN partner must identify itself with during PPP negotiation.
Local PPP ID	The BRICK 's PPP ID which is used during PPP negotiation with this VPN partner.
PPP Password	The password the VPN partners must use when challenged by the BRICK during PPP negotiation.
Keepalives	This option is only relevant for leased line connections.
Link Quality Monitoring	This option allows you to tell the BRICK to gather PPP Link Quality statistics for a specific PPP partner. When enabled, link statistics are continuously written to the BRICK 's biboPPPLQMTable (viewable from the SNMP shell), when a connection is established with this partner.

Table 5-2: **VPN** ► **ADD** ► **CONFIGURE VPN INTERFACE** ► **ADD** ► **PPP**

➤ Go to *IP*.

VPN partners will have two different IP addresses that define which network the host is on:

1. The Internet
This address must be an official address and defines where the host can be reached on the Internet. For the purposes of VPN, this address must be static (it may not be dynamically assigned by an ISP).
2. The VPN
The host's IP address on the local LAN.

BRICK Setup Tool		BinTec Communications AG
[VPN][ADD][IP]: IP Configuration (vpn1)		MyBRICK
VPN Partner's IP Address	192.168.12.99	
via IP Interface	ISP	
Identification by IP Address	no	
Partner's LAN IP Address	192.168.13.99	
Partner's LAN Netmask	255.255.255.0	
Advanced Settings>		
	SAVE	CANCEL
Enter string, max length = 25 chars		

Field	Meaning
VPN Partner's IP Address	The VPN partner's IP address where the partner can be reached on the Internet.
via IP Interface	The IP interface that packets received from this VPN partner will be received on. This will typically be the interface to the Internet Service Provider.
Identification by IP Address	When set to <i>yes</i> , the VPN partner can be identified by his IP address (static).
Partner's LAN IP Address	The VPN partner's LAN address.
Partner's LAN Netmask	The netmask the partner uses on its LAN. If left blank, a standard netmask for the respective network class will be used.

Table 5-3: **VPN** ➤ **ADD** ➤ **IP** ➤ **IP CONFIGURATION (VPN1)**

VPN Interface Settings ➤ Go to **ADVANCED SETTINGS**.

BRICK Setup Tool		BinTec Communications AG
[VPN][ADD][ADVANCED]: Advanced Settings (tunnel)		MyBRICK
RIP Send		none
RIP Receive		none
Dynamic Name Server Negotiation		yes
IP Accounting		off
Back Route Verify		off
Route Announce		up or dormant
Proxy Arp		off
	OK	CANCEL
Use <Space> to select		

Field	Meaning
RIP Send/Receive	Defines the which version of RIP packets to exchange with this partner.
Dynamic Name Server Negotiation	Defines whether (and how) the name server's address is configured.
IP Accounting	Enable/disable generation of IP accounting messages for this partner. When enabled, an accounting message is generated (and written in biboAdmSyslogTable) which contains detailed information regarding connection activity for this partner.
Back Route Verify	When enabled the BRICK verifies that the return route for all packets received from this partner interface uses the same interface the packet arrived on.

Field	Meaning
Route Announce	<p>This option allows you to control when IP routes defined for this interface will be propagated. This is dependent upon the interface's ifOperStatus (in the ifTable) as follows:</p> <p>Possible values:</p> <ul style="list-style-type: none"> ■ <i>up only</i>: Routes are propagated only when the operational status of the interface is up. ■ <i>up or dormant</i>: Routes are propagated only when the operational status of the interface is up or dormant. ■ <i>always</i>: Routes are propagated always, regardless of the current link's operational status.
Proxy Arp	<p>Proxy ARP (Address Resolution Protocol) for WAN links is disabled, or <i>off</i> by default. When enabled (<i>up only</i> or <i>up or dormant</i>) requests are answered in dependence of the ifOperStatus of the link.</p>

Table 5-4: **VPN** ➤ **ADD** ➤ **IP** ➤ **ADVANCED SETTINGS**

The settings defined here are similar to the **WAN** ➤ **PARTNERS** ➤ **ADVANCED SETTINGS** menu but apply specifically to a VPN partner interface.

- IPX Settings**
- Go to **IPX**.
The **VPN** ➤ **IPX** submenu defines IPX relevant settings for VPN partner interfaces that support IPX.
 - Select **Enable IPX**.

BRICK Setup Tool		BinTec Communications AG
[VPN][ADD][IPX]: IPX Configuration (tunnel)		MyBRICK
Enable IPX	yes	
IPX NetNumber	0	
Send RIP/SAP Updates	triggered+piggyback	
Update Time	60	
OK		CANCEL
Enter hex number range 0..ffffffe		

Field	Meaning
IPX NetNumber	The IPX network number of the network link (the PPTP link). This is required by some IPX routers.
Send RIP/SAP Updates	Determines how often RIP and SAP packets are transmitted to this VPN partner. The possible options are the same as those defined in the menu, see the User's Guide for additional information.
Update Time	Determines how often (in seconds) periodic updates are sent to this VPN partner.

Table 5-5: **VPN** ➤ **ADD** ➤ **IPX**

5.2 Overview of Virtual Private Networking

5.2.1 Overview

A Virtual Private Network can be considered as a virtual Wide Area Network. It is Virtual in the sense that the network is not physical but is established on demand by software that establishes a link between a client and the server. VPNs are typically established over public (TCP/IP-based) data networks such as the Internet.

A VPN is also considered Private since user data transmitted over the link is typically encrypted. Windows 95/NT based networks achieve this security via Microsoft's own Point-to-Point Encryption protocol, or MPPE. Since these VPN connections are encrypted (user data portion) network administrators can be assured that the use of the underlying public data network does not compromise data integrity.

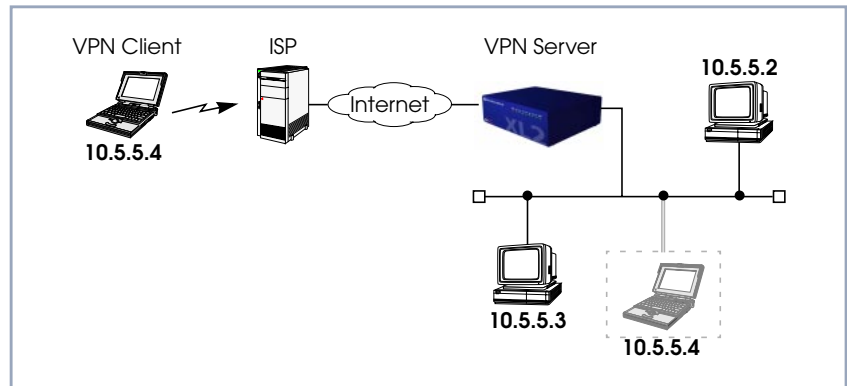


Figure 5-1: Typical VPN Scenario

PPTP The protocol that makes VPN possible is the Point-to-Point Tunneling Protocol or PPTP. PPTP is an IETF standard described in RFC 1171.

5.2.2 Tunnelling and PPTP

Simplified, tunnelling is a method of encapsulating packets of one high layer protocol within the envelope of another high layer protocol (typically IP), “IP-over-IP” if you will. This technique also allows protocol data such as IPX and NetBEUI to be tunneled via IP packets.

There are two commonly used scenarios for establishing VPN connections. The difference lies in which hosts involved in establishing the end-to-end connection support PPTP and which do not. Where PPTP support starts and stops also defines where the “tunnel” begins and ends.

Scenario 1: PPTP Client-to-VPN Server

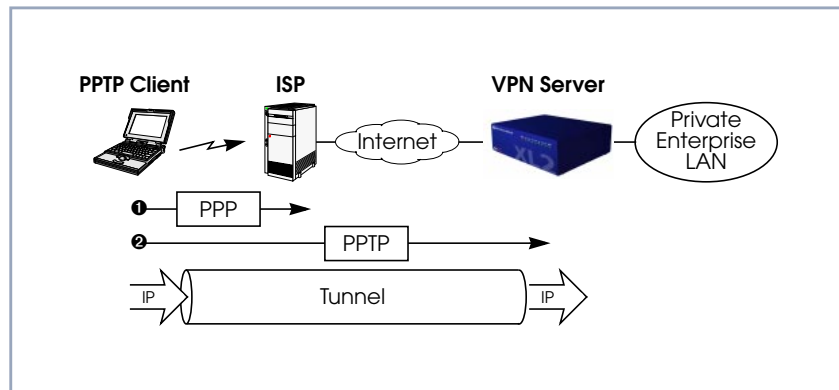
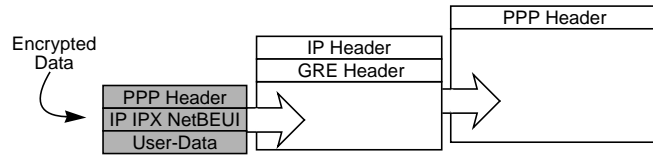


Figure 5-2: Scenario 1: PPTP Client-to-VPN Server

This is the most common scenario for PPTP. The remote client (mobile Win95 host) first establishes a standard PPP connection to a local ISP. The same client then initiates a second, logical connection, to the VPN Server. The ISP (and all intermediate Internet routers), unaware that it is participating in a VPN, simply routes IP packets from the PPTP Client.

To hosts on the Private Enterprise LAN the remote PPTP Client appears as if it were directly connected to the LAN.

When sending data to the enterprise LAN the PPTP Client encapsulates PPP packets in the user-data field of the IP packet which is later unpacked by the VPN Server.



In the diagram above, GRE refers to the Generic Routing Encapsulation protocol. The GRE header identifies PPTP relevant functions and allows for efficient use of the link.

**Scenario 2:
LAN-to-LAN VPN**

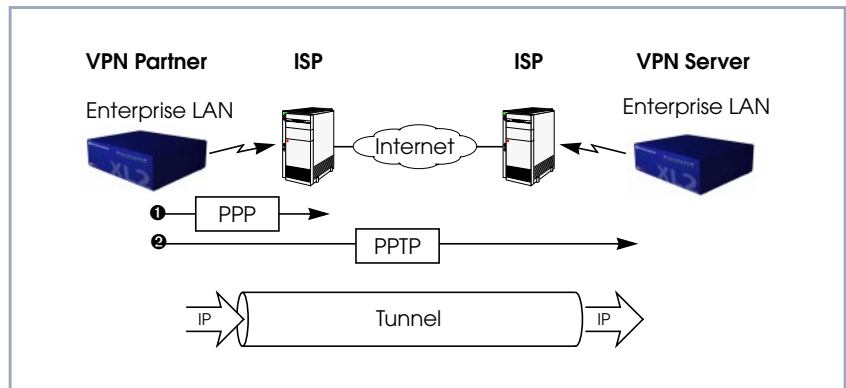


Figure 5-3: Scenario 2: LAN-to-LAN VPN

Here a Virtual Private Network that connects two enterprise LANs via the Internet is established via two VPN Servers. Either side may initiate a standard PPP link to a local ISP. Once the link is established the same server establishes a PPTP connection to the remote VPN server. Again, the ISP is unaware of its participation in the VPN.

All traffic routed via the ISP and destined for the remote LAN is encapsulated/unpacked by the respective VPN servers as mentioned in scenario 1.

5.2.3 Authentication – Encryption – Compression

In both scenarios above a second PPTP connection is established over an existing link. This second connection has its own PPP parameters (unique from

those of the underlying link) with respect to user authentication, encryption, and compression.

Authentication Both the ISP and the VPN Server will typically want to verify the initiating partner during connection establishment. Authentication is performed inband using PAP, CHAP, or MS-CHAP.

Data Encryption Data encryption allows you to be sure that all user data transmitted over public data networks via a VPN is secure. The **BRICK** supports Microsoft's Point-to-Point Encryption protocol, or MPPE. Data encryption/decryption is performed at each end of the tunnel. Each host separately generates a session-key (40 or 128 bit key) using the respective partner's PPP password which is known to each host ahead of time.



Since session-key generation is based upon the partner's password, data encryption is only possible if authentication (PAP, CHAP, or MS-CHAP) is enabled. Also, for 128 bit encryption the MS-CHAP authentication protocol is required (i.e., must be successfully negotiated at connect time.)

The Windows PPTP configuration dialog includes an option for password encryption. This option applies to transmittal of the PPP password and does not apply to data encryption.

Compression Data compression, depending on the data and the compression algorithm used, can increase performance over dial-up links as much as 30 fold (best case scenario using Stacker LZS). In both scenarios shown above, compression can be enabled for the initial PPP connection. Compression can also be enabled for PPTP links between **BRICKs** (see [Scenario 2: LAN-to-LAN VPN, page 135](#)).



The following limitation currently exists when combining compression and encryption for a PPTP link with Windows based hosts.

When the Enable software compression option is enabled in the **Server Types** tab, Windows PPTP Clients offer either MS-STAC Compression or MPP Encryption when tunnel parameters are negotiated. Currently, compression is only possible for the PPTP link if Encryption is set to *none* for the VPN partner interface on the **BRICK** (see [TO CREATE VPN INTERFACES, page 125](#)).

5.3 VPN and NAT

If the client wants to use the connection to the ISP not only for establishing a VPN connection to the headquarters, but also for using other services of the Internet, NAT (Network Address Translation) has to be activated on the client **BRICK**. Then all client PCs connected to the client **BRICK** appear in the Internet with the same IP address.

BinTec's NAT implementation supports connections with the protocols ICMP, TCP, UDP and GRE (Generic Routing Encapsulation). All VPN connections using the protocol GRE for transport, e.g. PPTP and L2TP (Layer 2 Tunneling Protocol) can be switched through a BinTec router when NAT is activated on the WAN interface.

BinTec's NAT implementation allows forwarding GRE packets to a specified endpoint. So it is possible to establish VPN connections also with NAT.

5.3.1 Constellation

The following scenario displays a LAN–LAN connection using a VPN tunnel between two partners (*CentralSite* and *SupplierNet*):

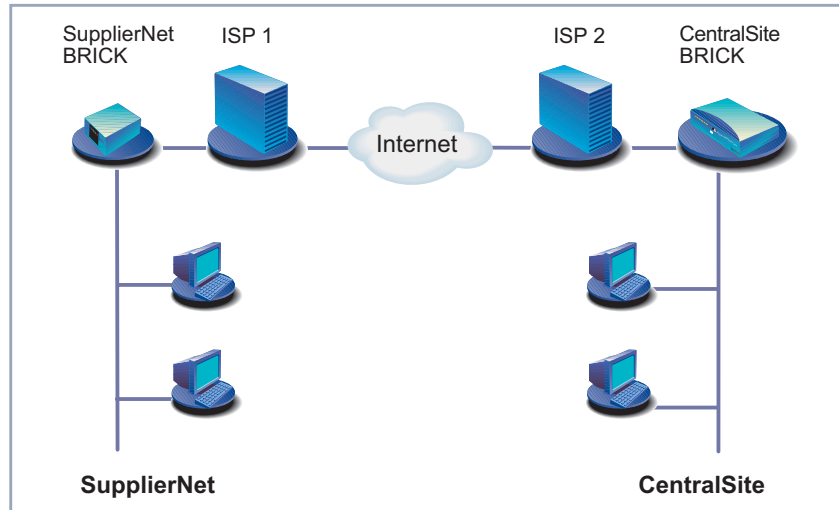


Figure 5-4: LAN–LAN connection using a VPN tunnel

Establishing and using a VPN connection with PPTP (see [figure 5-5, page 139](#)) requires two protocols between the two tunnel endpoints – TCP (over PPTP) and GRE (over IP):

1. A TCP connection to establish the tunnel (PPTP call control):
In our example *SupplierNet* opens the TCP connection to *CentralSite* (destination port 1723) to establish a PPTP connection.
2. A GRE session to use the tunnel as a transport medium:
After establishing the tunnel, *SupplierNet* can use the GRE session to exchange data with the *CentralSite* (PPP packets are encapsulated by GRE headers). If NAT is activated on the VPN interface of *SupplierNet*, the connection can not be realized because GRE packets from *CentralSite* can not be switched through the NAT firewall of *SupplierNet*. So the VPN connection fails.

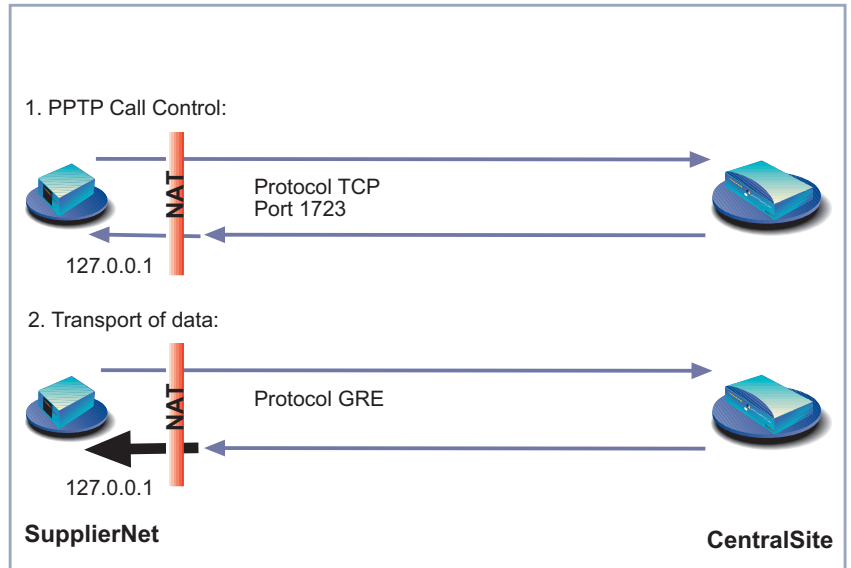


Figure 5-5: Establishing and using a VPN tunnel with PPTP

5.3.2 NAT on VPN Configuration

The configuration of NAT on a VPN interface can be realized by means of:

- Setup Tool
- MIB Variables

Setup Tool

The range of values **Protocol** can receive in the menu **IP ► NETWORK ADDRESS TRANSLATION ► RETURN ► ADD** has been extended with the value *gre* as protocol ID for the protocol GRE. In addition, the values *ah*, *esp* and *l2tp* have been introduced as protocol IDs for Authentication Header, Encapsulated Security Payload and Layer Two Tunneling Protocol respectively.

BRICK Setup Tool		BinTec Communications AG
[IP][NAT][CONFIG][EDIT]:NAT Configuration (headquarters)		MyBRICK
Service	user defined	
Protocol	gre	
Port (-1 for any)	-1	
Destination	127.0.0.1	
SAVE	CANCEL	
Use <Space> to select		

Now the following values are available for **Protocol**:

Field	Meaning
Protocol	Protocol to allow. Possible values: <i>icmp, tcp, udp, esp, ah, l2tp, gre.</i>

Table 5-6: **IP** ► **NETWORK ADDRESS TRANSLATION** ► **RETURN** ► **ADD**

Client with dynamic IP address assignment

If the client, i.e. **SupplierNet**, gets its IP address dynamically assigned by its ISP, the establishing of the VPN connection can only be done by **SupplierNet**, not by **CentralSite**.



CentralSite has to have a fixed IP address access to the Internet to enable **SupplierNet** to establish a VPN connection to the **CentralSite**.

To configure the client **BRICK**, proceed as follows:

- Go to **IP** ► **NETWORK ADDRESS TRANSLATION**.
- Select the VPN interface to be configured for NAT and press **Return**.
- Select **Network Address Translation: on**.

- Press **ADD**.
- Select **Service**: *user defined*.
- Select **Protocol**: *gre*.
- Enter **Port (-1 for any)**: *-1*.
- Enter **Destination**: *127.0.0.1*.
- Press **SAVE**.

Client with static IP address

If the client, i.e. **SupplierNet**, has a static IP address, the VPN connection can be established by both sites, **SupplierNet** or **CentralSite**.



Both **CentralSite** and **SupplierNet** have to have a fixed IP address access to the Internet to enable VPN connections to be established in both directions.

To configure the client **BRICK**, proceed as follows:

- Go to **IP** ➤ **NETWORK ADDRESS TRANSLATION**.
- Select the VPN interface to be configured for NAT and press **Return**.
- Select **Network Address Translation**: *on*.
- Press **ADD**.
- Select **Service**: *user defined*.
- Select **Protocol**: *gre*.
- Enter **Port (-1 for any)**: *-1*.
- Enter **Destination**: *127.0.0.1*.
- Press **SAVE**.
- Press **ADD**.
- Select **Service**: *user defined*.
- Select **Protocol**: *tcp*.
- Enter **Port (-1 for any)**: *1723*.

- Enter **Destination**: *127.0.0.1*.
- Press **SAVE**.

Central Site with static IP address

To configure the **CentralSite BRICK**, proceed as follows:

- Go to **IP** ➤ **NETWORK ADDRESS TRANSLATION**.
- Select the VPN interface to be configured for NAT and press **Return**.
- Select **Network Address Translation**: *on*.
- Press **ADD**.
- Select **Service**: *user defined*.
- Select **Protocol**: *gre*.
- Enter **Port (-1 for any)**: *-1*.
- Enter **Destination**: *127.0.0.1*.
- Press **SAVE**.
- Press **ADD**.
- Select **Service**: *user defined*.
- Select **Protocol**: *tcp*.
- Enter **Port (-1 for any)**: *1723*.
- Enter **Destination**: *127.0.0.1*.
- Press **SAVE**.



127.0.0.1 is the loopback address. It is entered as **IntAddr** or as **Destination** because the **BRICK** itself is an endpoint of the VPN tunnel.



When configuring a VPN connection over a dialup connection, it is recommended to set Short Hold for the VPN connection with a shorter time interval than the Short Hold for the underlying dial-up connection. Otherwise, unnecessary connections could be established because of termination of the VPN connection.

MIB Variables

The range of values of the MIB variables **ipNatProtocol** and **ipNatPrProtocol** in the **IpNatTable** and **IpNatPresetTable** respectively has been extended with the value *gre* as protocol ID for GRE. In addition, the values *ah*, *esp* and *l2tp* have been introduced as protocol IDs for Authentication Header, Encapsulated Security Payload and Layer Two Tunneling Protocol respectively.

The following values are available now:

Variable	Meaning
ipNatProtocol	Specifies the protocol the session is using. Possible values: <i>udp, tcp, icmp, ospf, esp, ah, l2tp, gre.</i>

Table 5-7: **IpNatTable**

Variable	Meaning
ipNatPrProtocol	Specifies the protocol for which the table entry shall be valid. Possible values: <i>udp, tcp, icmp, ospf, delete, esp, ah, l2tp, gre.</i>

Table 5-8: **IpNatPresetTable**

5.4 Virtual Private Networking Examples

5.4.1 Example Client-to-LAN Configuration

The Virtual Private Network shown in [Scenario 1: PPTP Client-to-VPN Server, page 134](#) would be configured as follows.

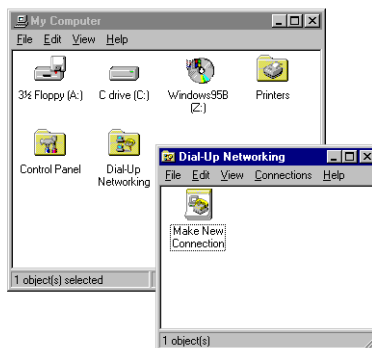
Configure PPTP Client

Requirements: VPN Partners must support the PPTP protocol. For Windows 95 hosts this involves installing Winsock and Dial-Up Networking 1.2 Updates. Software updates and configuration information can be retrieved via Microsoft's web site at:

<http://www.microsoft.com/communications/pptpdownnow.htm>

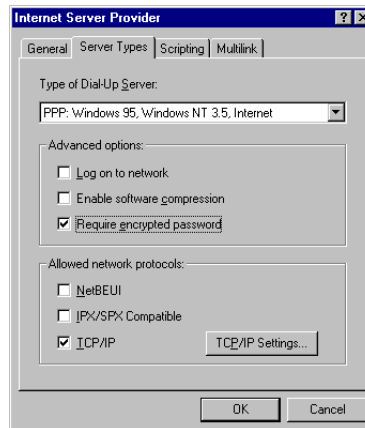
Configure PPP Link to the Internet Service Provider:

- Open the Dial-Up Networking folder by double-clicking **My Computer**, and then **Dial-Up Networking** from the desktop.



- Double-click the **Make New Connection** icon. In the resulting dialog:
 1. Specify a name for the ISP this host will be using.
 2. Select a modem device to use for the ISP PPP link.
 3. Click the **Next** button.

- Here you will need to enter the ISP's telephone number.
 - Click **Next** and then **Finish**. A new icon will be added to the Dial-Up Networking folder. Right-click this icon and select **Properties** to display the properties window.
 - Click the **Server Types** tab.
1. In the **Type of Dial-Up Server** field select: **PPP: Windows 95, Windows NT, Internet**.
 2. In the **Advanced options** box:
 - Disable **Log on to network**.
 - Disable **Enable software compression**.
 - Enable **Require encrypted password**.
 3. In the **Allowed network protocols** box:
 - Disable **NetBEUI**.
 - Disable **IPX**.
 - Enable **TCP/IP**.



- Click the **TCP/IP Settings...** button. Verify the IP address, name service, and compression settings are consistent with those required by the ISP and click **OK**.

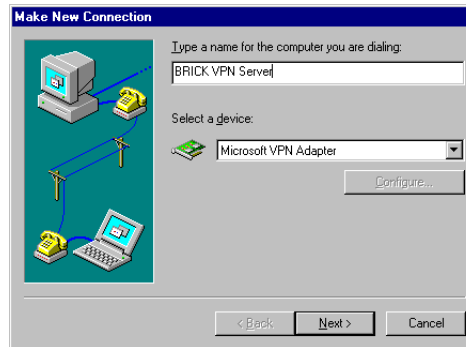


In most cases the default settings in the **Scripting** and the **Multilink** tabs can be left untouched.

- Click **OK** again. The initial PPP link to the Internet Service Provider is now configured. Proceed to the next section to configure the link to the **BRICK** VPN Server.

Configure the PPTP Link to the **BRICK** VPN Server

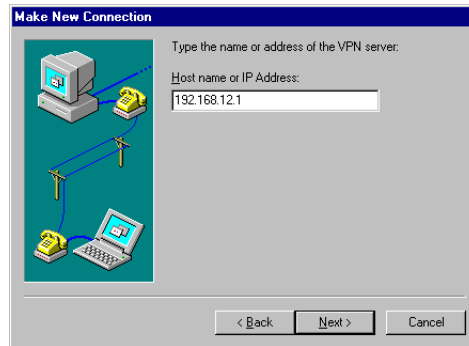
- From the **Dial-Up Networking** folder double-click the **Make New Connection** icon to configure the connection for the **BRICK** VPN Server.



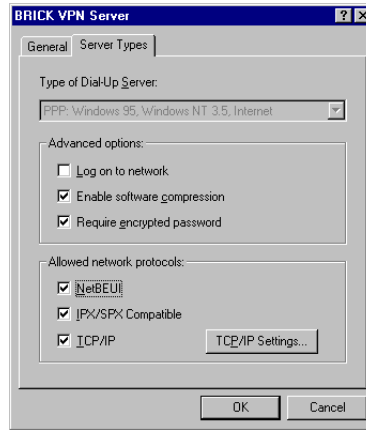
- In the **Type a name for the computer you are dialing** field specify a name for your **BRICK** VPN Server.
- From the **Select a device** drop menu select the device **Microsoft VPN Adapter** and click **Next**.
- In the dialog shown below enter the official IP address of the **BRICK** VPN Server.



If the **Microsoft VPN Adapter** device is not available verify that version 1.2 (or newer) of Microsofts Dial-Up Networking software is installed.



- Click **Next** and then **Finish**. A new icon for the **BRICK** VPN Server will be added to the Dial-Up Networking folder.
 - In the **Dial-Up Networking** folder right-click the new **BRICK** VPN Server icon and select **Properties** to verify the connection settings.
 - Click the **Server Types** tab
1. In the **Type of Dial-Up Server** field select: **PPP: Windows 95, Windows NT, Internet**
 2. In the **Advanced options** box:
 - Enable **Log on to network** if hosts are required to register with the network.
 - Enable **Enable software compression**.
 - Enable **Require encrypted password**.
 3. In the **Allowed network protocols** box enable only those protocols this host will use to communicate with remote hosts on the central site LAN. At a minimum **TCP/IP** must be selected.



4. Click the **TCP/IP Settings...** button.
 - Verify the IP address, name service, and compression settings are consistent with those on the **BRICK** and click **OK**. The settings used here must correspond to the respective **BRICK** VPN partner interface settings (see [VPN Interface Settings, page 129](#)).
5. Click **OK** again to accept the settings for the PPTP link. Once the respective **BRICK** partner interface is configured the Virtual Private Networking connection can be established as described in [Connecting to the BRICK VPN Server, page 151](#).

Configure **BRICK** VPN Server

Requirements A separate VPN license must be installed before the **BRICK** will support VPN connections. A VPN license can be purchased from BinTec Communications AG directly or from your local distributor.

Configure Link to the Internet Service Provider:

- The link to the **BRICK**'s ISP can be configured as a standard dial-up/leased PPP interface via Setup Tool's WAN Partners menu.

Configure the VPN Partner Interface:

- VPN partners are configured in the **VPN** menu. The settings below could be used for the VPN Partner (PPTP client) configured above.

BRICK Setup Tool	BinTec Communications AG
[VPN][ADD]: Configure VPN Interface	MyBRICK
Partner Name	vpn1
Encapsulation	PPP
Compression	none
Encryption	MPPE 40
PPP>	
Advanced Settings>	
IP>	
IPX>	
	SAVE CANCEL
Enter string, max length = 25 chars	

The following field is interesting here:

Field	Meaning
Encryption	you may select <i>MPPE</i> (40 bit or 128 bit session-key) or <i>none</i> .

Table 5-9: **VPN** ➤ **ADD** ➤ **CONFIGURE VPN INTERFACE**



If MPPE 128 was selected the MS-CHAP protocol is required here.

➤ Go to **PPP**.

BRICK Setup Tool		BinTec Communications AG
[VPN][ADD][PPP]: PPP Settings ()		MyBRICK
Authentication	MS-CHAP	
Partner PPP ID	vpnliid	
Local PPP ID	mybrick	
PPP Password	vpnlpass	
Keepalives	off	
Link Quality Monitoring	off	
OK		CANCEL
Use <Space> to select		

The following fields are interesting here:

Field	Meaning
Authentication	The authentication protocol to use when authenticating this partner.
Partner PPP ID	The PPP ID that the VPN partner must identify itself with during PPP negotiation.
Local PPP ID	The BRICK's PPP ID which is used during PPP negotiation with this VPN partner.
PPP Password	The password this VPN partner must use when challenged by the BRICK during PPP negotiation.
Keepalives	This option is only relevant for leased line connections.
Link Quality Monitoring	This option allows you to tell the BRICK to gather PPP Link Quality statistics for a specific PPP partner. When enabled, link statistics are continuously written to the BRICK's biboPPPLQMTTable (viewable from the SNMP shell), when a connection is established with this partner.

Table 5-10: **VPN** ➔ **ADD** ➔ **PPP**

- Because Windows 95 PPTP clients expect the VPN server to assign them an IP address when the “tunnel” is established the **Dynamic IP Address Server** option must be enabled.



For information on the other options available in this menu see the description of the **WAN PARTNERS** ➤ **ADVANCED SETTINGS** menu in your User's Guide.

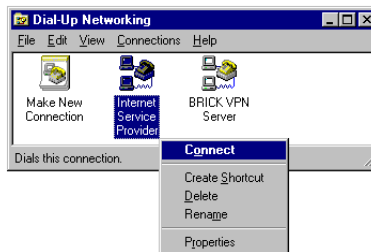
So that the **BRICK** can assign the PPTP client an IP address, make sure there are available IP addresses defined in the **IP** ➤ **DYNAMIC IP ADDRESSES** menu.

Connecting to the **BRICK** VPN Server

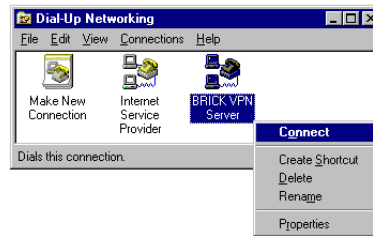
- Open the **Dial-Up Networking** folder by double-clicking **My Computer**, and then **Dial-Up Networking**.



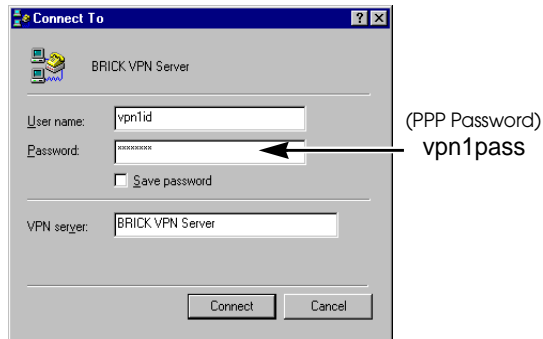
- Right-click the **Internet Server Provider** icon, select **Connect** and enter the user/password assigned by the ISP.



- After connecting to the ISP right-click the **BRICK VPN Server** icon and select **Connect**.



- In the **Connect To** window shown below enter the PPP ID and PPP Password settings configured on the **BRICK** (see [table 5-10, page 150](#)) in the **User name and Password** fields.



5.4.2 Example LAN-to-LAN Configuration

Two distant networks, a corporate central site LAN and a supplier or partner's network can be connected over the Internet via a Virtual Private Network using two **BRICKs** as follows.

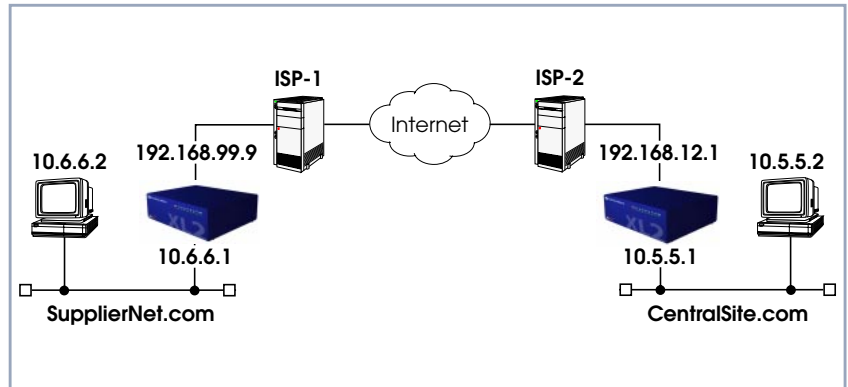


Figure 5-6: Example LAN-to-LAN Configuration

Once both **BRICKs** are configured for Virtual Private Networking hosts on either LAN can connect to hosts on the remote LAN. All traffic that is routed between the two networks is encrypted (user-data encryption). Individual hosts are not required to support PPP or PPTP, the VPN remains transparent.

Configuration on SupplierNet **BRICK**

- A separate license must be installed before Virtual Private Networking can be used. Verify the license is installed in Setup Tool's **LICENSES** menu.
- The link to the ISP-1 can be setup as a standard dial-up/leased PPP interface in the **WAN PARTNERS** menu.
- Configure the VPN Partner interface in the **VPN** menu. The VPN Partner interface for the BRICK-XL2 on **CentralSite.com** could be configured as follows:
 - Define a partner name (*csite*) and enable one or more protocols to support on the link.
 - In the **Encryption** field you may select *MPPE (40 bit or 128 bit session-key)* or *none*. The options specified here must be the same for each partner.
 - Enable (*yes*) the **Identify by Calling Address** option. The VPN partner will be identified by the IP address it uses when establishing the PPP link.
 - In the **Authentication** field select which authentication to use.



If MPPE 128 was selected the MS-CHAP protocol is required here.

- Set **Partner PPP ID** and **PPP Password** as needed.

BRICK Setup Tool	BinTec Communications AG
[VPN][ADD][PPP]: PPP Settings (csite)	MyBRICK
Authentication	CHAP
Partner PPP ID	csiteid
Local PPP ID	mybrick
PPP Password	csitepass
Keepalives	off
Link Quality Monitoring	off
SAVE	CANCEL
Use <Space> to select	

- In the **IP** menu you will need to define the IP addresses the VPN Partner will be using.
The **VPN Partner's IP Address** field for *csite* would be set to *192.168.12.1*. Under **via IP Interface** select the PPP interface for the local ISP. VPN connections to **CentralSite.com** may only be established over this interface. Specify *csite's* LAN address and netmask in the **Partner's LAN IP Address/Netmask** fields.

BRICK Setup Tool	BinTec Communications AG
[VPN][ADD][IP]: IP Configuration (csite)	MyBRICK
VPN Partner's IP Address via IP Interface	192.168.12.1 ISP-1
Identification by IP Address	yes
Partner's LAN IP Address	10.5.5.1
Partner's LAN Netmask	255.0.0.0
SAVE	CANCEL
Enter string, max length = 25 chars	

Configuration on Central Site **BRICK**

- A separate license must be installed before Virtual Private Networking can be used. Verify the license is installed in Setup Tool's **LICENSES** menu. The status for "tunnel" must be *valid*.
- The link to the ISP-2 can be setup as a standard dial-up/leased PPP interface in the **WAN PARTNER** menu.
- Configure the VPN Partner interface in the **VPN** menu. The VPN Partner interface for the BRICK-XL2 on **SupplierNet.com** could be configured as follows:
 - Define a partner name (*SupplierNet*) and enable one or more protocols to support on the link.
 - In the **Encryption** field you may select *MPPE (40 bit or 128 bit session-key)* or *none*. The options specified here must be the same for each partner.
 - Enable (*yes*) the **Identify by Calling Address** option. The VPN partner will be identified by the IP address it uses when establishing the PPP link.
- In the **Authentication** field select which authentication to use.



If *MPPE 128* was selected the *MS-CHAP* protocol is required here.

- Set Partner **PPP ID** and **PPP Password** as needed.

BRICK Setup Tool	BinTec Communications AG
[VPN][ADD][PPP]: PPP Settings (SupplierNet)	csite
Authentication	CHAP
Partner PPP ID	supplierid
Local PPP ID	mybrick
PPP Password	supplierpass
Keepalives	off
Link Quality Monitoring	off
OK	CANCEL
Use <Space> to select	

- In the **IP** menu you will need to define the IP addresses the VPN Partner will be using.
The **VPN Partner's IP Address** field for *SupplierNet* would be set to *192.168.99.99*.
- Under **via IP Interface** select the PPP interface for the local ISP. VPN connections to **SupplierNet.com** may only be established over this interface. Specify **SupplierNet's** LAN address and netmask in the Partner's **LAN IP Address/Netmask** fields.

BRICK Setup Tool	BinTec Communications AG
[VPN][ADD][IP]: IP Configuration (SupplierNet)	csite
VPN Partner's IP Address via IP Interface Identification by IP Address	192.168.12.1 ISP-1 yes
Partner's LAN IP Address Partner's LAN Netmask	10.5.5.1 255.0.0.0
Advanced Settings>	
SAVE	CANCEL
Enter string, max length = 25 chars	

6 X.25

We start this chapter with an introduction to X.25 to give you an overview of the X.25 protocol.

Then we will cover all of the menus and settings you will see while using Setup Tool to configure the X.25 protocol on your router.

Following that are several brief examples for configuring the available X.25 features on your router.

Under Utilities you find the X.25 PAD and a reference of X.25 relevant SNMP shell commands.

Lastly, hardware specifications for the CM-X21 communications module are covered.

6.1 An Introduction to X.25

Packet Switching X.25 is commonly referred to as being a Connection-Oriented, Reliable, Packet-Switched network. These catchwords describe some of the important characteristics of X.25 networks which are explained briefly here to help you better understand X.25.

Connection-Oriented X.25 is connection-oriented which means that when data needs to be transferred, a connection must first be established. Communications parameters such as window size and packet sizes are negotiated when the connection is first established.

Multiple connections between two end points can be achieved by multiplexing logical connections onto data links. Different logical connections (or “Virtual Circuits”) are identified by assigning a virtual circuit number for each logical connection. This number is included in the header of each X.25 data-packet.

Packet Switched X.25 is a packet switched network which means that user data is divided up and placed into X.25 packets of a predefined maximum length (usually 128 bytes). Each packet is assigned a virtual circuit number and is transmitted over the data link.

With a 128 byte packet size, user data must normally be fragmented into many packets. The X.25 frame format defines a special field, M-bit (M for more), which is used to allow fragmented packets to be reunited at the receiving station.

Reliable X.25 connections are reliable connections which means that all data packets sent are confirmed by the receiving station. This is achieved using either special packets (Receiver Ready packets) or by having the receiving station “piggy-back” confirmation messages onto other packets. Also, in X.25, packets always arrive in sequence at the receiving station.

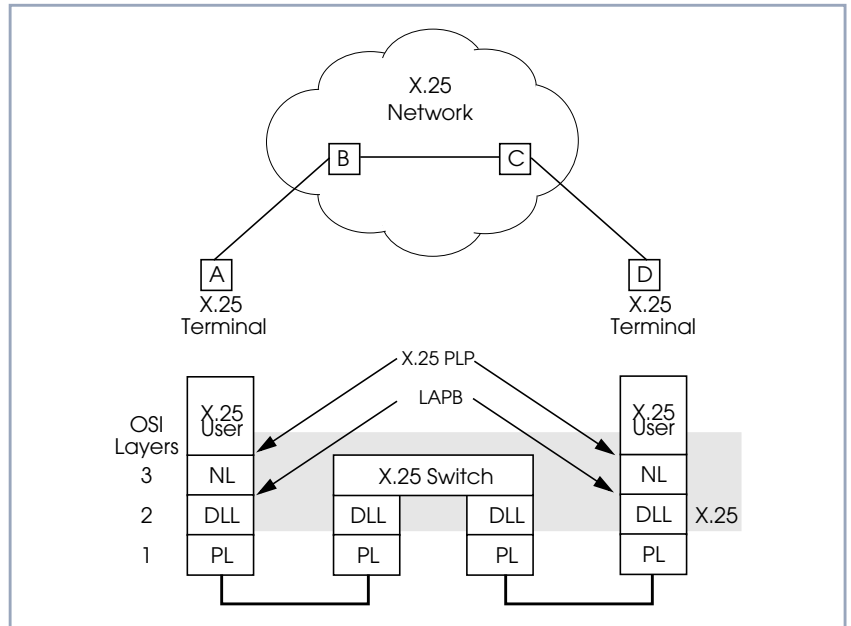


Figure 6-1: X.25 Network Scenario

6.1.1 Call Setup

Before data can be exchanged among X.25 partners an X.25 call must be set-up. An X.25 CALL packet is sent by the calling partner to the called partner who can accept/refuse the connection. Once a call has been established, a unique Virtual Circuit (VC) number is assigned to the connection which is used throughout the duration of the connection.

If an X.25 network lies between two end stations, the VC numbers used by each end station may be different. For example, if hosts A and D in the diagram above are communicating, the VC number used for the A–B connection may be different from the one used for C–D.

After the call is initially setup all packets exchanged between the partners follow a fixed path defined during the initial call setup phase. Once the connection is no longer needed, it can be disconnected, and later reused by the same or different communications partners.

6.1.2 Data Links and Virtual Circuits

Data Link A data link is a direct, point-to-point, connection between two X.25 stations. This physical connection can be via an ISDN B or D channel, an X.21 connection, or an ethernet connection (LLC2). On a point-to-multipoint physical medium (i.e. ethernet), multiple point-to-point data links are multiplexed over the same physical interface.

Virtual Channel A virtual channel (VC) is a Logical Connection that is multiplexed onto a data link. This means that multiple X.25 connections can exist over the same physical medium, simultaneously.

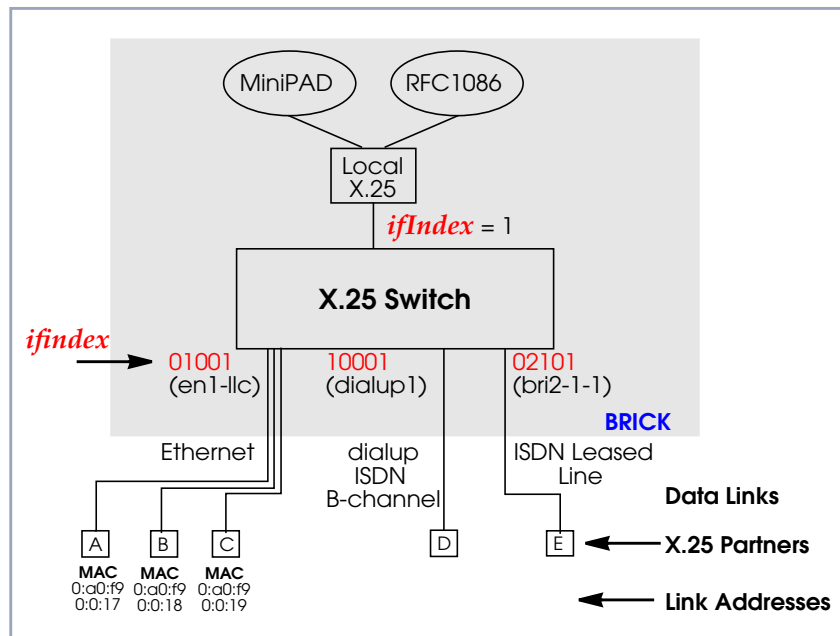


Figure 6-2: Data Links and Virtual Channels

In X.25, each data link uses one interface. The characteristics of each data link are defined in **SETUP TOOL** ➤ **X.25** ➤ **LINK CONFIGURATION** menu or in the **x25LinkPresetTable**. These characteristics, such as window and packet size, can be changed by editing these links.

To display a list of all available interfaces known to the system you can use the `ifstat` command.

There are three types of interfaces available on the **BRICK**; the first of which is always available. The other interface types will depend on your particular configuration.

■ Local Interface

The local interface is a special interface and is always available on the **BRICK**.

■ Point-to-Point Interface

This interface is referred to as being Point-to-Point because the two end stations of the connection are determined solely by the **IfIndex**. These interfaces include: ISDN dialup, ISDN leased lines, and X.31 interfaces.

■ Point-to-Multipoint Interface

The Point-to-Multipoint interface is referred to as such because the **IfIndex** does not completely specify an end-to-end connection. Additional information is required (such as the end stations MAC address) when creating these interfaces to provide an end-to-end link. These interfaces include: LAN connections over LLC2.

6.1.3 Point-to-Point and Point-to-Multipoint Interfaces

One of the characteristics of an X.25 interface that must be defined is the encapsulation it uses.

When creating X.25 Point-to-Point interfaces in the **WAN PARTNER** ► **Add** menu in Setup Tool or in the **bibOPPTTable**, you can specify either **x25** or **x25_ppp** encapsulation. By default, x25 encapsulation is used. This allows an interface to be used solely for X.25 traffic. Using **x25_ppp** allows PPP and X.25 traffic to be routed over the same interface (i.e. multiplexing IP datagrams and X.25 packets simultaneously over the same ISDN channel).

For X.25 Point-to-Multipoint interfaces such as ethernet, you must use the `enx*-llc` interfaces, since not all ethernet interfaces on the **BRICK** support X.25 (i.e. `enx`, `enx-snap`, and `enx-nov802.3`)

6.1.4 X.25 Addressing Schemes

As in TCP/IP networks, each host in an X.25 network must be uniquely identified before communication between them is possible. However, there is one important difference. In TCP/IP, each data packet contains the source/destination addresses and is routed individually (packets can take different paths). In X.25, addresses are only used during call setup and all subsequent data packets follow the same exact route.

In X.25, three different address formats, can be used to identify X.25 hosts.

- Standard X.25 Addressing (X.121)
- Extended X.25 Addressing
- NSAP (Network Service Access Point) Addresses (X.213)

Standard X.25 Addressing (X.121)

The X.121 addressing scheme is the oldest and most common format used in X.25 networks. X.121 addresses consist of up to 15 digits and may begin with a leading escape digit (normally a 0). If the leading 0 is present, it is assumed to be an international address, otherwise a national address is assumed. For example (Note that spaces in the example addresses are used only for added readability):

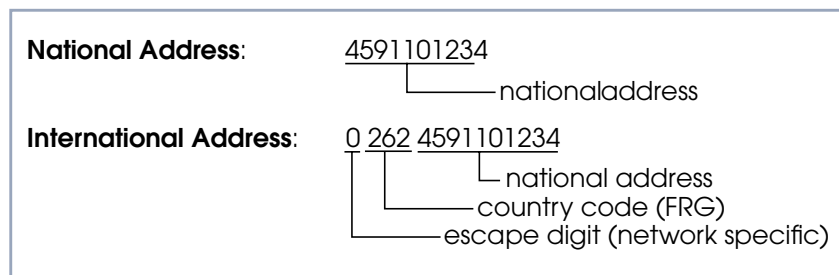


Figure 6-3: Standard X.25 Addressing (X.121)

When working within ISDN, E.164 addresses are used instead of X.121 addresses. E.164 describes the numbering plan of the ISDN network and the commonly known telephone numbering system consisting of country code, area

code, and subscriber number. To address other ISDN devices, an international ISDN number (according to E.164) is used which is similar to a national X.121 address. An additional zero following the escape code specifies an ISDN address for internetworking. For example:

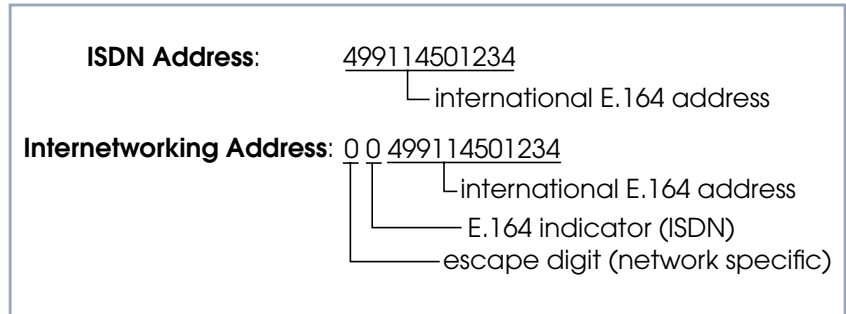


Figure 6-4: E.164 Addressing within ISDN

Extended X.25 Addressing

The extended addressing format provides a standardized way for distinguishing different types of addresses in X.25. However, many public networks do not support this addressing format (The **BRICK** supports extended addresses and differentiates between standard and extended addresses using a leading @ in the ~Addr field).

When the call is setup, a special bit (the A bit) in the call packet is used to define whether the addresses used are standard or extended. When the A bit is set, an extended address is used which consists of up to 255 digits (Most implementations are currently using less than 42 digits). The first two digits have special

meanings and specify the Type of Address (TOA) and Numbering Plan Identification (NPI) respectively.

Sequence	Digits	TOA and NPI Digits
First Digits	0	Network dependent number
	1	International number
	2	National number
Second Digits	1	E.164 ISDN numbering plan
	3	X.121 numbering plan

Table 6-1: Extended X.25 Addressing

For example, the following addresses are characterized according to their TOA and NPI digits (Spaces in the example addresses are used only for added readability).

Addresses	Digits
A national X.121 address	@2 3 4591101234
An international X.121 address	@2 3 4591101234
National E.164 address	@2 1 9114501234
International E.164 address	@1 1 49 9114501234

NSAP Addresses (X.213)

An alternative to the standard and extended formats is the NSAP (Network Service Access Point) address format. The NSAP format is defined in X.213. Only a few public networks support this format.

The NSAP format is complex. For our purposes it should be sufficient to say that NSAP addresses consist of up to 40 hexadecimal characters.

Two types of NSAP addresses also exist, OSI conform (indicated by a leading X) and Non-OSI conformant (indicated by a leading N).

Some example NSAP addresses are as follows:

Addresses	Digits
OSI compatible address	X 37 26245911012340 4711 abc
Non-OSI compatible address	N 0123456789abcdef

NSAPS can be used, instead of or in addition to, the other address formats.

6.1.5 X.25 Routing

To give you an overview of X.25 routing we use the **x25RouteTable** of the MIB, which shows X.25 routing systematically. To configure routes via the Setup Tool, you must enter the menu **X.25 ► ROUTING ► ADD** as described in the following chapter.

The routing of X.25 packets is accomplished via a routing table similar to the **ipRouteTable**. The **BRICK** uses entries in the **x25RouteTable** to determine which link to route X.25 calls it receives. Routing decisions can be made based on the source link and/or different parameters found in the call packet.

The routing table for our example switch (see [chapter 6.1.2, page 162](#)) might look as follows:

	SrcIfIndex	SrcLinkAddr	DstAddr	DstIfIndex	DstLinkAddr
00	en1-llc	0:a0:f9:0:0:17		dialup1	
01	dialup1			en1-llc	0:a0:f9:0:0:17
02	en1-llc	0:a0:f9:0:0:18		bri2-1-1	
03	bri2-1-1			en1-llc	0:a0:f9:0:0:18
04	en1-llc	0:a0:f9:0:0:19	[0-4]*	dialup1	
05	en1-llc	0:a0:f9:0:0:19	[5-9]*	bri2-1-1	

Table 6-2: Example Switch Routing Table

Here, the first two entries route all calls between partners A and D. The third and fourth entries provide routes for all calls between partners B and E. The last

two entries specify routes for calls originating from partner C. Any calls to an X.25 destination address beginning with 0, 1, 2, 3, or 4 are routed to D. All calls beginning with 5, 6, 7, 8, or 9, originating from C, are routed to E.

Calls with extended addresses are not routed since no routing entry for calls with a leading “@” is present. Therefore, such calls are refused.

Since some calls may match more than one route in the table, a metric can be used to prioritize routes. A route with the lowest metric value always has higher priority.

6.2 Setup Tool Menus

After entering `setup` from the shell prompt Setup Tool's Main Menu is displayed as below. Depending on your hardware setup and software configuration your router's menu may differ slightly.

```
BRICK Setup Tool                                     BinTec Communications AG
                                                    MyBRICK

Licenses                System
Slot1:                  CM-BNC/TP, Ethernet
Slot2:                  CM-2XBRI, ISDN S0, Unit 0
                       CM-2XBRI, ISDN S0, Unit 1
Slot3:                  CM-1BRI, ISDN S0

WAN Partner
IP          IPX          X.25

Configuration Management
Monitoring and Debugging
EXIT

Press <Ctrl-n>, <Ctrl-p> to scroll through menu items, <Return> to
enter
```

➤ **Go to X.25.**

This is the point where our exploration of Setup Tool begins.

The X.25 menu contains several submenus used to configure the X.25 protocol on the router.

BRICK Setup Tool [X.25]: X.25 Configuration	BinTec Communications AG MyBRICK
<pre> Static Settings Link Configuration Routing Multiprotocol over X.25 EXIT </pre>	
Press <Ctrl-n>, <Ctrl-p> to scroll through menu items, <Return> to enter	

Field	Meaning
Static Settings	contains the router's X.25 address
Link Configuration	lists all X.25-compatible interfaces on the router, and is used to configure them respectively
Routing	contains the router's X.25 routing table.
Multiprotocol over X.25	is used to configure the Multiprotocol Routing over X.25 (MPX25) feature.

Table 6-3: *X.25 CONFIGURATION*

- Static Settings** ➤ Go to **STATIC SETTINGS**.
The X.25 Static Settings menu contains the router's local X.25 address.

BRICK Setup Tool	BinTec Communications AG
[X.25][STATIC]: X.25 Static Settings	MyBRICK
Local X.25 Address	
SAVE	CANCEL
Press <Ctrl-n>, <Ctrl-p> to scroll through menu items, <Return> to enter	

Field	Meaning
Local X.25 Address	<p>The router's official X.25 address. Setting this variable is only required if the router is not directly connected to an official X.25 data network. When connected directly, the router ascertains its X.25 address automatically.</p> <p>The X.25 address must be set here for sites implementing private X.25 networks, or when X.25 in the B-channel is used.</p>

Table 6-4: X.25 ► STATIC SETTINGS

Link Configuration ► Go to **LINK CONFIGURATION**.

This menu displays a list of all interfaces that support the X.25 protocol. The number of available interfaces listed here is a combination of hardware (which modules are installed) and software interfaces (configured WAN partners).

- Dialup interfaces
Entries for each X.25-compatible WAN partner configured on the system.
- Hardware interfaces
Depending on which slot the X.21 module is installed in (1 through 3 on a BRICK-XM, 1 through 6 on a BRICK-XL2), the system creates an initial link using xi1 through xi3 (xi6).

- X.31 interfaces

If you are receiving X.31 services from your ISDN provider an X.31 link is also present. X.31 links have the format:

x31d-<slot number>-<unit number>-<TEI>

```
BRICK Setup Tool                               BinTec Communications AG
[X.25][LINK]: X.25 Link Configuration          MyBRICK

Select Link to Configure

x31d2-0-1
enl-11c (create new configuration)

DELETE CONFIGURATION      EXIT

Press <Ctrl-n>, <Ctrl-p> to scroll, <Space> tag/untag DELETE, <Return>
to edit
```

- Before an X.25-compatible interface can be used, its link characteristics must first be set.
- To edit an X.25 link mark the entry and then press **Enter**.
- To remove an X.25 link, tag the entry for deletion (spacebar) and select **DELETE CONFIGURATION**.

Configure the X.25 link

- Select **EDIT**.
This menu is used to configure the basic characteristics of the X.25 link.

BRICK Setup Tool		BinTec Communications AG	
[X.25][LINK][EDIT]: Change X.25 Link Configuration		MyBRICK	
Link	enl-llc		
L3 Mode	dte		
L3 Packet Size	default: 128	max: 128	
L3 Window Size	default: 2	max: 7	
Window size/Packet size Neg.	when necessary (default)		
Lowest Two-Way-Channel (LTC)	1		
Highest Two-Way-Channel (HTC)	2		
Partner MAC Address (LLC)			
Layer 2 Behavior	disconnect after timeout		
Disconnect Timeout	1000		
SAVE	CANCEL		
Use <Space> to select			

Field	Meaning
Link	This is the name of the link your are editing and cannot be changed here
L3 Mode	This defines the mode the router operates in at Layer 3 of the X.25 protocol stack. Set to DCE if the router must provide clocking information or DTE if provided by the remote side of link
L3 Window Size / Packet Size	Defines the <i>default</i> and <i>maximum</i> values for Packet size (128, ..., 4096 bytes) and Window size (2 through 127)

Field	Meaning
Window size/Packetsize Neg.	Decides whether window/packetsize negotiation is made for this X.25 link. The possible values are <i>never</i> , <i>always</i> and <i>when necessary</i> , where <i>when necessary</i> is the default value. The value <i>never</i> means no negotiation. When a call arrives that does not correspond to the default size, the call is cleared. <i>Always</i> means negotiations are always made and when <i>when necessary</i> is selected, there are only negotiations, when the requested values differ from the default values.
Lowest Two-Way-Channel (LTC)	LTC and HTC must be set to reflect the number of Virtual Channel(s) you have arranged for from your X.25 network provider.
Highest Two-Way-Channel (HTC)	Defines the highest number that can be assigned to a Virtual Channel.
Partner MAC Address (LLC)	Used when configuring a link for a partner on the LAN and specifies the host's MAC or hardware address.
Layer 2 Behaviour	Defines whether (and if so, when) the link should be disconnected when no virtual channels are active.
Disconnect Timeout	Time in milliseconds to wait before closing the link once the line becomes inactive.

Table 6-5: X.25 ► LINK CONFIGURATION ► EDIT



Caution!

When establishing X.25 connections via ISDN, it may occur that unintentional permanent connections are established in combination with certain settings.

It is important to note that if **L2IdleTimer** is set to **-1** in the **X25LinkPresetTable**, or in the Setup Tool field **Layer 2 Behaviour** to *always active*, the **BRICK** will continue to establish layer 2 with the effect of permanent B-channel connections and increased costs.

- Thus, if you want to prevent this, make sure to give the **L2IdleTimer** variable a value other than **-1** or to a setting other than *always active* in Setup Tool's **Layer 2 Behaviour** field.

Configure X.25 Routes

- Go to **ROUTING**.

This menu displays the X.25 routing table. X.25 routes are used for routing traffic over X.25 interfaces. Routes can be added, removed, or changed here.

Source Link	Dest. Link	Dest. Link Addr.	Dest X.25 Addr.	Metric
ADD	DELETE	EXIT		

To edit an X.25 route, mark the entry and then press **Return**.

- Select **ADD**.
X.25 routes configured with Setup Tool are based on two factors:
 - Source link
Link X.25 call_packet first arrived on.
 - Dest. X.25 Address
The address the packet is addressed to.

You must define the destination link where the X.25 packets will be routed by specifying these two parameters. Standard wildcard characters can also be used in the Destination Address parameter.

Example	Meaning
{123}45	Either 12345 or 45
[68]*	Any # starting with 6 or 8
[^5]*	Any # not starting with 5
624*	All #s starting with 624

Table 6-6: Examples for Wildcard Usage

Since some calls may match more than one route in the table, a metric can be used to prioritize routes. A route with the lowest metric value always has higher priority.

When your destination link is a multipoint interface, you additionally have to adjust the Destination Link Address (LLC).

Also note that there are different X.25 addressing standards, and depending on where the X.25 partner is calling from, the actual X.25 address received by the router may differ.

BRICK	BinTec Communications AG
[X.25][ROUTING][EDIT]: Add or Change X.25 Routes	MyBRICK
Source Link	any
Destination Link	local
Destination X.25 Address	45*
Metric	0
SAVE	CANCEL
Use <Space> to select	

Multiprotocol Routing over X.25

➤ Go to **MULTIPROTOCOL OVER X.25**.

This menu lists the Multiprotocol Routing over X.25, or MPX25, interfaces configured on the system. MPX25 allows the router to route IP, IPX, and Bridge, traffic over X.25 links. Each MPX25 interface defines an X.25 link to route one or more protocols over.



The underlying X.25 subsystem must first be configured before any MPX25 interface can be configured here. See the menus:

- **X.25** ➤ **STATIC SETTINGS**
- **X.25** ➤ **LINK CONFIGURATION**
- **X.25** ➤ **ROUTING**

BRICK Setup Tool	BinTec Communications AG	
[X.25][MPR]: Multiprotocol over X.25	MyBRICK	
Interface Name	Destination X.25 Address Encapsulation	
ADD	DELETE	EXIT

- Select **ADD**.
Use this menu to add or change MPX25 interfaces.

BRICK Setup Tool	BinTec Communications AG
[X.25][MPR][ADD]: Add or change X.25 MPR	MyBRICK
Partner Name	mpxpartner1
Encapsulation	ip_rfc877
X.25 Destination Address	49911555
Advanced Settings>	
IP>	
IPX>	
SAVE	CANCEL
Enter string, max length = 25 chars	

Field	Meaning
Partner Name	Enter a unique name to identify this MPX25 partner
Encapsulation	<p>Here you select the type of encapsulation/protocol to use. Note that the remote MPX25 partner must be configured to use the same encapsulation.</p> <p>When selecting <i>ip_rfc877</i> or <i>ip</i>, you must define the IP settings in the IP Submenu (see below).</p> <p>When selecting <i>mpr</i>, you can enter IP and IPX settings in the respective submenus (see below). When you define the settings for both submenus, both will be routed, but you can also decide to configure just one of the protocols or none of it. The Bridge functionality is always available, when <i>mpr</i> is selected and needs no configuration.</p> <p>When selecting <i>ipx</i>, you must define the IPX settings in the IP menu.</p>

Field	Meaning
X.25 Destination Address	The X.25 address for this partner. There must be an appropriate X.25 route for this address in the X.25 routing table. The special "{" and "}" characters can be used to define an optional string of digits to use when matching incoming X.25 calls. For outgoing calls to this partner, the digits between these characters are used. <i>{00}4991155</i> matches both <i>004991155</i> and <i>4991155</i> for incoming calls, outgoing calls are placed using <i>004991155</i> .

Table 6-7: X.25 ► MULTIPROTOCOL OVER X.25 ► ADD

Protocol			Encapsulation
IP	IPX	Bridge	
X			ip_rfc877
X			ip
X	X	X	mpr
	X		ipx

Table 6-8: Encapsulation

Configuring IP Settings► Go to **IP**.

This is where you configure the IP settings for this remote MPX25 partner and is only available if the IP protocol or *mpr* has been enabled.



The settings used in this menu are the same as those used in the **WAN PARTNER ► ADD ► IP** menu but only apply to this MPX25 partner.

Configuring IPX Settings► Go to **IPX**.

This is where you configure the IP settings for this remote MPX25 partner and is only available if the IP protocol or *mpr* has been enabled.



The settings used in this menu are the same as those used in the **WAN PARTNER ► ADD ► IPX X.25** menu but only apply to this MPX25 partner.

- Go to **X.25 ► MULTIPROTOCOL OVER X.25 ► ADD ► ADVANCED SETTINGS**.

This menu can be used to configure advanced features.



The settings used in this menu are a subset of those used in the **WAN PARTNER ► ADD ► ADVANCED SETTINGS** menu but only apply to this MPX25 partner.

Monitoring the Router's Operational Status

- Go to **MONITORING AND DEBUGGING**.
This menu consists of several submenus which allow you to monitor the router's operational status (and debug problems) in different ways.

BRICK Setup Tool [MONITOR]: Monitoring and Debugging	BinTec Communications AG MyBRICK
ISDN Monitor ISDN Credits X.25 Monitor Interfaces Messages TCP/IP OSPF EXIT	

Field	Meaning
ISDN Monitor	lets you track incoming and outgoing ISDN calls.
ISDN Credits	lets you track credits based accounting
X.25 Monitor	lets you track incoming and outgoing X.25 calls
Interfaces	lets you monitor traffic by interface
Messages	displays system messages generated by the router's system logging and accounting mechanisms
TCP/IP	menu lets you monitor IP traffic by protocol
OSPF	menu lets you monitor IP traffic by protocol

Table 6-9: **MONITORING AND DEBUGGING**

➤ Go to **X.25 MONITOR**.

The X.25 Monitor menu initially display all active X.25 connections. These calls include leased and dialup connections made through X.25 public networks or over ISDN.

As when using the ISDN Monitor, the menu commands (c, h, d, and s) listed at the bottom of the screen list different statistics relating to X.25 calls.

BRICK Setup Tool				BinTec Communications AG	
[MONITOR][X.25 CALLS]: X.25 Monitor				MyBRICK	
From	To	Calling Addr	Called Addr	Duration	
xi3	local	1 0	0	591	
EXIT					
(c)alls(h)istory(d)etails(s)tstatistics					

The **(c)alls** listing shows currently established X.25 connections.

From	To	Calling Addr	Called Addr	Duration
xi1	local	1	0	591
mpr-1	london2	3	2	139

Figure 6-5: (C)alls Listing

The **(h)istory** listing shows a list of completed X.25 connections (both incoming and outgoing) since the last system reboot.

From	To	Starttime	Duration	Cause
xi1	central	19:33:52	0	(0x01) number busy
local	london2	19:34:01	2	(0x03) network congestion

Figure 6-6: (H)istory Listing

For completed calls, you can display additional information about the call. Select a call from the list, then enter **d** to see a detailed listing.

The **(d)etails** listing shows specific information about completed calls.

```

Clear Cause                      Clear Diag
Proro ID      1                   State      dataxfer

Source:
Interface     paris-dialup
VC Number     1
X.25 Address
Link Address

Destination:
Interface     local
VC Number     1
X.25 Address  555
Link Address

Packet Size (In/Out)  128/128   Window Size (In/Out) 2/2
EXIT

```

Figure 6-7: (D)etails Listing

The **(s)tatistics** listing shows transfer activity for established X.25 calls.

```

Duration 971

Send:                                Receive:
Packets      1555                      Packets      1552
Bytes        10032                     Bytes        20999

Packets/s    0                          Packets/s    0
Bytes/s      0                          Bytes/s      0

```

Figure 6-8: (S)tatistics Listing

6.3 X.25 Features

The following pages describe configuring some of the most common X.25 features on the router such as:

How do I configure an X.31 link (X.25 in the D-channel)?

How do I route IP traffic over X.25 with MPX25?

How do I configure X.31 in the B-channel (Case A/Case B)?

How do I configure my X.21 module so I can access my X.25 network?

How do I configure X.25 access for a host on my LAN?

How do I configure ISDN dialup access for an X.25 partner?

How do I configure X.25 dialout without configuration?

How do I use the router as a TCP-X.25 bridge?

How do I configure the routing for using an X.25 PAD?

Special Note: The X.25 Local Interface

In X.25 routing the router decides where to forward X.25 calls based on the configured X.25 routes. An X.25 route can lead to a point-to-multipoint interface such as an ethernet, or a point-to-point interface such as a dialup ISDN or X.25 network partner. Another option is the router's special "local" interface.

This local interface is an internal virtual interface. Here, the X.25 packet is given to one of the router's software processes depending on contents (user data field) of the X.25 packet. The respective software process may need to reroute the call in which case the packet is passed back to the lower level routing in-

stance. For example, when routing IP traffic over X.25 links ([Configure a New MPX Partner](#), page 206).

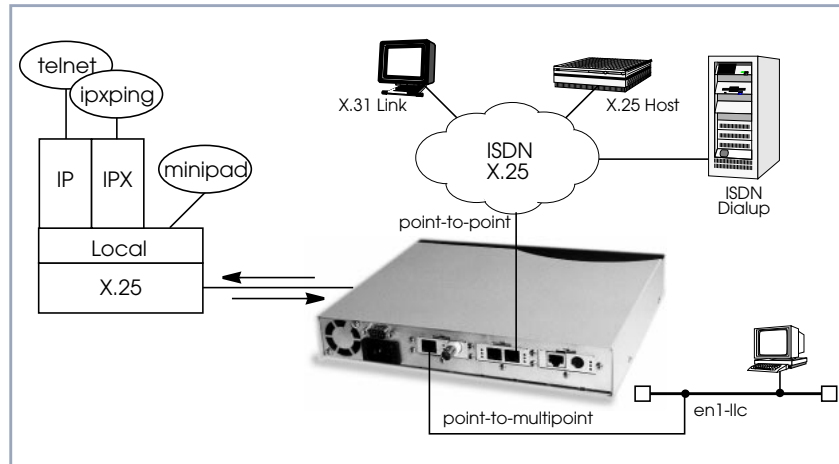


Figure 6-9: Local Interface

6.3.1 How do I Configure an X.31 Link (X.25 in the D-Channel)?

X.31 is a supplementary service offered by your ISDN provider which allows X.25 packets to be transmitted over an ISDN D-channel. This section describes configuring the X.31 data link that can be used by hosts on the LAN to connect to stations on the public X.25 network.



Before you begin

Before you start, verify the following information from your ISDN carrier.

- The TEI value assigned to this interface.
- The Window and Packet size to use for Layer 3.
- The router's X.25 address.
- The ISDN telephone number for this subscriber outlet.

- Verify License** ➤ Verify in the **LICENSES** menu that your X.25 license is valid. You should find “X25 (valid)”
- Configure the X.31 Link** ➤ Go to **X.25** ➤ **LINK CONFIGURATION**.
If the router is connected to the ISDN subscriber outlet you’re receiving the X.31 service on, you should see an X.31 link in this menu, otherwise connect the cabling and reboot the system. When autodetected properly this link has the form:
- ```
x31d<Module Slot>-<ISDN Unit>-<TEI Value>
```
- Verify the detected TEI value is correct then mark the link and press **Return** to define the characteristics of this data link.

| BRICK Setup Tool                           |                          | BinTec Communications AG |  |
|--------------------------------------------|--------------------------|--------------------------|--|
| [X.25][LINK][ADD]: X.25 Link Configuration |                          | MyBRICK                  |  |
| Link                                       | x31d2-0-1                |                          |  |
| L3 Mode                                    | dte                      |                          |  |
| L3 Packet Size                             | default:128              | max:128                  |  |
| L3 Window Size                             | default:2                | max:7                    |  |
| Window size/Packetsize Neg.                | when necessary (default) |                          |  |
| Lowest Two-Way-Channel (LTC)               | 1                        |                          |  |
| Highest Two-Way-Channel (HTC)              | 2                        |                          |  |
| L2 Window Size                             | 2                        |                          |  |
| Layer 2 Behavior                           | disconnect when idle     |                          |  |
|                                            | SAVE                     | CANCEL                   |  |
| Use <Space> to select                      |                          |                          |  |

- Create Route for Incoming Calls** ➤ Go to **ROUTING**.  
Create a route for incoming calls. This will allow calls arriving on the X.31 link that are addressed to the router’s X.25 address to be given to the local interface (see for information [Special Note: The X.25 Local Interface, page 185](#)).
- Result** PAD calls are given to the PAD subsystem, calls containing IP data go to the IP subsystem, etc.

|                                   |            |                          |                  |
|-----------------------------------|------------|--------------------------|------------------|
| BRICK Setup Tool                  |            | BinTec Communications AG |                  |
| [X.25][Routing]: X.25 Route Table |            | MyBRICK                  |                  |
| Source link                       | Dest. Link | Dest. Link Addr.         | Dest X.25 Metric |
| ADD                               | DELETE     | EXIT                     |                  |

The following entries should be made:

| Field                           | Value                   |
|---------------------------------|-------------------------|
| <b>Source link</b>              | x31d<slot>-<unit>-<TEI> |
| <b>Destination Link</b>         | local                   |
| <b>Destination X.25 Address</b> | router's ISDN telno     |

Table 6-10: **X.25 ROUTE TABLE**: Incoming Calls



The router's ISDN telephone number used here should be in the format:  
<country code><area code><local number>

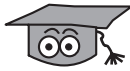
### Create Route for Outgoing Calls

- Go to **ROUTING**.
- Create an X.25 route for outgoing calls. This route says that all calls from the local interface are routed to the X.31 link (see for information [Special Note: The X.25 Local Interface, page 185](#)).

| Field                    | Possible Value          |
|--------------------------|-------------------------|
| Source link              | <i>local</i>            |
| Destination Link         | x31d<slot>-<unit>-<TEI> |
| Destination X.25 Address | leave empty             |

Table 6-11: **X.25 ROUTE TABLE:** Outgoing Calls

### More Info



Testing the X.31 Link:

You can test the X.31 link from a remote X.25 host using a PAD (Packet Assembler Disassembler) by calling the router at its X.25 address.

In Germany, a special “Echo Port” provided by the Deutsche Telekom can be used to verify your router is accessible over X.31.

- Using minipad from the SNMP shell call the echo port with: `minipad 026245911029002.`
- You should see a login prompt. Close the X.25 call with **Control-P**.
- You can also connect to the Deutsche Telekom’s Traffic Generator service to verify data transfers are possible over the X.31 link. This can be done with: `minipad 026245911029003.`

### 6.3.2 How do I Configure X.31 in the B-Channel (Case A/Case B)?

The router supports X.31 in the B-channel according to Case A and B. Case A and B are alternative procedures that can be used to access the public X.25

network from an S0 interface. In both scenarios the router accesses X.25 hosts through the Packet Handler Interface (PHI) provided by the ISDN carrier.

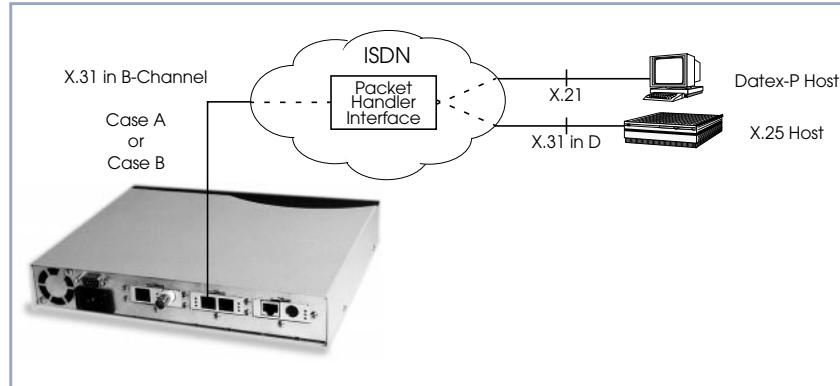


Figure 6-10: X.31 in B-Channel

When using the X.31 in the B-channel on the router, a WAN Partner interface can be configured for this PHI that can be used as a virtual router for all X.25 hosts. Individual X.25 Partner interfaces are not required.



### Before you begin,

you will need the following information:

- The router's ISDN telephone number.
- Case A only) The telephone number of your local PHI. Contact your local carrier for this information.

### Configure WAN Partner

- Go to **WAN PARTNER** ➤ **ADD**.
- Firstly, configure the PHI as a new WAN partner.

|                                     |                |                          |
|-------------------------------------|----------------|--------------------------|
| BRICK Setup Tool                    |                | BinTec Communications AG |
| [WAN][EDIT]: Configure WAN Partner  |                | MyBRICK                  |
| Partner Name                        | phi            |                          |
| Encapsulation                       | X.31 B-Channel |                          |
| Compression                         | none           |                          |
| Encryption                          | none           |                          |
| Calling Line Identification         | no             |                          |
| PPP>                                |                |                          |
| Advanced Settings>                  |                |                          |
| WAN Numbers>                        |                |                          |
| IP>                                 |                |                          |
| IPX>                                |                |                          |
| SAVE                                |                | CANCEL                   |
| Enter string, max length = 25 chars |                |                          |

- Go to **WAN NUMBERS**.
- Set your PHI's ISDN number if your carrier supports Case A. For Case B you do not need to configure the number.

| Field             | Possible Value         |
|-------------------|------------------------|
| <b>WAN Number</b> | PHI's telephone number |
| <b>Direction</b>  | <i>both</i>            |

Table 6-12: **WAN NUMBERS** Configuration

- Configure the Link**
- Go to **X.25** ➤ **LINK CONFIGURATION** ➤ **X.25 LINK CONFIGURATION**.
  - Set the link characteristics for the partner you just created in the previous step. In most cases the following can be used. If connections can not be established, verify with you carrier.

| BRICK Setup Tool                           |                          | BinTec Communications AG |  |
|--------------------------------------------|--------------------------|--------------------------|--|
| [X.25][Link][ADD]: X.25 Link Configuration |                          | MyBRICK                  |  |
| Link                                       | x31d2-0-1                |                          |  |
| L3 Mode                                    | dte                      |                          |  |
| L3 Packet Size                             | default:128              | max:128                  |  |
| L3 Window Size                             | default:2                | max:7                    |  |
| Window size/Packetsize Neg.                | when necessary (default) |                          |  |
| Lowest Two-Way-Channel (LTC)               | 1                        |                          |  |
| Highest Two-Way-Channel (HTC)              | 2                        |                          |  |
| L2 Window Size                             | 2                        |                          |  |
| Layer 2 Behavior                           | disconnect when idle     |                          |  |
|                                            | SAVE                     | CANCEL                   |  |

- Route for Incoming Calls**
- Go to **X.25** ➤ **ROUTING** ➤ **ADD**.
  - Create a route for incoming calls. This will allow calls coming from our PHI interface that are addressed to the router's X.25 telephone number to be given to the local interface (see for information [Special Note: The X.25 Local Interface, page 185](#)).
  - Insert the **Source Link**, e.g. **x31d2-0-1**.
  - Insert the interface name for PHI, e.g. **local** as **Destination Link**.
  - Insert the router's ISDN telephone number as **Destination X.25 Address**, e.g. **12345**.



|                                        |           |                          |
|----------------------------------------|-----------|--------------------------|
| BRICK Setup Tool                       |           | BinTec Communications AG |
| [X.25][ROUTING][ADD]: X.25 Route Table |           | MyBRICK                  |
| Source Link                            | x31d2-0-1 |                          |
| Destination Link                       | local     |                          |
| Destination X.25 Address               | 12345     |                          |
| Metric                                 |           |                          |
| SAVE                                   | CANCEL    |                          |
| Use <Space> to select                  |           |                          |

### Route for Outgoing Calls

- Create another route for outgoing calls. This route says that all calls from the local interface are routed to the PHI (see for information [Special Note: The X.25 Local Interface, page 185](#)).
- Insert the **Source Link**, e.g. **local**.
- Insert the interface name for PHI, e.g. **x31d2-0-1** as **Destination Link**.
- Leave **Destination X.25 Address** empty.

|                                        |           |                          |
|----------------------------------------|-----------|--------------------------|
| BRICK Setup Tool                       |           | BinTec Communications AG |
| [X.25][ROUTING][ADD]: X.25 Route Table |           | MyBRICK                  |
| Source Link                            | local     |                          |
| Destination Link                       | x31d2-0-1 |                          |
| Destination X.25 Address               |           |                          |
| Metric                                 |           |                          |
| SAVE                                   | CANCEL    |                          |

### More Info



- Depending on how you have set up X.25 routing, you can test your X.25 configurations using minipad. See [Minipad, page 238](#). In Germany, call the local echo port to verify X.25 calls can reach the X.25 network with: `minipad 45911029002`.

- Or, if you have more than 1 virtual channel available, you can also place a call to your own router's X.25 address with: `minipad your_router's X.25 address.`
- The call should go out one virtual channel, and come back in on a second virtual channel and you should receive a new login prompt. This can be verified by displaying the `x25CallTable` from the shell, or in Setup Tool under **MONITORING AND DEBUGGING** ➤ **X.25 MONITOR**.

### 6.3.3 How do I Configure X.25 Access for a Host on my LAN?

LAN hosts can utilize X.25 WAN links provided by the router to connect to remote X.25 hosts. The appropriate WAN links should already be configured. This section describes how to configure the LLC link (X.25 over ethernet), the local portion of the end-to-end communication link. An LLC link is specific to a particular LAN host.

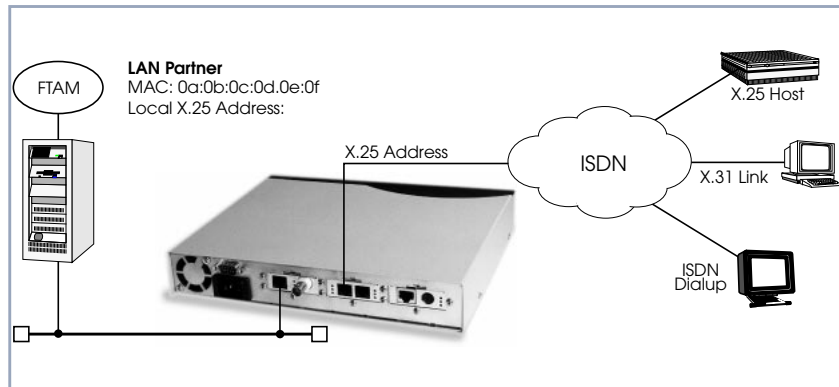


Figure 6-11: Configuration of the LLC Link



### Before you begin,

you will need the following information:

- The router's X.25 address.
- The LAN partner's MAC address.
- A locally assigned X.25 address for the LAN partner.

### Configure X.25 Local Address

- Go to **X.25** ➤ **STATIC SETTINGS**.
- First, verify the router's local X.25 address is configured, e.g. **22**.

|                                      |                          |
|--------------------------------------|--------------------------|
| BRICK Setup Tool                     | BinTec Communications AG |
| [X.25][STATIC]: X.25 Static Settings | MyBRICK                  |
| Local X.25 Address                   | 22                       |
| SAVE                                 | CANCEL                   |
| Enter string, max length = 35 chars  |                          |

### Create LAN Host Link

- Go to **LINK CONFIGURATION**.
- Create a new link for the host on the router's LAN.
- Select the appropriate link template from the list depending on which LAN this host is on.  
Ethernet templates have the format: en<slot>-llc (create new configuration).
- Mark the entry and press **Return** to configure the link. For ethernet links the following settings should be acceptable:

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                          |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| BRICK Setup Tool                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | BinTec Communications AG |
| [X.25][LINK][ADD]: X.25 Link Configuration                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | MyBRICK                  |
| <pre> Link L3 Mode                                dce L3 Packet Size                          1024 bytes L3 Window Size                            5 Window size/Packetsize Neg.              when necessary (default)  Lowest Two-Way-Channel (LTC)             1 Highest Two-Way-Channel (HTC)            4095 Partner MAC Address (LLC)                 &lt;LAN Partner's MAC address&gt;  L2 Window Size Layer 2 Behavior                          disconnect when idle                                  SAVE                                CANCEL </pre> |                          |
| Use <Space> to select                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                          |

An X.25 (LLC) link now exists for our LAN host. You may need to verify the Packet and Window sizes and the number of Virtual Channels for this link are compatible with the settings used on the LAN host.

- Edit X.25 Routing Table** ➤ Go to **ROUTING** ➤ **ADD**.
- Create an X.25 route that says: give incoming calls from this LAN Partner that are addressed to the router's X.25 address to the special local interface (see for information [Special Note: The X.25 Local Interface, page 185](#)).

|                                        |                         |                          |
|----------------------------------------|-------------------------|--------------------------|
| BRICK Setup Tool                       |                         | BinTec Communications AG |
| [X.25][Routing][ADD]: X.25 Route Table |                         | MyBRICK                  |
| Source Link                            | en1-llc                 |                          |
| Destination Link                       | local                   |                          |
| Destination X.25 Address               | <router's X.25 address> |                          |
| Metric                                 |                         |                          |
|                                        | SAVE                    | CANCEL                   |
| Use <Space> to select                  |                         |                          |

- Another Route...** ➤ Create another route so that X.25 calls addressed to our LAN host find the correct link. This route says: all X.25 calls received from the local interface that are addressed to our LAN host should be routed to the host at MAC address over the ethernet link.

|                                        |                              |                          |
|----------------------------------------|------------------------------|--------------------------|
| BRICK Setup Tool                       |                              | BinTec Communications AG |
| [X.25][Routing][ADD]: X.25 Route Table |                              | MyBRICK                  |
| Source Link                            | local                        |                          |
| Destination Link                       | en1-llc                      |                          |
| Destination Link Address (LLC)         | <LAN Partner's MAC address>  |                          |
| Destination X.25 Address               | <LAN Partner's X.25 address> |                          |
| Metric                                 |                              |                          |
|                                        | SAVE                         | CANCEL                   |
| Use <Space> to select                  |                              |                          |

### More Info



Depending on how you have set up X.25 routing, you can test your X.25 configurations using minipad. See [Minipad, page 238](#).

### 6.3.4 How do I Configure ISDN Dialup Access for an X.25 Partner?

This section describes how to configure an ISDN dialup access for an X.25 partner. Here an available ISDN B-channel will be used to transfer X.25 user data with this remote host.

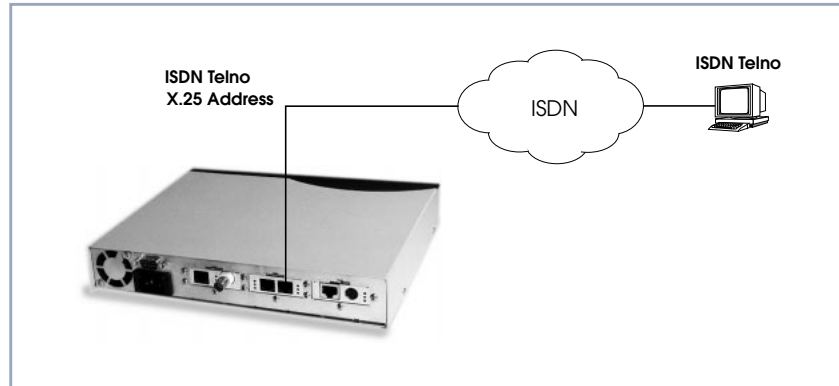


Figure 6-12: ISDN Dialup Access



#### Before you begin,

you will need the following information:

- The router's ISDN telephone number and X.25 address.
- The remote X.25 partner's ISDN telephone number.

#### Configure X.25 Local Address

- Go to **X.25** ➤ **STATIC SETTINGS**.
- Verify the router's X.25 address is set here.

#### Edit WAN Partner

- Go to **WAN PARTNER** ➤ **ADD**.
- Create a new WAN partner interface and enable X.25 traffic.

```

BRICK Setup Tool BinTec Communications AG
[WAN][ADD]: Configure WAN Partner MyBRICK

Partner Name

Encapsulation X.25
Compression none
Encryption none
Calling Line Identification no

PPP>
Advanced Settings>
WAN Numbers>

IP>
IPX>

 SAVE CANCEL

Use <Space> to select

```

- Go to **WAN NUMBERS** ➤ **ADD**.
- Set the partner's ISDN number as WAN Number:

```

BRICK Setup Tool BinTec Communications AG
[WAN][ADD][WAN NUMBERS]: WAN Numbers () MyBRICK

Number <the X.25 partner's ISDN telephone number>
Direction both

Advanced Settings>

 SAVE CANCEL

Use <Space> to toggle

```



If the remote site is another BinTec router verify the Incoming Call Answering settings configured there to ensure this number will be dispatched to the routing service.

Return to the previous menu and select **SAVE**.

### 6.3.5 How do I Configure X.25 Dialout Without Configuration?

In an X.25 network there is often a large amount of connection partners. Because the number of X.25 partners can theoretically be infinite, there is the possibility to configure dial-out to X.25 partners without configuring the partners individually.

For outgoing X.25 calls a feature is implemented, which generates a ISDN number out of the destination X.25 address or the destination NSAP (Network Service Access Point).



#### Before you begin,

you will need the following information:

- The router's ISDN telephone number and X.25 address.

#### Configure X.25 Local Address

- Go to **X.25** ➤ **STATIC SETTINGS**.
- Verify the router's X.25 address is set here (optional).

#### Edit WAN Partner

- Go to **WAN PARTNER** ➤ **ADD**.
- Create a new WAN partner interface and enable X.25 without configuration:



|                                   |                                      |                          |
|-----------------------------------|--------------------------------------|--------------------------|
| BRICK Setup Tool                  |                                      | BinTec Communications AG |
| [WAN][ADD]: Configure WAN Partner |                                      | MyBRICK                  |
| Partner Name                      |                                      |                          |
| Encapsulation                     | X.25 No Configuration, No Signalling |                          |
| Compression                       | none                                 |                          |
| Encryption                        | none                                 |                          |
| Calling Line Identification       | no                                   |                          |
| PPP>                              |                                      |                          |
| Advanced Settings>                |                                      |                          |
| WAN Numbers>                      |                                      |                          |
| IP>                               |                                      |                          |
| IPX>                              |                                      |                          |
| Bridge>                           |                                      |                          |
|                                   | SAVE                                 | CANCEL                   |
| Use <Space> to select             |                                      |                          |

**The following steps must be configured via the SNMP shell in the MIB, because the necessary variables cannot be configured via the Setup Tool:**

**x25RouteTable** By adding the new WAN partner as described before, a new interface was created.

In the **x25RouteTable** now a route for this new interface must be defined.

Example:

```
inx SrcIfIndex(*rw) SrcLinkAddr(rw) DstIfIndex(*rw)
 DstLinkAddr(rw) DstLinkAddrMode(-rw) SrcAddr(rw)
 SrcNSAP(rw) DstAddr(rw) DstNSAP(rw)
 ProtocolId(rw) CallUserData(rw) RPOA(rw)
 NUI(rw) RewritingRule(rw) Metric(rw)
 Cug(rw) CugOutgoing(rw) CugBilateral(rw)

00 1 10008
 rule
 "*11499119673123"
 -1 -1
 8 0
 -1 -1
```

■ For the variables **SrcAddr** and **DestAddr** you can use wildcards.

- The variable **DstLinkAddrMode** can be set to *auto* or *rule*.  
When set to *auto* the **BRICK** can generate the destination ISDN number automatically. A requirement for this function is that the X.25 address contains the ISDN number conform to the (extended) X.121 address format.



### X.121 Address Format

When the extended X.121 address format is used for the destination X.25 address contained in the X.25 call packet, the **BRICK** assumes that the address starts with an “@” followed by a “0” (TOA) and a “1” (NPI for ISDN). These three digits are deleted and the rest of the X.25 address is taken over as the destination ISDN number.

When the normal X.121 address format is used, the **BRICK** looks for a “0” (escape character for ISDN) or a “9” (escape character for analog connections) as the first digit of the X.25 address, deletes this first digit and again takes the rest of the X.25 address as the destination ISDN number.

These conventions are the requirement for using the value *auto* in the variable **DstLinkAddrMode**.

In case the ISDN number is not contained in the X.25 address of the call packet, the generation of the destination ISDN number must be defined via a rule like explained in the following.

You can set the variable **DstLinkAddrMode** to *rule*. When done so, the variable **RewritingRule** must be assigned an integer from 0 to 999999, which is the number of the rewriting rule used. Then you must generate an entry in the **x25RewriteTable** with this rewriting rule number.

**x25RewriteTable** The rule for converting the destination X.25 address respectively NSAP into an ISDN number is defined in the variable **dstLinkAddr** of the **x25RewriteTable**. This table contains table entries, which each belong to one rewriting rule number (variable **RewritingRule**). These numbers are referenced in the **x25RouteTable** described earlier.

**Example:**

```

inx RewritingRule(*rw)ReverseCharging(-rw) RPOA(rw)
 NUI(rw) SrcAddr(rw) SrcNSAP(rw)
 DstAddr(rw) DstNSAP(rw) ProtocolId(rw)
 CallUserData(rw) RespSrcAddr(rw) RespSrcNSAP(rw)
 RespDstAddr(rw) RespDstNSAP(rw) RespProtocolId(rw)
 RespCallUserData(rw) Cug(rw) CugOutgoing(rw)
 CugBilateral(rw) DstLinkAddr(rw)

00 8 dont_change dont_change

 -1

 -1

 -1

-1 -1 "X%%00.....%%456"

```

The format of the variable **dstLinkAddr** consists of the following components:

## [Layer 1/Address Type] Input Rule

- Layer 1/Address Type
  - This part of the variable **dstLinkAddr** is optional.
  - When nothing is defined “data\_64k” is used as default.

| Part of dstLinkAddr | Meaning        |
|---------------------|----------------|
| 1                   | analog (modem) |
| 2                   | V110_9600      |
| 3                   | MAC address    |
| 4                   | IP address     |

Table 6-13: Part of **dstLinkAddr**

- Input
  - This part of the variable **dstLinkAddr** is mandatory.
  - It defines whether the input for the conversion is an X.25 address or a NSAP.

| Part of <b>dstLinkAddr</b> | Meaning      |
|----------------------------|--------------|
| X                          | X.25 address |
| N                          | NSAP         |

Table 6-14: Part of **dstLinkAddr**

■ Rule

- This part of the variable **dstLinkAddr** is mandatory.

| Part of <b>dstLinkAddr</b> | Meaning                        |
|----------------------------|--------------------------------|
| .                          | take over one digit            |
| %                          | delete one digit               |
| *                          | take over the remaining digits |
| 0-9                        | insert digits                  |

Table 6-15: Part of **dstLinkAddr**

Examples:

| Rule                        | X.25 Address/NSAP | ISDN Number /<br>MAC Address / IP<br>Address |
|-----------------------------|-------------------|----------------------------------------------|
| X%%%%%%%%00.....%%%%%%%%456 | @11499119673123   | 009119673456                                 |
| X%%%%%%%%00.....4*          | @11499119673123   | 0091196734123                                |
| N%%00.....4*                | 499119673123      | 0091196734123                                |
| 3X%%%*                      | @5200a0f9000123   | 00:a0:f9:00:01:23                            |
| 4X%%%*                      | @53c03635a0       | 192.54.53.160                                |

Table 6-16: Examples

### 6.3.6 How do I Route IP Traffic over X.25 with MPX25?

The router can be configured to route multiple protocols (IP, IPX, and Bridging) over X.25. This mechanism allows you to use existing X.25 links as the transport medium for routing other protocols. We call these interfaces MPX25 for short. We assume that the X.31 link has already been configured and that the appropriate routes are set. (Configuring different X.25 links are described beginning with [How do I Configure an X.31 Link \(X.25 in the D-Channel\)?](#), page 186.)

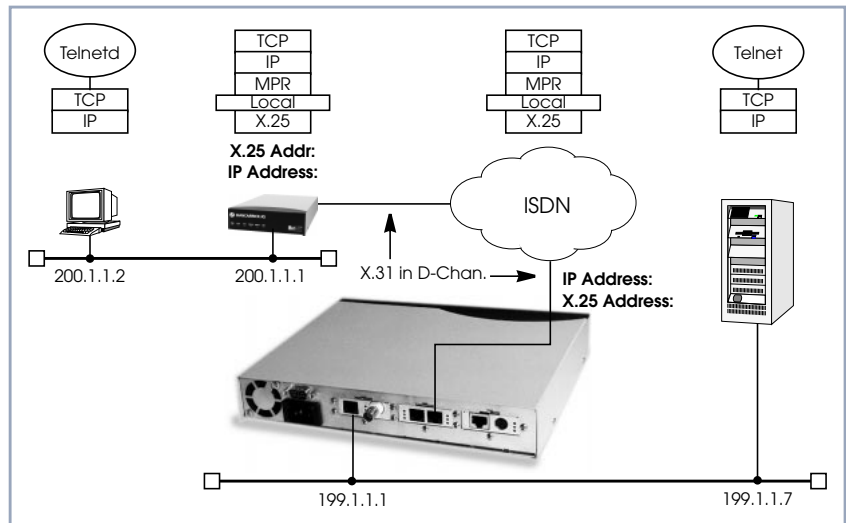


Figure 6-13: Routing over X.25



#### Before you begin,

you will need the following information:

- The router's X.25 address.
- The remote partner's X.25 address.
- The remote partner's IP address.

### Configure a New MPX Partner

- Go to **X.25** ➤ **MULTIPROTOCOL OVER X.25** ➤ **ADD**.
- Create a new MPX25 interface for the remote X.25 partner. Here we define the types of traffic (IP, IPX, and Bridge) to transport over this link. For our example earlier, we will only route IP.
- Select an **Encapsulation**, e.g. **ip\_rfc877**.

|                                             |                                |                          |
|---------------------------------------------|--------------------------------|--------------------------|
| BRICK Setup Tool                            |                                | BinTec Communications AG |
| [X.25][MPR][ADD]:Configure X.25 MPR Partner |                                | MyBRICK                  |
| Partner Name                                |                                |                          |
| Encapsulation                               | ip_rfc877                      |                          |
| X.25 Destination Address                    | <MPX25 partner's X.25 address> |                          |
| Advanced Settings>                          |                                |                          |
| IP>                                         |                                |                          |
| IPX>                                        |                                |                          |
| SAVE                                        |                                | CANCEL                   |
| Enter string, max length = 25 chars         |                                |                          |



Only if an X.31 in D-channel link is being used as the transport medium, the X.25 address entered here should be preceded by {00}. This will allow outgoing calls to be placed correctly (using: 00<country code><area code><local number>) and incoming calls to be identified (the X.25 network delivers calls without the preceding 00).

- Edit the protocol-relevant settings for this partner. In our example, we are routing IP over X.25, so we need to set the remote partner's IP address here.
- Go to **IP**.

|                                            |                              |                          |
|--------------------------------------------|------------------------------|--------------------------|
| BRICK Setup Tool                           |                              | BinTec Communications AG |
| [X.25][MPR][ADD][IP]: IP Configuration ( ) |                              | MyBRICK                  |
| IP Transit Network                         | yes                          |                          |
| local ISDN IP Address                      | 172.16.98.91                 |                          |
| Partner's ISDN IP Address                  | <MPX25 partner's IP address> |                          |
| Partner's LAN IP Address                   |                              |                          |
| Partner's LAN Netmask                      |                              |                          |
| Advanced Settings>                         |                              |                          |
| SAVE                                       |                              | CANCEL                   |



### More Info

Depending on how you have set up X.25 routing, you can test your X.25 configurations using minipad. See [Minipad, page 238](#).

## 6.3.7 How do I Use the Router as a TCP-X.25 Bridge?

The router can be used as a TCP-X.25 bridge as described in RFC 1086.

Using this mechanism, the router can be used to allow X.25 and TCP hosts to communicate by providing an end-to-end ISO-TP0 connection.

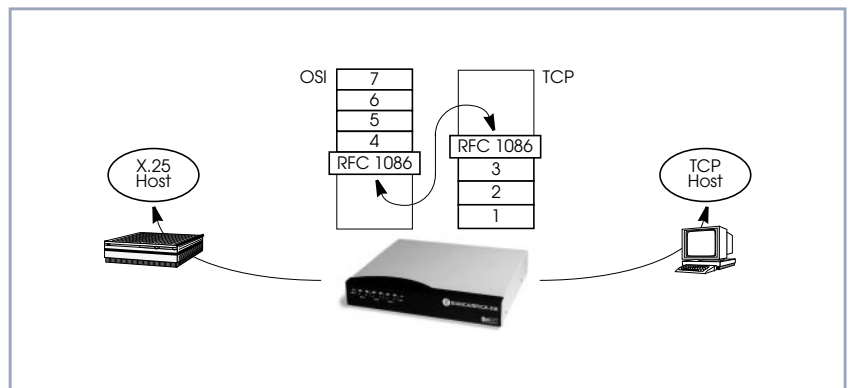


Figure 6-14: The Router as a TCP-X.25 Bridge

Depending on which side initiates the connection (see the examples under [More Info, page 189](#)) the router performs the appropriate protocol mappings as shown earlier.



### Before you begin:

No special information is required to configure the router as an ISO-TP0 bridge. Please note, however, that TCP clients must support RFC 1006 which describes how to transmit TP0 packets over TCP.

### Verify License

- Go to **LICENSES**.
- Verify your X.25 license. You should see **X.25(valid)** in this menu.

### Route for outgoing calls

- Go to **X.25** ➤ **ROUTING** ➤ **ADD**.
- X.25 routing must be configured so that incoming and outgoing calls can be established. Using the special local interface (see [Special Note: The X.25 Local Interface, page 185](#)) a minimum X.25 routing setup could be used as follows:

|                                        |                          |
|----------------------------------------|--------------------------|
| BRICK Setup Tool                       | BinTec Communications AG |
| [X.25][Routing][ADD]: X.25 Route Table | MyBRICK                  |
| Source Link                            | local                    |
| Destination Link                       | x31d2-0-1                |
| Destination X.25 Address               |                          |
| Metric                                 | 0                        |
| SAVE                                   | CANCEL                   |
| Use <Space> to select                  |                          |

| Possible Values | Meaning                                                                                                                                                                                                                  |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| x31d2-0-1       | Use an available X.25 compatible interface name here. By default interfaces for ISDN: x31d- <i>&lt;slot #&gt;</i> - <i>&lt;unit #&gt;</i> - <i>&lt;TEI&gt;</i> and X.21 modules: xi <i>&lt;slot #&gt;</i> are available. |

Table 6-17: **Destination Link**



- Route for Incoming Calls**
- Go to **X.25** ➤ **ROUTING** ➤ **ADD**.
  - Create another route for incoming calls. The interface name used in the **Source Link** field should be the same interface used in the previous step.

|                                        |      |                          |        |
|----------------------------------------|------|--------------------------|--------|
| BRICK Setup Tool                       |      | BinTec Communications AG |        |
| [X.25][Routing][ADD]: X.25 Route Table |      | MyBRICK                  |        |
| Source Link                            |      | x31d2-0-1                |        |
| Destination Link                       |      | local                    |        |
| Destination X.25 Address               |      |                          |        |
| Metric                                 |      | 0                        |        |
|                                        | SAVE |                          | CANCEL |
| Use <Space> to select                  |      |                          |        |



### More Info

Two common uses for this mechanism are as follows:

- TCP Client requests connection to X.25 Server
- X.25 Client requests connection to TCP Server

For more detailed reference please refer to RFCs 1006 and 1086 respectively.

1. TCP Client requests connection to X.25 Server

Here the TCP-Client initiates a connection (as defined in RFC 1086) with the router using TCP port 146. The router then contacts the remote X.25-Server

and transparent TPO packets can begin to be exchanged between the two endpoints.

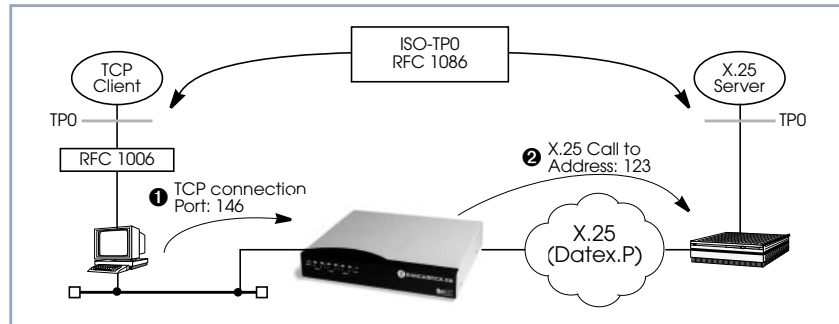


Figure 6-15: TCP Client requests connection to X.25 Server

## 2. X.25 Client requests connection to TCP Server

Here the TCP-Server must first initiate a connection with the router at TCP port 146 where it registers its IP address and port number. It instructs the router to accept incoming calls addressed to an X.25 address (123) and route the connection to the registered TCP port number (6002) and IP address (10.5.5.5).

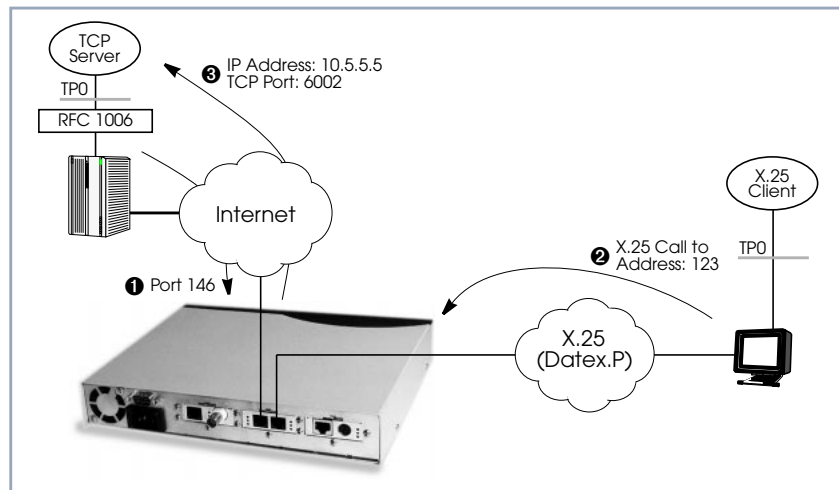


Figure 6-16: X.25 Client requests connection to TCP Server



The router will listen for incoming calls to the registered address only as long as the TCP (port 146) connection between the registering host and the router exists.

### 6.3.8 How do I Configure the Routing for Using an X.25 PAD?

To configure the X.25 PAD utility the ISDN interface configuration must be extended and a new software interface for the X.25 PAD must be created.



#### Before you begin:

Before you start you will need the following information:

The X.25 PAD's unique MSN (Multiple Subscriber Number)

The remote X.25 network partner's name and possibly X.25 address

#### Configure Hardware Interface

- Go to **CM-1BRI, ISDN S0** ➤ **INCOMING CALL ANSWERING** ➤ **ADD**.
- Create a new entry for incoming calls on the ISDN interface to be routed to the X.25 PAD.

|                                                          |                  |                          |        |
|----------------------------------------------------------|------------------|--------------------------|--------|
| BRICK Setup Tool                                         |                  | BinTec Communications AG |        |
| SLOT 2 ISDN BRI][INCOMING][ADD]: Incoming Call Answering |                  | MyBRICK                  |        |
| Item                                                     | X.25 PAD         |                          |        |
| Number                                                   | <X.25 PAD's MSN> |                          |        |
| Mode                                                     | right to left    |                          |        |
| Username                                                 |                  |                          |        |
| Bearer                                                   |                  |                          |        |
|                                                          | SAVE             |                          | CANCEL |
| Use <Space> to select                                    |                  |                          |        |

- #### Edit WAN Partner
- Next you must add the *X.25 PAD* as a new WAN partner. Because the X.25 PAD's WAN partners can not be identified by their caller's numbers, you must create one WAN Partner.

- Go to **WAN PARTNER** ➤ **ADD**.
- Create a new WAN partner interface:

|                                   |                           |
|-----------------------------------|---------------------------|
| BRICK Setup Tool                  | BinTec Communications AG  |
| [WAN][ADD]: Configure WAN Partner | MyBRICK                   |
| Partner Name                      | <X.25 PAD's partner name> |
| Encapsulation                     | X.25 PAD                  |
| Compression                       |                           |
| Encryption                        |                           |
| Calling Line Identification       |                           |
| PPP>                              |                           |
| Advanced Settings>                |                           |
| WAN Numbers>                      |                           |
| IP>                               |                           |
| IPX>                              |                           |
| SAVE                              | CANCEL                    |
| Use <Space> to select             |                           |

- Create X.25 PAD Link**
- Go to **X.25** ➤ **LINK CONFIGURATION**.
  - Create a new link for the X.25 PAD's partner.
  - Select the appropriate link template from the list:

|                                                                        |                          |
|------------------------------------------------------------------------|--------------------------|
| BRICK Setup Tool                                                       | BinTec Communications AG |
| [X.25][LINK]: X.25 Link Configuration                                  | MyBRICK                  |
| Select Link to configure                                               |                          |
| <X.25 PAD's partner name> (create new configuration)                   |                          |
| DELETE CONFIGURATION                                                   | EXIT                     |
| Press <Ctrl-n>, <Ctrl-p> to scroll, <Space> tag/untag DELETE, <Return> |                          |

- Edit the items and change them, if necessary. You might e.g. want to configure special values for **L3 Packet Size**, **L3 Window Size** or **Window size/ Packetsize Neg**.
- In general the default values you will find in this menu do not have to be changed. But even, if you do not make any changes you must leave the

menu with **SAVE** to configure the Link Configuration for the X.25 PAD Partner.

**Edit X.25 Routing Table** Depending on whether you want to define a static route from the X.25 PAD's partner interface to a single X.25 host/remote partner or multiple routes between several X.25 partners, the routing information differs.

1. Routing configuration for a static routing between two X.25 partners (the X.25 PAD's partner and a remote X.25 host/partner).

➤ Go to **X.25** ➤ **ROUTING** ➤ **ADD**.

➤ Create an X.25 route that routes outgoing calls from the X.25 PAD to the remote X.25 network partner (X.25 host).

|                                        |                             |
|----------------------------------------|-----------------------------|
| BRICK Setup Tool                       | BinTec Communications AG    |
| [X.25][Routing][ADD]: X.25 Route Table | MyBRICK                     |
| Source Link                            | <X.25 PAD's partner name>   |
| Destination Link                       | <X.25 network partner name> |
| Destination X.25 Address               |                             |
| Metric                                 |                             |
| SAVE                                   | CANCEL                      |
| Use <Space> to select                  |                             |



The partner used in the Destination Link must be configured before as an X.25 partner.

2. This second configuration is an example for connecting three X.25 partners, one of them the X.25 PAD's partner.

|                                        |                               |                          |
|----------------------------------------|-------------------------------|--------------------------|
| BRICK Setup Tool                       |                               | BinTec Communications AG |
| [X.25][Routing][ADD]: X.25 Route Table |                               | MyBRICK                  |
| Source Link                            | <X.25 PAD's partner name>     |                          |
| Destination Link                       | <X.25 network partner name A> |                          |
| Destination X.25 Address               | 1*                            |                          |
| Metric                                 |                               |                          |
|                                        | SAVE                          | CANCEL                   |
| Use <Space> to select                  |                               |                          |

|                                        |                               |                          |
|----------------------------------------|-------------------------------|--------------------------|
| BRICK Setup Tool                       |                               | BinTec Communications AG |
| [X.25][Routing][ADD]: X.25 Route Table |                               | MyBRICK                  |
| Source Link                            | <X.25 PAD's partner name>     |                          |
| Destination Link                       | <X.25 network partner name B> |                          |
| Destination X.25 Address               | 2*                            |                          |
| Metric                                 |                               |                          |
|                                        | SAVE                          | CANCEL                   |
| Use <Space> to select                  |                               |                          |



The partners used in the Destination Link must be configured before as X.25 partners.



### More Info

For further information see [X.25 PAD, page 215](#).

## 6.4 X.25 Utilities

### 6.4.1 X.25 PAD

**General** The PAD is a data assembly/disassembly facility used to connect character-oriented asynchronous data terminal equipment (DTE) to the packet-oriented X.25 network (Datex-P). It is the task of PAD to convert character streams coming from the DTE into data packets and resolve data packets coming from the network into individual character streams that can be displayed on the DTE. In this context the character-oriented data terminal equipment is also called start-stop mode DTE (short: DTE) and a remote X.25 host is defined as packet mode DTE.

Recommendation X.29 defines the procedures between a PAD and a packet-mode DTE or another PAD and recommendation X.28 defines the DTE interface of a start-stop mode DTE accessing the PAD.

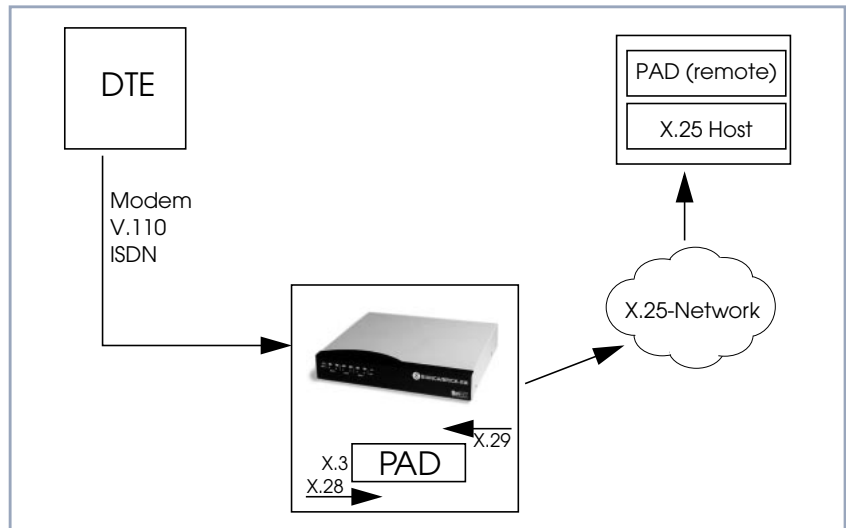


Figure 6-17: X.25 PAD

The PAD program is an implementation of the X.25 PAD according to the three following ITU-T recommendations:

- X.3: Parameter definition
- X.28: User interface / commands
- X.29: PAD to PAD protocol

In each case, the standard of 1988 is implemented. The implementation should, however, also be compatible to earlier versions.

PAD features one command mode and one data transfer state. The commands are described below. PAD can manage only exiting calls, it cannot be called itself.

PAD command signals are directed from the DTE to the PAD and are described under [Commands Conforming to X.28, page 229](#). PAD service signals are directed from the PAD to the DTE and serve for e.g acknowledging PAD commands and or transmitting call progress signals to the DTE.

#### Additional Features

There are two additional features built into the PAD to extend the standard X.25 PAD functionality.

One is the additional variable **AutoCallDstAdr** in the **x25PadProfileTable**, which can contain an X.25 address, the PAD automatically establishes a connection to. The value of this variable must be defined in the **x25PadProfileTable** on the **BRICK**.

The second item is a timer that determines, when to close down a connection to the remote X.25 station, after the DTE has sent the CLR command to the PAD. This time period is defined by configuring the X.25 PAD's partner. It results from the sum of the values of two items in Setup Tool: **Static Short Hold** in the **WAN ► EDIT ► ADVANCED** menu (Short Hold in the **biboPPPTTable** of the MIB) and **Disconnect Timeout** in the **X25 ► LINK ► EDIT** menu (**L2IdleTimer** in the **X25LinkPresetTable** of the MIB).

#### PAD Parameters

All PAD parameters are stored in the variables of the **x25PadProfileTable** on the **BRICK** and can be edited there.



**Additional Entries**

| Additional Entries     | Meaning                                                                                                                                                                                                                                                                                                                                      |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Number</b>          | <p>The value of this parameter defines the unique number of the PAD Profile.</p> <p>Possible values:</p> <p>0-99: <b>PadProfileTable</b> numbers.</p> <p>The <b>PadProfileTables</b> 0, 90 and 91 (see later) are implemented in the <b>BRICK</b>.</p>                                                                                       |
| <b>State</b>           | <p>This parameter describes the state of the profile.</p> <ol style="list-style-type: none"> <li>1. The Profile is valid. (<b>valid</b>)</li> <li>2. The Profile is set to delete. (<b>delete</b>)</li> </ol> <p>The default value is 1 resp. <i>valid</i>.</p>                                                                              |
| <b>AutoCallDstAddr</b> | <p>When this parameter is set to a non-empty string, a call will automatically be established to this PAD address.</p> <p>By default this variable is empty. To activate the autocal function the user must enter a value (valid X.25 address) for this variable in the <b>x25PadProfileTable</b> (described later) on the <b>BRICK</b>.</p> |

Table 6-18: Additional Entries

**Standard Parameters** The 22 standard PAD parameters defined in X.3 are listed in the table:

| Number | Parameter   | Description                                |
|--------|-------------|--------------------------------------------|
| 1      | Escape      | PAD recall using a character               |
| 2      | Echo        | Echo                                       |
| 3      | ForwardChar | Selection of the data forwarding character |
| 4      | IdleTimer   | Selection of idle timer delay              |
| 5      | DevControl  | Ancillary device control                   |
| 6      | SigControl  | Control of PAD service control             |
| 7      | BrkControl  | Operation on receipt of the break signal   |
| 8      | Discard     | Discard output                             |
| 9      | CRPadding   | Padding after carriage return              |
| 10     | LineFold    | Line Folding                               |
| 11     | Speed       | Binary speed (read only)                   |
| 12     | FlowControl | Flow control of the PAD                    |
| 13     | LFInsert    | Linefeed insertion after carriage return   |
| 14     | LFPadding   | Padding after linefeed                     |
| 15     | Edit        | Editing                                    |
| 16     | CharDel     | Character delete                           |
| 17     | LineDel     | Line delete                                |
| 18     | LineDisp    | Line display                               |
| 19     | SigEdit     | Editing PAD service signals                |
| 20     | EchoMask    | Echo mask                                  |
| 21     | Parity      | Parity treatment                           |
| 22     | PageWait    | Page wait                                  |

Table 6-19: Standard Parameters

The exact meanings of the individual parameters and their possible values are described in the following sections; **^X** stands for the simultaneously pressing the **Control** key (also **Ctrl**) and the **X** key; terms such as **BEL** or **ACK** refer to the corresponding characters in the International Alphabet No. 5 (IA5) according to ITU-T T.50.

## Standard Parameters and their Meaning

**1 Escape** Definition of a character which causes PAD to switch from the data transfer to the command mode (escape character).

Possible values:

- 0: It is not possible to leave the data transfer state.
- 1: Leave the data transfer state with **^P**.
- 32-126: Defines the character of the IA5 with the number specified as escape character.

The default value is *0*.

If a connection exists, the PAD automatically switches back to the data transfer state after input of a valid command. An exception is the clear command.

**2 Echo** Defines whether the echo mode is enabled or not.

Possible values:

- 0: The echo mode is disabled; no echo. (**no\_echo**).
- 1: The echo mode is enabled. (**echo**).

The default value is *0* resp. *no\_echo*.

Specifies whether an echo is to be created by the PAD or not.

Using parameter 20, **EchoMask**, specific characters can be exempted from the echo mode.

**3 ForwardChar** Definition of characters upon which the PAD forwards the data entered up to that point as a packet (data forwarding character).

Possible values:

- 0: No data forwarding character assigned.

- 1: The characters <A>-<Z>, <a>-<z>, and <0>-<9> serve as data forwarding characters.
- 2: Data forwarding via activation of the **Return** key (IA5 character 0/13, CR).
- 4: Data forwarding after input of either ESC, BEL, ENQ or ACK.
- 8: Data forwarding after input of either DEL, CAN or DC2.
- 16: Data forwarding after input of either EOT or EXT.
- 32: Data forwarding after input of either HT, LF, VT or FF.
- 64: All characters in columns 0 and 1 of the IA5 not specified above serve for data forwarding.

The default value is 0.

These values correspond to the individual bits in the 1-byte value that can be assigned to this parameter. The values can also be freely combined, e.g.:

- 126: All characters of columns 0 and 1 of the IA5 and the character 7/15, DEL serve for data forwarding (combination of the values 2+4+8+16+32+64).

Using the national parameters 121 and 122, another data forwarding character can be defined for each of them. Data forwarding takes place additionally via the BREAK signal and timer delay in the PAD (parameter 4, IdleTimer).

**4 Idle Timer** Defines whether after a specific amount of time all data entered up to this point are to be forwarded as a packet.

Possible values:

- 0: No timer-controlled data forwarding.
- 1-255:  $n \cdot 50\text{ms}$  after the last input of a character, the data entered up to that point are forwarded as a packet.

The default value is 5 (= 250 ms).

The parameter value  $n$  indicates the delay time as a multiple of 50 ms, thus times of up to approx. 12s are possible.

If parameter 15, Edit, is set to 1, timer-controlled data forwarding is disabled.

**5 DevControl** Defines use of the characters DC1 and DC3 for the control of ancillary devices.

Possible values:

- 0: No use of DC1 and DC3. (*no\_use*)

DC1 corresponds to X-ON or **^Q**, DC3 corresponds to X-OFF or **^S**.

**6 SigControl** Defines whether, and if so how, PAD service signals are forwarded to the DTE.

Possible values:

- 0: X.28 mode without PAD service signals.

- 1: X.28 messages are transmitted to the DTE.

5: X.28 messages are transmitted to the DTE, additionally a prompt (\*) is output in the command mode.

The default value is 1.

**7 BrkControl** Defines the reaction of the PAD to the reception of the BREAK signal from the start-stop mode DTE in data transfer state.

Possible values:

- 0: No reaction.

- 1: Data forwarding, an interrupt packet is transmitted, the PAD remains in data transfer state.

- 2: Data forwarding, the virtual connection is reset with possible data loss, the PAD remains in data transfer state.

- 4: Send an "indication of break" PAD message to the packet-mode DTE (remote PAD).

- 5: Send an interrupt packet followed by an "indication of break" PAD message to the packet-mode DTE.

- 8: Data forwarding, switch to command mode.

- 16: Discard output data to the DTE.

- 21: Discard all output data to the DTE, data forwarding, send an interrupt packet and the PAD service signal BREAK indication with parameter field in which parameter 8 is set to 1, the PAD remains in data transfer state.

The default value is 8.

If no connection has been established, the BREAK signal is ignored.

The BREAK signal is not a character of the IA5. It always consists of an approx. 150 ms long continuous string of the level for binary 0.

Receiving a BREAK signal is a requirement for packet forwarding by the PAD except for parameter 7 is set to 0.

**8 Discard** Defines whether user sequences in packets are output to the DTE or not.

If parameter 7 is set to 21, parameter 8 is set to 1 when a BREAK signal is received. From now on, all data outputs to the DTE are ignored until parameter 8 is reset to 0.

Possible values:

- 0: Normal data output to the DTE. (*normal\_data\_delivery*).
- 1: Data outputs to the DTE are ignored. (*discard\_output*).

The default value is 0 resp. *normal\_data\_delivery*.

**9 CRPadding** Defines the number of padding characters (NUL) generated after a CR to the DTE.

This parameter has meaning only for purely mechanical DTE (e.g. teletyper - it bridges the time required for the actual carriage return. For modern DTE this parameter is unnecessary, sometimes even interferes (e.g. with direct storing of data in a file).

Possible values:

- 0: No padding characters.
- 1–255: Number of padding characters (NUL) - only useful for purely mechanical DTE.

The default value is 0.

This parameter is only used upon PAD service signals.

**10 LineFold** Defines the number of characters after which automatic line folding (inserting the character CR) is to take place.

Possible values:

- 0: No automatic line folding.

Depending on the settings of parameters 13 or 126, LF is inserted in addition to CR.

**11 Speed** Defines the transmission speed of the DTE. This parameter is set automatically by the PAD. The parameter is only used internally and not listed in the **x25PadProfileTable**. The possible values are described in ITU X.3.

**12 FlowControl** Defines whether the user can effect a short-time stop (DC3) and restart (DC1) of the data flow to the DTE via input of the control characters DC1 and DC3.

Possible values:

- 0: No use of DC1 and DC3 for data flow control. (*no\_use\_DC1\_DC3*).

DC1 corresponds to X-ON or ^Q, DC3 corresponds to X-OFF or ^S.

**13 LFInsert** Defines whether the PAD inserts a LF after receiving CR.

Possible values:

0: No LF insertion.

1: LF insertion after each CR in the data stream to the start-stop mode DTE.

2: LF insertion after each CR from the start-stop mode DTE.

4: LF insertion after each CR in the echo stream to the start-stop mode DTE.

5: Combination of 1 and 4.

6: Combination of 2 and 4.

7: Combination of 1, 2 and 4.

The default value is 0.

This parameter is only applied in data transfer mode.

**14 LFPadding** Defines the number of padding characters (NUL) which are output after an LF to the DTE.

Possible values:

- 0: No padding characters.

**15 Edit** Defines whether editing of user data is possible in data transfer state or not. If parameter 15 is set to 1, parameter 4 is disabled.

Possible values:

- 0: Editing not possible (*no\_editing\_user\_data*).
- 1: Editing possible (*editing\_user\_data*).

The default value is 0 resp. *no\_editing\_user\_data*.

### **16 CharDel**

Defines whether it is possible to delete characters already entered and which character is used for this function.

Possible values:

- 0–127: Decimal value of the character from the IA5 to be used for character delete.

The default value is 0.

### **17 LineDel**

Defines whether it is possible to delete a line already entered and which character is to be used for this function.

Possible values:

- 0–127: Decimal value of the character from the IA5 to be used for character delete.

The default value is 0.

### **18 LineDisp**

Defines whether the characters entered and not yet forwarded can be output again on the DTE and which character is to be used for this function.

Possible values:

- 0–127: Decimal value of the character from the IA5 to be used for character delete.

The default value is 0.



**19 SigEdit** Defines which PAD service signals are output after editing (character or line delete).

Possible values:

- 0: No editing PAD service signals.
- 1: Editing PAD service signals for printer; "XXX" is output to confirm line delete, "\" to confirm character delete.
- 2: Editing PAD service signals for display units; characters and lines are deleted visibly on the screen.
- 8, 32-126: Decimal value of the character from the IA5 that is to be output as editing PAD service signal for character delete.

The default value is 0.

**20 EchoMask** Defines which characters are to be exempted from the echo function.

Possible values:

- 0: No echo mask.
- 1: No echo of character CR.
- 2: No echo of character LF.
- 4: No echo of characters VT, HT and FF.
- 8: No echo of characters BEL and BS.
- 16: No echo of characters BEL and BS.
- 32: No echo of characters ACK, NAK, STX, SOH, EOT, ETB and ETX.
- 64: No echo of the editing characters defined in parameters 16, 17 and 18.
- 128: No echo of DEL and of all characters in columns 0 and 1 of the IA5 not mentioned above.

The default value is 0.

Combinations of the given values are permitted.

The echo mask is effective only if parameter 2 is set to 1.

**21 Parity** Defines whether parity bits are checked and/or generated in the PAD.

Possible values:

- 0: No parity bit checking or generation (*no\_parity*).

**22 PageWait** Defines the number of lines (or LF characters) after which the PAD is to interrupt output to the DTE.

Possible values:

- 0: Page wait disabled.

### National Parameters According to Datex-P

If a national parameter is changed, the respective standard parameter is changed also, and vice versa.

| National Parameters                                | Meaning                                                                                                                                                                                                                                                                                                                       |
|----------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>118 XCharDel</b>                                | This parameter is a repetition of parameter 16.<br>The default value is 0.                                                                                                                                                                                                                                                    |
| <b>119 XLineDel</b>                                | This parameter is a repetition of parameter 17.<br>The default value is 0.                                                                                                                                                                                                                                                    |
| <b>120 XLineDisp</b>                               | This parameter is a repetition of parameter 18.<br>The default value is 0.                                                                                                                                                                                                                                                    |
| <b>121 XForwardChar1 and<br/>122 XForwardChar2</b> | Allow the definition of up to two data forwarding characters in addition to parameter 3.<br><br>Possible values:<br><br>■ 0: No additional data forwarding character.<br><br>■ 1–126: Decimal value of the character from the IA5 to be used as data forwarding character.<br><br>The default value for both parameters is 0. |
| <b>123 XParity</b>                                 | Corresponds to parameter 21.<br><br>Possible values:<br><br>■ 0: No parity bit checking or generation ( <i>no_parity</i> ).                                                                                                                                                                                                   |

| National Parameters         | Meaning                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>125 XDelay</b></p>    | <p>Defines how long data forwarding is to be delayed if it occurs simultaneously with a data input.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>■ 0: No delay of data forwarding. Only with full-duplex connections (parameter 2 is set to 1).</li> <li>■ 1–255: Number of seconds by which data forwarding is to be delayed.</li> </ul> <p>The default value is 0.</p> <p>If input editing is possible (parameter 15 is set to 1), a sufficiently large value should be selected for parameter 125 (e.g. 60 seconds) so that incoming data are not written into the data to be edited.</p> <p>Each character entered resets the delay counter to 0. However, after input of an appropriate character, data forwarding starts immediately.</p> |
| <p><b>126 XLFInsert</b></p> | <p>This parameter is a repetition of parameter 13. The default value is 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

Table 6-20: National Parameters and their Meaning

## PAD Commands **Guidelines on Notation**

The PAD understands the commands described below.

The character "↵" stands for pressing the **Return** key (carriage return).

Alternatives are separated by a "|"; for example, "yes|no" means, that either "yes" or "no" can be entered.

Terms in [square brackets] are optional, terms in {curved brackets} are optional and can be repeated any number of times, terms in <angle brackets> must be

replaced by an appropriate character sequence (e.g., <ParNo> stands for a specific parameter number).

Except for the characters {[<|>]} and text in parentheses, all characters of the commands must be entered exactly as indicated in this section.

Upper and lower case letters as well as spaces can be used freely within the commands - internally, lower case letters are converted to upper case letters, spaces are ignored, and the command is executed only after these processes.

The service signals output by the PAD are given here for the standard setting (parameter 6 has the value 1).

### Commands Conforming to X.28

**STAT** ↓ Queries the status of a connection. In response, one of the following messages is given, depending on whether the connection is free or engaged:

- FREE: not connected.
- ENGAGED: connected

**CLR** ↓ Disconnects the selected virtual connection. The command is acknowledged with the message:

- CLR CONF: Disconnect, local cause.

Data that are still in the network when the command is transmitted can be lost.

Within a specified time interval (see [Additional Features, page 216](#)) after a CLR command has been sent, another command can be sent or a new connection can be initiated.

**ICLR** ↓ After having received this command the PAD transmits an "Invitation to clear" to the remote partner, i.e. an "invitation" to disconnect the existing connection.

In all the following commands, possible inputs for <ParNo> are the number of the respective parameter (1-22, 118-123, 125-126).

Generally, only the parameter number is indicated in PAD outputs.

**PAR?** [**<ParNo>**{**<ParNo>**}] ↓ Queries the current values of the parameters indicated or of all parameters if no parameter number is given (here the square brackets indicate that the specification of the parameters is optional).

The parameter values are output as follows:

PAR <ParNo>:<value>>{,<ParNo>:<value>}

If an invalid parameter number was entered for <ParNo>, the following message is output:

PAR <ParNo>:INV.

**RPAR? [<ParNo>{,<ParNo>}]**

Queries the current values of the parameters indicated or of all parameters if no parameter number is given (here the square brackets indicate that the specification of the parameters is optional) of the remote PAD (= the packet-mode DTE). The local PAD won't put out a message until the remote PAD has answered. When the remote PAD answers with the value(s) of the parameter(s), the local PAD puts them out to the start-stop mode DTE.

The parameter values are output as follows:

PAR <ParNo>:<value>>{,<ParNo>:<value>}

If an invalid parameter number was entered for <ParNo>, the following message is output:

PAR <ParNo>:INV.

**SET <ParNo>:<value>{,<ParNo>:<value>}]**

Used for setting the parameter values.

The value ranges for the individual parameters are described in detail in the sections "Additional Entries", "PAD Parameters", "National Parameters".

If an invalid parameter number was entered for <ParNo>, the following message is output:

PAR <ParNo>:INV.

If the parameter number and value entered were valid no confirmation message is put out.

**SET? <ParNo>:<value>{,<ParNo>:<value>}]**

Used for setting and querying the parameter values.

The value ranges for the individual parameters are described in detail in the sections "Additional Entries", "PAD Parameters", "National Parameters".

If an invalid parameter number was entered for <ParNo>, the following message is output:

PAR <ParNo>:INV.

If the parameter number and value entered were valid, the parameters just set are output for checking purposes in the following form:

PAR <ParNo>:<value> {,<ParNo>:<value>}.

**RSET? <ParNo>:<value>{,<ParNo>:<value>}.**

Used for setting and querying the parameter values of the remote PAD. When the local PAD receives this command, it will send a request to set and put out the specified parameters to the remote PAD. The local PAD won't put out a message until the remote PAD has answered. When the remote PAD answers with the value(s) of the parameter(s), the local PAD puts them out to the start-stop mode DTE.

The value ranges for the individual parameters are described in detail in the sections "Additional Entries", "PAD Parameters", "National Parameters".

If an invalid parameter number was entered for <ParNo>, the following message is output:

PAR <ParNo>:INV.

If the parameter number and value entered were valid, the parameters just set are output for checking purposes in the following form:

PAR <ParNo>:<value> {,<ParNo>:<value>}

### Priorities

It is possible and permissible to assign the same value (especially the same character) to different parameters (and thus to the functions controlled by them). If the PAD receives such a character assigned to several functions, it executes only the function with the highest priority.

The priorities are defined as indicated in the table

| Priority | PAD Functions                           | ParNo   |
|----------|-----------------------------------------|---------|
| highest  | Recall of the PAD                       | 1       |
|          | Command separating character (“+”, “␣”) | -       |
|          | DC1, DC3                                | 12, 22  |
|          | Output of last line                     | 18/ 120 |
|          | Delete one character                    | 16/ 118 |
|          | Delete one line                         | 17/ 119 |
| lowest   | Data forwarding character               | 3       |

Table 6-21: Priority Table

**PROF <ProfileNo>** Used for selection of settings for profile <ProfileNo>.

The values 0-99 are possible as <ProfileNo>; the settings of profiles 0, 90 and 91 are summarized in the following table. User-specific settings for profiles 0-89 and 92-99 are possible.



| Parameters    | Profiles |     |     |
|---------------|----------|-----|-----|
|               | 0        | 90  | 91  |
| Escape        | 0        | 1   | 0   |
| Echo          | 0        | 1   | 0   |
| ForwardChar   | 0        | 126 | 0   |
| IdleTimer     | 30       | 0   | 20  |
| DevControl    | 0        | 1   | 0   |
| SigControl    | 1        | 1   | 0   |
| BrkControl    | 8        | 2   | 2   |
| Discard       | 0        | 0   | 0   |
| CRPadding     | 0        | 0   | 0   |
| LineFold      | 0        | 0   | 0   |
| FlowControl   | 0        | 1   | 0   |
| (X)LFInsert   | 0        | 0   | 0   |
| LFPadding     | 0        | 0   | 0   |
| Edit          | 0        | 0   | 0   |
| (X)CharDel    | 0        | 127 | 127 |
| (X)LineDel    | 0        | 24  | 24  |
| (X)LineDisp   | 0        | 18  | 18  |
| SigEdit       | 0        | 1   | 1   |
| EchoMask      | 0        | 0   | 0   |
| Parity        | 0        | 0   | 0   |
| PageWait      | 0        | 0   | 0   |
| XForwardChar1 | 0        | 0   | 0   |
| XForwardChar2 | 0        | 0   | 0   |
| XParity       | 0        | 0   | 0   |

| Parameters | Profiles |    |    |
|------------|----------|----|----|
|            | 0        | 90 | 91 |
| XDelay     | 0        | 0  | 0  |

Table 6-22: Settings for PROF <ProfileNo>

- Profile 0 is the initial profile set at the start of PAD.
- Profile 90 is the simple standard profile according to X.28.
- Profile 91 is the standard transparent profile according to X.28.

(The settings for the individual profiles can be queried on the **BRICK** in the **x25PadProfileTable**.)

- RESET** ↓ Resets an existing connection to the initial state without disconnecting it, i.e. all data packets sequence numbers are set to 0 and no data packets are on the transfer section.
- INT** ↓ Transmits an interrupt packet. The PAD only sends a line feed (CR LF) as acknowledgment of this command.
- <address>** ↓ Establishes a connection to the <address> (valid X.25 address) indicated after a physical connection has been established.
- ^P** After input of this character the PAD switches from the data transfer state to the command mode, if parameter 1 has the value 1. Other characters are also possible instead of **^P (Control-P)**, please refer to the description of parameter 1 in [Standard Parameters and their Meaning, page 219](#).
- This command is acknowledged by a prompt \* only if parameter 6 is set to an appropriate value.
- The PAD now waits for the input of a PAD command.
- In the X.28 mode, the PAD automatically returns to the data transfer state after each command (except the CLR command).
- Under certain conditions, it is possible to effect a short-time stop and restart of the output by entering DC1 and DC3.

## Further Commands

In addition, the following command is implemented:

**BYE.** Terminates PAD (and disconnects an existing connection).

**Validity of PAD Commands** The following matrix shows the validity of PAD command signals in dependence of the state of the DTE (start-stop mode DTE):

| PAD Commands | Valid before virtual call setup | Valid after escaping from data transfer state |
|--------------|---------------------------------|-----------------------------------------------|
| <address>    | X                               |                                               |
| PROF         | X                               | X                                             |
| SET          | X                               | X                                             |
| SET?         | X                               | X                                             |
| PAR?         | X                               | X                                             |
| CLR          |                                 | X                                             |
| STAT         | X                               | X                                             |
| RESET        |                                 | X                                             |
| INT          |                                 | X                                             |
| RSET?        |                                 | X                                             |
| RPAR?        |                                 | X                                             |
| ICLR         |                                 | X                                             |

Table 6-23: Validity of PAD Commands

**Initial Profile** Whenever a new PAD is created by accepting an ISDN call, the values of the parameters are initialized according to the initial profile, which is always profile 0.

The profiles 0 (initial profile), 90 (simple standard profile) and 91 (transparent standard profile) are by default implemented in the **BRICK**. These profiles can be selected with the command PROF (see [PROF <ProfileNo>](#), page 232).

These three profiles can also be selected, when they are not entered in the **x25PadProfileTable**.

In the following paragraphs, the default settings for all parameters are indicated, with the number (here the PAD parameter number, not the number of the table entry) and name of the parameter followed by a description of the value selected.

| Parameter             | Default Setting | Meaning                                                       |
|-----------------------|-----------------|---------------------------------------------------------------|
| <b>1 Escape</b>       | 0               | It is not possible to leave the data transfer state.          |
| <b>2 Echo</b>         | 0               | The echo mode is disabled; no echo. (no_echo)                 |
| <b>3 ForwardChar</b>  | 0               | No data forwarding character assigned.                        |
| <b>4 Idle Timer</b>   | 5               | 5*50ms= 250 ms                                                |
| <b>5 DevControl</b>   | 0               | No use of DC1 and DC3 (no_use).                               |
| <b>6 SigControl</b>   | 1               | X.28 messages are transmitted to the DTE.                     |
| <b>7 BrkControl</b>   | 8               | Data forwarding, switch to command mode.                      |
| <b>8 Discard</b>      | 0               | Normal data output to the DTE (normal_data_delivery).         |
| <b>9 CRPadding</b>    | 0               | X.28 messages are transmitted to the DTE.                     |
| <b>10 LineFold</b>    | 0               | No automatic line folding.                                    |
| <b>11 Speed</b>       |                 | Detected automatically; internal value                        |
| <b>12 FlowControl</b> | 0               | No use of DC1 and DC3 for data flow control. (no_use_DC1_DC3) |
| <b>13 LFInsert</b>    | 0               | Editing not possible (no_editing_user_data).                  |
| <b>14 LFPadding</b>   | 0               | No padding characters.                                        |
| <b>15 Edit</b>        | 0               | Editing not possible (no_editing_user_data).                  |
| <b>16 CharDel</b>     | 0               | No editing.                                                   |
| <b>17 LineDel</b>     | 0               | No editing.                                                   |
| <b>18 LineDisp</b>    | 0               | No display.                                                   |

| Parameter                | Default Setting | Meaning                                                                                   |
|--------------------------|-----------------|-------------------------------------------------------------------------------------------|
| <b>19 SigEdit</b>        | 0               | No editing PAD service signals.                                                           |
| <b>20 EchoMask</b>       | 0               | No echo mask.                                                                             |
| <b>21 Parity</b>         | 0               | No parity bit checking or generation (no_parity).                                         |
| <b>22 PageWait</b>       | 0               | Page wait disabled.                                                                       |
| <b>118 XCharDel</b>      |                 | Repetition of parameter 16.                                                               |
| <b>119 XLineDel</b>      |                 | Repetition of parameter 17.                                                               |
| <b>120 XLineDisp</b>     |                 | Repetition of parameter 18.                                                               |
| <b>121 XForwardChar1</b> | 0               | No additional data forwarding character.                                                  |
| <b>122 XForwardChar2</b> | 0               | No additional data forwarding character.                                                  |
| <b>123 XParity</b>       | 0               | No parity bit checking or generation (no_parity).                                         |
| <b>125 XDelay</b>        | 0               | No delay of data forwarding; Only with full-duplex connections (parameter 2 is set to 1). |
| <b>126 XLFInsert</b>     |                 | Repetition of parameter 13.                                                               |

Table 6-24: Default Parameter Settings

**Disconnect by the remote PAD**

If a connection is cleared by the remote PAD or by the network, the local PAD returns to the command mode. If parameter 6 (PAD messages) is set to 0, the PAD cannot communicate the disconnect to the user. The PAD is terminated in this case.

**Configuration Necessities for the PAD**

The configuration of the X.25 PAD is described in [chapter 6.3.8, page 211](#).

**Minipad**

The following prompt is displayed with the command `minipad`:

```
[-7] [-p pktksz] [-w winsz] [-c cug] [-o outgocug] [-b bcug]
<x25address>
```

The minipad program is a basic PAD (Packet Assembler/Disassembler) program that can be used to provide a remote login services for remote X.25 hosts. Minipad takes the following arguments:

| Command | Argument     | Meaning                                                                            |
|---------|--------------|------------------------------------------------------------------------------------|
| -7      |              | Use 7 bit data bytes only.                                                         |
| -p      | pktsz        | Open data connection with packet size <pktsz>.                                     |
| -w      | winsz        | Open data connection with window size <winsz>.                                     |
| -c      | cug          | Closed user group. Possible values for <cug>: 0-9999.                              |
| -o      | outgocug     | Closed user group with outgoing access.<br>Possible values for <outgocug>: 0-9999. |
| -b      | bcug         | Bilateral Closed user group.<br>Possible values for <bcug>: 0-9999.                |
|         | <x25address> | Either a standard X.121 address or an extended address.                            |

Table 6-25: Minipad Usage

Minipad is also useful for testing X.25 routes. To disable X.25 connections to the minipad, **x25LocalPadCall** must be set to `dont_accept`.

## 6.5 X.25 Diagnostic Code

X.25 diagnostic codes are reported in the **x25CallHistoryTable**. Note that only clear and diagnostic causes reported by the ISDN are stored in this table (via the **ClearCause** and **ClearDiag** fields). Restart and Reset causes may be detected when tracing ISDN channels.

The diagnostic codes are divided up in following groups:

- Clear Causes
- Diagnostic Causes
- Restart Causes
- Reset Causes



### 6.5.1 Clear Causes

Clear causes are reported in the **ClearCause** field of the **x25CallHistoryTable**.

| Decimal | Hexa | Meaning                                    |
|---------|------|--------------------------------------------|
| 1       | 0x01 | number busy                                |
| 3       | 0x03 | invalid facility request                   |
| 5       | 0x05 | network congestion                         |
| 9       | 0x09 | out of order                               |
| 11      | 0x0B | access barred                              |
| 13      | 0x0D | not obtainable                             |
| 17      | 0x11 | remote procedure error                     |
| 19      | 0x13 | local procedure error                      |
| 21      | 0x15 | RPOA out of order                          |
| 25      | 0x19 | reverse charging acceptance not subscribed |
| 33      | 0x21 | incompatible destination                   |
| 41      | 0x29 | fast select acceptance not subscribed      |
| 57      | 0x39 | ship absent                                |

Table 6-26: Clear Causes

## 6.5.2 Diagnostic Causes

Diagnostic causes are reported in the **ClearDiag** field of the **x25CallHistoryTable**.

| Decimal | Hexa | Meaning                              |
|---------|------|--------------------------------------|
| 0       | 0x00 | no additional information            |
| 1       | 0x01 | invalid P (S)                        |
| 2       | 0x02 | invalid P (R)                        |
| 16      | 0x10 | packet type invalid                  |
| 17      | 0x11 | for state r1                         |
| 18      | 0x12 | for state r2                         |
| 19      | 0x13 | for state r3                         |
| 20      | 0x14 | for state p1                         |
| 21      | 0x15 | for state p2                         |
| 22      | 0x16 | for state p3                         |
| 23      | 0x17 | for state r1                         |
| 24      | 0x18 | for state p5                         |
| 25      | 0x19 | for state p6                         |
| 26      | 0x1a | for state p7                         |
| 27      | 0x1b | for state d1                         |
| 28      | 0x1c | for state d2                         |
| 29      | 0x1d | for state d3                         |
| 32      | 0x20 | packet not allowed                   |
| 33      | 0x21 | unidentifiable packet                |
| 34      | 0x22 | call on one-way logical channel      |
| 35      | 0x23 | invalid packet type on a PVC         |
| 36      | 0x24 | packet on unassigned logical channel |

| Decimal | Hexa | Meaning                                                |
|---------|------|--------------------------------------------------------|
| 37      | 0x25 | reject not subscribed to                               |
| 38      | 0x26 | packet too short                                       |
| 39      | 0x27 | packet too long                                        |
| 40      | 0x28 | invalid GFI                                            |
| 41      | 0x29 | restart packet with nonzero logical channel identifier |
| 42      | 0x2a | packet type not compatible with facility               |
| 43      | 0x2b | unauthorized interrupt confirmation                    |
| 44      | 0x2c | unauthorized interrupt                                 |
| 45      | 0x2d | unauthorized reject                                    |
| 48      | 0x30 | time expired                                           |
| 49      | 0x31 | for incoming call                                      |
| 50      | 0x32 | for clear indication                                   |
| 51      | 0x33 | for reset indication                                   |
| 52      | 0x34 | for restart indication                                 |
| 53      | 0x35 | for call deflection                                    |
| 64      | 0x40 | call set-up, call clearing or registration problem     |
| 65      | 0x41 | facility/registration code not allowed                 |
| 66      | 0x42 | facility parameter not allowed                         |
| 67      | 0x43 | invalid called DTE address                             |
| 68      | 0x44 | invalid calling DTE address                            |
| 69      | 0x45 | invalid facility/registration length                   |
| 70      | 0x46 | incoming call barred                                   |
| 71      | 0x47 | no logical channel available                           |
| 72      | 0x48 | call collision                                         |

| Decimal | Hexa | Meaning                                                  |
|---------|------|----------------------------------------------------------|
| 73      | 0x49 | duplicate facility request                               |
| 74      | 0x4a | nonzero address length                                   |
| 75      | 0x4b | nonzero facility length                                  |
| 76      | 0x4c | facility not provided when expected                      |
| 77      | 0x4d | invalid CCITT-specified DTE facility                     |
| 78      | 0x4e | max number of call redirections/<br>deflections exceeded |
| 80      | 0x50 | miscellaneous                                            |
| 81      | 0x51 | improper cause code from DTE                             |
| 82      | 0x52 | non aligned octet                                        |
| 83      | 0x53 | inconsistent Q bit setting                               |
| 84      | 0x54 | NUI problem                                              |
| 112     | 0x70 | international problem                                    |
| 113     | 0x71 | remote network problem                                   |
| 114     | 0x72 | international protocol problem                           |
| 115     | 0x73 | international link out of order                          |
| 116     | 0x74 | international link busy                                  |
| 117     | 0x75 | international link busy                                  |
| 118     | 0x76 | remote network facility problem                          |
| 119     | 0x77 | international routing problem                            |
| 120     | 0x78 | temporary routing problem                                |
| 121     | 0x79 | unknown called DNIC                                      |
| 122     | 0x7a | maintenance action                                       |
| 144     | 0x90 | timer expired or retransmission count<br>surpassed       |
| 145     | 0x91 | for interrupt                                            |

| Decimal | Hexa | Meaning                                                                       |
|---------|------|-------------------------------------------------------------------------------|
| 146     | 0x92 | for data                                                                      |
| 147     | 0x93 | for reject                                                                    |
| 160     | 0xa0 | DTE-specific signals                                                          |
| 161     | 0xa1 | DTE operational                                                               |
| 162     | 0xa2 | DTE not operational                                                           |
| 163     | 0xa3 | DTE resource constraint                                                       |
| 164     | 0xa4 | fast select not subscribed                                                    |
| 165     | 0xa5 | invalid partially full data packet                                            |
| 166     | 0xa6 | D-bit procedure not supported                                                 |
| 167     | 0xa7 | registration/cancellation confirmed                                           |
| 224     | 0xe0 | OSI network service problem                                                   |
| 225     | 0xe1 | disconnection (transient condition)                                           |
| 226     | 0xe2 | disconnection (permanent condition)                                           |
| 227     | 0xe3 | connection rejection– reason unspecified (transient condition)                |
| 228     | 0xe4 | connection rejection - reason unspecified (permanent condition)               |
| 229     | 0xe5 | connection rejection - quality of service not available (transient condition) |
| 230     | 0xe6 | connection rejection - quality of service not available (permanent condition) |
| 231     | 0xe7 | connection rejection - NSAP unreachable (transient condition)                 |
| 232     | 0xe8 | connection rejection - NSAP unreachable (permanent condition)                 |
| 233     | 0xe9 | reset - reason unspecified                                                    |

| Decimal | Hexa | Meaning                                                                       |
|---------|------|-------------------------------------------------------------------------------|
| 234     | 0xea | reset - congestion                                                            |
| 235     | 0xeb | connection rejection - NSAP address unknown (permanent condition)             |
| 240     | 0xf0 | higher layer initiated                                                        |
| 241     | 0xf1 | disconnection - normal                                                        |
| 242     | 0xf2 | disconnection - abnormal                                                      |
| 243     | 0xf3 | disconnection - incompatible information in user data                         |
| 244     | 0xf4 | connection rejection - reason unspecified (transient condition)               |
| 245     | 0xf5 | connection rejection - reason unspecified (permanent condition)               |
| 246     | 0xf6 | connection rejection - quality of service not available (transient condition) |
| 247     | 0xf7 | connection rejection - quality of service not available (permanent condition) |
| 248     | 0xf8 | connection rejection - incompatible information in user data                  |
| 249     | 0xf9 | connection rejection - unrecognizable protocol identifier in user data        |
| 250     | 0xfa | reset - user synchronization                                                  |

Table 6-27: Diagnostic Causes

### 6.5.3 Restart Causes

Restart causes are reported by the ISDN and may be detected when tracing ISDN channels.

These causes are not stored on the **BRICK**.

| Decimal | Hexa | Meaning               |
|---------|------|-----------------------|
| 1       | 0x01 | local procedure error |
| 3       | 0x03 | network congestion    |
| 7       | 0x07 | network operational   |

Table 6-28: Restart Causes

### 6.5.4 Reset Causes

Reset causes are reported by the ISDN and may be detected when tracing ISDN channels.

These causes are not stored on the **BRICK**.

| Decimal | Hexa | Meaning                      |
|---------|------|------------------------------|
| 3       | 0x03 | remote procedure error       |
| 5       | 0x05 | local procedure error        |
| 7       | 0x07 | network congestion           |
| 17      | 0x11 | incompatible destination     |
| 1       | 0x01 | out of order (PVC)           |
| 9       | 0x09 | remote DTE operational (PVC) |
| 15      | 0x0F | network operational (PVC)    |
| 29      | 0x0D | network out of order (PVC)   |

Table 6-29: Reset Causes

## 6.6 X.25 Syslog Messages

(`biboAdmSyslogSubject = x25`)



The value `<fd>` used in X.25 system messages is an internal file number to discriminate between the different X.25 and TCP connections.

| <code>biboAdmSyslogMessage</code>                                                                                            | Meaning                                                                                                            | -Level |
|------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|--------|
| <code>ifc 1 vc &lt;vc&gt;: receive window exceeded, call cleared.</code>                                                     | Protocol error in X.25 connection directly to <b>BRICK</b> (Interface 1).                                          | err    |
| <code>ifc 1 vc &lt;vc&gt;: N(R) out of range, call cleared.</code>                                                           | Protocol error in X.25 connection directly to <b>BRICK</b> (Interface 1).                                          | err    |
| <code>Cannot rewrite call packet; Rule ... does not exist.</code>                                                            | A rewriting rule has been referenced in <b>x25RouteTable</b> , that is not defined in <b>x25RewriteTable</b> .     | err    |
| <code>Unable to route call to IFC ... (X.25 not supported) cannot use ifc ... for routing (ifc does not support X25).</code> | The specified target interface in an entry of the <b>x25RouteTable</b> does not support X.25.                      | err    |
| <code>source address too long (... bytes)</code>                                                                             | The Link Layer Address (MAC) of a target interface specified in the <b>x25RouteTable</b> is longer than 20 Octets. | err    |
| <code>cannot use undefined ifc ... for routing</code>                                                                        | The target interface of an entry in the <b>x25RouteTable</b> does not exist.                                       | err    |



| biboAdmSyslogMessage                                                                                                                                                                                                                | Meaning                                                                                                                                                                                                                         | -Level |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| channel misconfiguration (HIC) on ifc <ifc><br>channel misconfiguration (LTC) on <ifc><br>channel misconfiguration (HTC) on ifc <ifc><br>channel misconfiguration (LOC) on ifc <ifc><br>channel misconfiguration (HOC) on ifc <ifc> | The channel specification of a link in the <b>x25LinkPresetTable</b> does not match the condition:<br>$LIC \leq HIC < LTC \leq HTC < LOC \leq HOC$                                                                              | err    |
| ifc=<ifc> [addr=...] vc=<vc> recv CALL<br><SrcAddr> -> <DstAddr> fac=<fac><br>cud=<user data>                                                                                                                                       | An X.25 CALL-REQUEST/INDICATION has been received. The message contains the interface index, optionally the link-address, the virtual circuit number, source and target address, the call facilities and the call user data.    | debug  |
| ifc=<ifc> [addr=...] vc=<vc> send CALL<br><SrcAddr> -> <DstAddr> fac=<fac><br>cud=<user data>                                                                                                                                       | An X.25 CALL-REQUEST/INDICATION is being sent The message contains the interface index, optionally the link-address, the virtual circuit number, source and target address, the call facilities and the call user data.         | debug  |
| ifc=<ifc> [addr=...] vc=<vc> recv CALL CONFIRM<br><SrcAddr> -> <DstAddr> fac=<fac><br>cud=<user data>                                                                                                                               | An X.25 CALL-RESPONSE/CONFIRMATION has been received. The message contains the interface index, optionally the link-address, the virtual circuit number, source and target address, the call facilities and the call user data. | debug  |
| ifc=<ifc> [addr=...] vc=<vc> send CALL CONFIRM<br><SrcAddr> -> <DstAddr> fac=<fac><br>cud=<user data>                                                                                                                               | An X.25 CALL-RESPONSE/CONFIRMATION is being sent. The message contains, the interface index, optionally the link-address, the virtual circuit number, source and target address, the call facilities and the call user data.    | debug  |

| biboAdmSyslogMessage                                | Meaning                                                                                                                                           | -Level |
|-----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| ifc=<ifc> [addr=...] vc=<vc> recv CLEAR             | A X.25 CLEAR-REQUEST/INDICATION has been received with the given cause and diagnostic codes. The value -1 means, cause or diagnostic not present. | debug  |
| ifc=<ifc> [addr=...] vc=<vc> recv CLEAR             | A X.25 CLEAR-REQUEST/INDICATION is being sent with the given cause and diagnostic codes. The value -1 means, cause or diagnostic not present.     | debug  |
| ifc=<ifc> [addr=...] vc=<vc> send CLEAR             | A X.25 CLEAR-REQUEST/INDICATION is being sent without cause and diagnostic.                                                                       | debug  |
| ifc=<ifc> [addr=...] vc=<vc> recv CLEAR CONFIRM     | A X.25 CLEAR-RESPONSE/CONFIRM has been received on the given VC.                                                                                  | debug  |
| ifc=<ifc> [addr=...] vc=<vc> send CLEAR CONFIRM     | A X.25 CLEAR-RESPONSE/CONFIRM is being sent.                                                                                                      | debug  |
| ifc=<ifc> [addr=...] vc=<vc> recv RESET             | A X.25 RESET-REQUEST/INDICATION has been received on the given VC.                                                                                | debug  |
| ifc=<ifc> [addr=...] vc=<vc> recv RESET CONFIRM     | A X.25 RESET-RESPONSE/CONFIRM is being sent.                                                                                                      | debug  |
| ifc=<ifc> [addr=...] vc=<vc> recv INTERRUPT         | A X.25 INTERRUPT has been received on the given VC.                                                                                               | debug  |
| ifc=<ifc> [addr=...] vc=<vc> send RESET CONFIRM     | A X.25 RESET-RESPONSE/CONFIRM is being sent.                                                                                                      | debug  |
| ifc=<ifc> [addr=...] vc=<vc> recv INTERRUPT         | A X.25 INTERRUPT has been received on the given VC                                                                                                | debug  |
| ifc=<ifc> [addr=...] vc=<vc> send INTERRUPT         | A X.25 INTERRUPT is being sent.                                                                                                                   | debug  |
| ifc=<ifc> [addr=...] vc=<vc> recv INTERRUPT CONFIRM | A X.25 INTERRUPT-CONFIRM has been sent on the given VC                                                                                            | debug  |
| ifc=<ifc> [addr=...] vc=<vc> send INTERRUPT CONFIRM | A X.25 INTERRUPT-CONFIRM is being sent.                                                                                                           | debug  |

| biboAdmSyslogMessage                                                                                 | Meaning                                                                                                                                                                       | -Level |
|------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| ifc=<ifc> [addr=...] vc=<vc> recv DIAG<br>cause=<causecode> diag=<diagcode>                          | A X.25 DIAG has been received on the given VC. This message is ignored.                                                                                                       | debug  |
| ifc=<ifc> [addr=...] vc=<vc><br>invalid VC number                                                    | A call on an unassigned VC number was received.                                                                                                                               | debug  |
| ifc=<ifc> [addr=...] vc=<vc> call collision                                                          | A call collision occurred on the given VC and will be handled according to X.25.                                                                                              | debug  |
| ifc=<ifc> [addr=...] vc=<vc><br>TIMEOUT                                                              | A timeout condition occurred on a VC while waiting for a CALL-RESPONSE/CONFIRMATION, CLEAR-RESPONSE/CONFIRMATION, or a RESET-RESPONSE/CONFIRMATION. The call will be cleared. | debug  |
| ifc=<ifc> [addr=...] vc=<vc><br>windowsize=<incoming>/<outgoing><br>packetsize=<incoming>/<outgoing> | The call's incoming/outgoing parameters for window size and packet size will be used according to the given values (possibly after negotiation).                              | debug  |
| ifc=<ifc> [addr=...] recv RESTART<br>cause=<cause>                                                   | A restart packet has been received on the given link with the given cause. If the cause value is set to -1, the cause was not present in the message.                         | debug  |
| ifc=<ifc> [addr=...] send RESTART                                                                    | A RESTART packet is being sent over the given link.                                                                                                                           | debug  |
| ifc=<ifc> [addr=...] recv RESTART<br>CONFIRM                                                         | A RESTART-CONFIRM packet has been received on the given link.                                                                                                                 | debug  |
| ifc=<ifc> [addr=...] send RESTART<br>CONFIRM                                                         | A RESTART-CONFIRM packet is being sent over the given link.                                                                                                                   | debug  |
| ifc=<ifc> [addr=...] vc=<vc> recv<br>ILLEGAL message                                                 | An unknown message has been received on the given VC.                                                                                                                         | debug  |
| ifc=<ifc> [addr=...] vc=<vc><br>invalid VC number                                                    | An unknown message has been received on the given VC.                                                                                                                         | debug  |

| biboAdmSyslogMessage                        | Meaning                                                                                                                                   | -Level |
|---------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|--------|
| ifc=<ifc> [addr=...] TIMEOUT                | A timeout occurred on the given link, while waiting for RESTART, RESTART-CONFIRMATION, XID negotiation, link establishment or being idle. | debug  |
| ifc=<ifc> [addr=...] restarting             | The restart procedure starts on the given link and a restart packet is being sent.                                                        | debug  |
| ifc=<ifc> [addr=...] resetting layer 2      | The layer 2 of the given link is being reset due to a timeout while waiting for a RESTART. A SABM[E] will be sent.                        | debug  |
| ifc=<ifc> [addr=...] disconnecting layer 2  | The given link will be disconnected, while being idle, i.e. no VCs being established. A DISC will be sent.                                | debug  |
| ifc=<ifc> [addr=...] connecting layer 2     | The given link will be established and a SABM[E] will be sent.                                                                            | debug  |
| ifc=<ifc> [addr=...] layer 2 connected      | The connect request (SABM[e]) has been accepted by the peer and a UA frame has been received.                                             | debug  |
| ifc=<ifc> [addr=...] accept layer 2 connect | An incoming connect indication (SABM[E]) on the given link will be accepted and a UA frame being sent.                                    | debug  |
| ifc=<ifc> [addr=...] accept layer 2 reset   | An incoming reset indication (SABM[E]) on the given link will be accepted and a UA frame being sent.                                      | debug  |
| ifc=<ifc> [addr=...] layer 2 reset          | The reset request (SABM[e]) has been accepted by the peer and a UA frame has been received.                                               | debug  |

| biboAdmSyslogMessage                                                        | Meaning                                                                                                                                                                                                                                       | -Level |
|-----------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| ifc=<ifc> [addr=...] layer 2 disconnected                                   | A disconnect indication (DISC) has been received on the given link and the link is no longer established.                                                                                                                                     | debug  |
| dialup ifc ...                                                              | The given interface is dialed up due to an X.25 call routed to it. The message contains, the interface index, optionally the link-address, the virtual circuit number, source and target address, the call facilities and the call user data. | debug  |
| txd[<fd>]: <tcpaddr>:<port> New TCP connection                              | A new incoming TCP connection from the specified TCP address via the local port 146 has been established.                                                                                                                                     | debug  |
| txd[<fd>]: <tcpaddr>:<port> First byte ... - not supported                  | The first byte the TCP host sent to port 146 isn't supported by the <b>BRICK</b> . Only the values 1 and 2 are allowed.                                                                                                                       | debug  |
| txd[<fd>]: <tcpaddr>:<port> Connect to a particular X.25 host               | The host with the specified TCP address wants to connect to a particular X.25 host.                                                                                                                                                           | debug  |
| txd[<fd>]: <tcpaddr>:<port> Listen for incoming X.25 call on addr=<address> | The host with the specified TCP address wants to listen for incoming X.25 connections for the specified X.25 listening address.                                                                                                               | debug  |
| txd[<fd>]: <tcpaddr>:<port>                                                 | Timeout while reading X.25 address The specified TCP host didn't send the X.25 address completely within a certain amount of time.                                                                                                            | debug  |
| txd[<fd>]: <tcpaddr>:<port> unsupported X.25 address type                   | The address type field entry of the X.25 address, the TCP host sent, isn't supported by the <b>BRICK</b> . Only the values 3 and 4 are allowed.                                                                                               | debug  |

| biboAdmSyslogMessage                                                         | Meaning                                                                                                                                                 | -Level |
|------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| txd[<fd>]: <tcpaddr>:<port> Could not read 16 byte TCP/IP packet             | The specified TCP host didn't send the complete TCP/IP address of the listening TCP host within a certain amount of time.                               | debug  |
| txd[<fd>]: <tcpaddr>:<port> IP Address type ... not supported                | The address type field entry of the TCP/IP address of the listening TCP host, isn't supported by the <b>BRICK</b> . Only the value 2 is allowed.        | debug  |
| txd[<fd>]: <tcpaddr>:<port><br>Connection to X.25 host addr=... failed       | The TCP host wanted to connect to the specified X.25 address but the <b>BRICK</b> could not reach the X.25 host.                                        | debug  |
| txd[<fd>]: X.25 CALL_IND<br>dest_addr=<address>                              | An X.25 call indication for the specified X.25 address was received by the <b>BRICK</b> .                                                               | debug  |
| txd[<fd>]: Connection failed -<br>wrong X.25 address                         | There is currently no TCP host bound to the X.25 address of the previously received X.25 call indication.                                               | debug  |
| txd[<fd>]: Connected to X.25<br>addr=...                                     | An incoming X.25 connection was established                                                                                                             | debug  |
| txd[<fd>]: Connected to TCP<br><tcpaddr>:<port>                              | The <b>BRICK</b> opened a new TCP connection to the specified listening TCP host.                                                                       | debug  |
| txd[<fd>]: <tcpaddr>:<port> TCP<br><--> txd[<fd>] X.25 addr=...<br>connected | The <b>BRICK</b> connected an incoming X.25 call to the specified TCP host.                                                                             | debug  |
| txd[<fd>]: Disconnect and close<br>connection                                | The <b>BRICK</b> disconnects the TCP host and the X.25 host.                                                                                            | debug  |
| txd[<fd>]: Received disconnect,<br>cause=<causecode> diag=<diagcode>         | The <b>BRICK</b> received a disconnect message from the X.25 connection. The cause and diagnostic codes of the X.25 clear indication message are shown. | debug  |

| biboAdmSyslogMessage           | Meaning                                                                 | -Level |
|--------------------------------|-------------------------------------------------------------------------|--------|
| txd[<fd>]: Received disconnect | The <b>BRICK</b> received a disconnect message from the TCP connection. | debug  |
| No License                     | An attempt has been made to use X.25 without a valid license.           | info   |

Table 6-30: **biboAdmSyslogMessage** Table and Meaning

## 6.7 X.21 Communications Module

**Normal Operation Mode** During normal operation, PWR (power) always displays whether the router is receiving power. ERR (error) is normally off but may blink when an error, such as a cabling problem, has occurred.

Depending on which slots your communications modules are installed in, the A/B LEDs for slots 1, 2, and 3 are as follows:

| CM-X21 |       |                                    |
|--------|-------|------------------------------------|
| LED    | State | Meaning                            |
| A      | On    | Currently receiving an X.21 frame. |
| B      | On    | Currently sending an X.21 frame.   |

Table 6-31: LED Status

Depending on which slots your communications modules are installed in, the LEDs for slots 1 through 6 (S1... S6) are as follows:

| Modules | State | Meaning                        |
|---------|-------|--------------------------------|
| CM-X21  | On    | Sending or receiving a packet. |

Table 6-32: Module Status

### 6.7.1 CM-X21Adapter

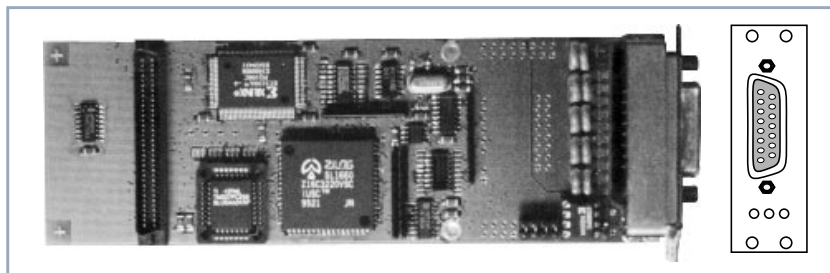


Figure 6-18: CM-X21Adapter



The CM-X21 module provides a standard X.21 interface which complies with the V.11 recommendation. The X.21 interface provides a full-duplex synchronous mode and can be configured to operate as either a DTE (passive mode) or DCE (active mode). When in active mode the X.21 interface can be set to operate at baud rates between 2400 and 2048k.

There are also three status indicators located on the back plane. The LEDs indicate various status conditions, as follows:

| Color | State | Meaning                                                             |
|-------|-------|---------------------------------------------------------------------|
| Red   | On    | Error transmitting a packet.                                        |
| Amber | On    | Frame being sent/received.                                          |
| Green | On    | Layer 1 is active (i.e., incoming and outgoing calls are possible). |

Table 6-33: CM-X21 back plane LEDs



The four jumper settings on the X.21 module are intended for future use. They should remain bridged (or jumpered), these are the default settings and should not be changed.

#### 15 Pin Port for the CM-X21

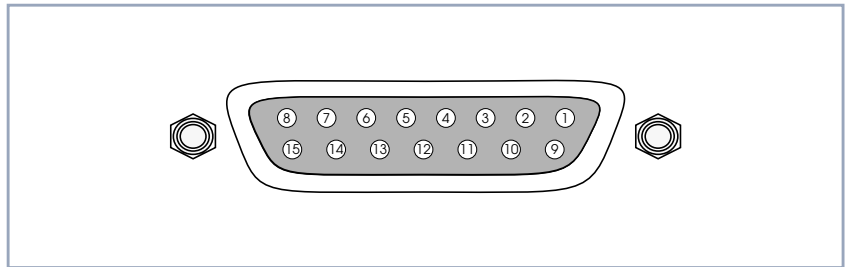


Figure 6-19: 15 Pin X21 Port

The pin assignments for the CM-X21 module conform to the V.11 recommendations and are as follows:

| Pin | Function                  | Mnemonic |
|-----|---------------------------|----------|
| 1   | Protection Ground         | PG       |
| 2   | Transmit (A)              | T        |
| 3   | Control (A)               | I        |
| 4   | Receive (A)               | R        |
| 5   | Indicate (A)              | I        |
| 6   | Signal Timing Element (A) | S        |
| 7   | Not Connected             |          |
| 8   | Signal Ground             | SG       |
| 9   | Transmit (B)              | T        |
| 10  | Control (B)               | I        |
| 11  | Receive (B)               | R        |
| 12  | Indicate (B)              | I        |
| 13  | Signal Timing Element (B) | S        |
| 14  | Not Connected             |          |
| 15  | Not Connected             |          |

Table 6-34: Pin Assignment

## 7 Frame Relay

In this chapter we will give you an overview of Frame Relay technology.

Secondly, we will describe the protocol structures of Frame Relay.

After that some Frame Relay services will be introduced.

Following that, the Frame Relay subsystems will be described.

Concluding, we will describe some example configurations using Setup Tool.

### Frame Relay on BinTec Routers

Frame Relay is officially supported on the BIANCA/BRICK-XL2, BIANCA/BRICK-XMP, BIANCA/BRICK-XM with 2MB flash, BIANCA/BRICK-XS with 2MB flash, and on the BinGO! Plus/Professional. The **BRICK** (the expression **BRICK** in the further text of this Chapter also includes the BinGO! Plus/Professional) can be used as a Frame Relay Switch or a Frame Relay Router and supports the following official and defacto standards:

- RFC 1490 Multiprotocol Interconnect over Frame Relay
- RFC 1293 Inverse Address Resolution Protocol
- ITU-T Q933a, Appendix II, X6 Line Management Extensions
- FRF 1.1 Congestion Management



Frame Relay requires a separate license to be installed on the **BRICK** and may be purchased directly from BinTec Communications AG or your local distributor.

Frame relay is a connection-oriented technology that provides a fast packet-switching service for access to Wide Area Networks. It makes optimum use of available bandwidth using a complex statistical multiplexing algorithm. Due to the omission of some layer three network functions, Frame Relay is often thought of as a “streamlined version for X.25”.

Frame Relay is a flexible and cost-effective alternative to existing WAN technologies best suited for network installations exemplifying any of the following characteristics:

- Applications generate significant amounts of bursty traffic.
- Network traffic is delay-sensitive.
- High network availability is a major priority.
- Dispersed enterprise (locations separated by long distances).
- Integration with existing public and/or private, packet-switched networks is required.

## 7.1 An Overview of Frame Relay Technology

As the name suggests, it works by breaking data streams into variable length frames and forwards (relays) these frames into the network via predetermined logical connections called Permanent Virtual Circuits, or PVCs.

Some of the key concepts of Frame Relay are listed below:

- Small, variable length frames are used to transport user data; this makes frame relay well suited for data applications (particularly those generating bursty-traffic) - video and voice transmissions are generally not appropriate.
- Improved overall performance (compared to X.25 - a result of limited error correction and acknowledgment routines).
- Users are guaranteed a minimum amount of bandwidth which is always available (the Committed Information Rate [chapter 7.3.1, page 267](#), or CIR).
- High network availability is achieved through statistically multiplexing virtual connections (data streams) onto logical connections, or Permanent Virtual Circuits (PVCs).
- Integrated bandwidth allocation (true bandwidth on demand) allows users to take up additional bandwidth, when available, at no extra charge - based on the user's Committed Burst Rate [chapter 7.3.2, page 267](#) (CBR) and Excess Burst Rate [chapter 7.3.3, page 267](#) (EBR).

There are different types of equipment found in a typical Frame Relay Network based on the various tasks they perform.

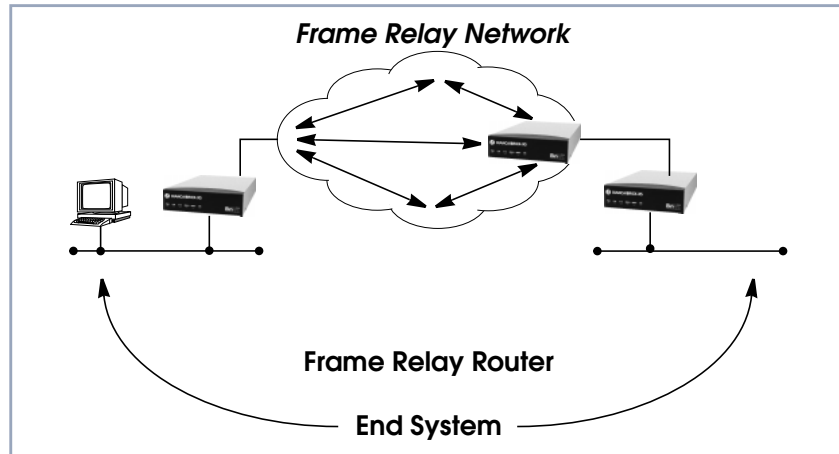


Figure 7-1: Frame Relay Network

#### ■ End Systems

End systems are typically end-user devices that take advantage (make use of) the underlying Frame Relay network. Depending on the application running on the end stations bandwidth requirements of end systems on the LAN can be different. Some applications generate large amounts of intermittent bursty traffic (typical of data applications, telnet, ftp, www) while others (like voice or video) require a constant bitrate.

#### ■ Frame Relay Routers

Frame Relay Routers are used to connect point-to-multipoint networks (LANs) to a public (or private) Frame Relay network. It is the router's job to encapsulate data into Frame Relay frames for transport over the network link. A Frame Relay Router encapsulates LAN frames in frame relay frames and feeds those frames to a Frame Relay Switch for transmission across the network. A Frame Relay Router also receives frame relay frames from the network, strips the frame relay frame off each frame to produce the original LAN frame, and passes the LAN frame on to the end device. A Frame Relay Router communicates directly with one or more Frame Relay Switches to negotiate the opening/closing of virtual circuits and to control network congestion.

- Frame Relay Switches

Switches are typically owned by public network providers but may be owned by private sites implementing private Frame Relay Networks. Aside from the FECN, BECN, and DE frame fields (used for congestion management) the content and final destination of individual frame is of no interest to the switch. Using a simple mapping scheme frames are passed from one interface (DLCI) to another.

## 7.2 Protocol Structure

### 7.2.1 Frame Relay Protocol Stack

Although similar in concept to X.25, frame relay operates at layer 2 of the OSI reference model. This is where the main differences between the two lie. Frame relay simply leaves out the extensive error detection/correction and end-to-end flow control found in X.25. This greatly simplifies the tasks a frame relay switch must perform.

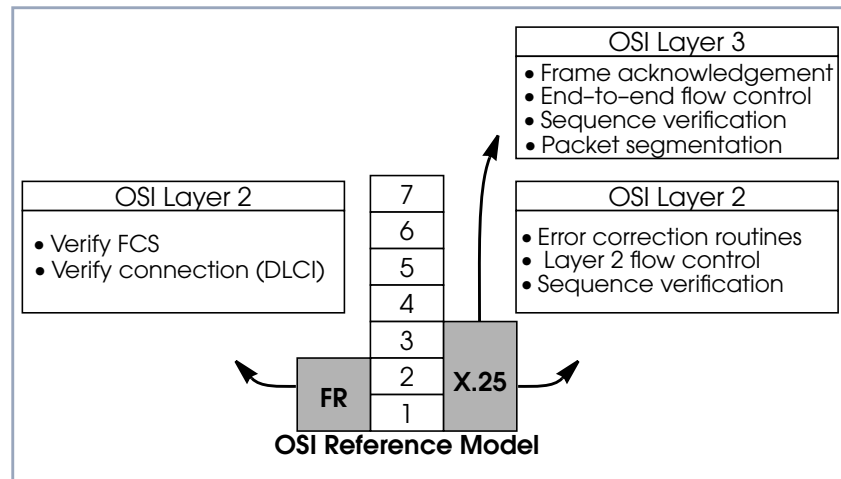


Figure 7-2: Frame Relay in OSI Reference Model



## 7.2.2 Frame Relay Frame Format

As shown below frame relay is a streamlined protocol that uses HDLC framing. Virtual frame relay connections are routed based on the DLCI field of incoming frames.

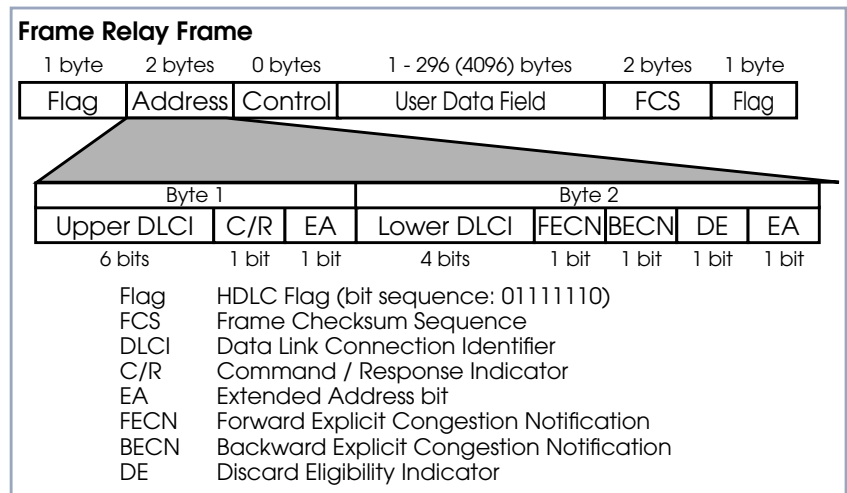


Figure 7-3: Frame Relay Frame

## 7.2.3 Frame Relay Addressing

The basic (unextended) Frame Relay specification only supports locally significant addressing. These addresses are up to 2 bytes long. Using the EA fields extended addresses can be used which may be up to 4 bytes long.

When a frame is read the first EA bit that is set (i.e., its value = 1) determines the address.

## 7.2.4 Congestion Notification

The FECN and BECN bits (see above) are used to notify neighboring frame relay devices of possible congestion.

## 7.2.5 Virtual Circuits

In Frame Relay multiple connections are mapped to a single physical network connection.

## 7.2.6 Data Link Connection Identifier

The DLCI field is used to route virtual frame relay connections. A standard DLCI (2 byte address field) consists of 10 bits and is based on the frame's Upper and Lower DLCI fields. These 10 bits establish an upper limit of 1024, 210, possible simultaneous virtual channels that can be multiplexed on to a PVC.

The DLCI field is used to route virtual frame relay connections. A standard DLCI (2 byte address field) consists of 10 bits and is based on the frame's Upper and Lower DLCI fields. These 10 bits establish an upper limit of 1024, 210, possible simultaneous virtual channels that can be multiplexed on to a PVC.

| DLCI        | Use (Q.922)                                   | Use (LMI)    |
|-------------|-----------------------------------------------|--------------|
| 0           | Signalling                                    | Reserved     |
| 1 - 15      | Reserved                                      | Reserved     |
| 16 - 511    | Available (except when the D-channel is used) | Available    |
| 512 - 991   | Available                                     | Available    |
| 992 - 1007  | Layer 2 management                            | Available    |
| 1008 - 1018 | Reserved                                      | Reserved     |
| 1019 - 1022 | Reserved                                      | Multicasting |
| 1023        | Consolidated Link Layer Management            | Signalling   |



A DLCI is only significant to the local station. Though it is used locally to identify both directions of a virtual circuit it has no meaning to the next station (or the destination) in the frame relay network.

## 7.3 Frame Relay Services

Frame relay access can be purchased in a variety of configurations depending of your site's needs. Characteristics of the service you will receive include:

1. The type of physical connection you have to the frame relay network, ISDN or X.21.
2. The amount (from 56Kbps up to 2Mbps) and type of bandwidth available via this connection; this will include your guaranteed and excess rates. See CIR, CBR, and EBR earlier.
3. The number of PVCs you are receiving.

### 7.3.1 Committed Information Rate

When purchasing frame relay services from your provider, you will be assigned a Committed Information Rate. This defines the minimum amount of bandwidth that your provider guarantees to be available to your site at all times.

### 7.3.2 Committed Burst Rate

You will also receive a Committed Burst Rate with your service package. This is an additional amount of bandwidth (in excess of your CIR) you may use when network resources are available. The CBR is free of charge, but be aware that all frames that are in excess of your CIR will be DE (Discard Eligible) flagged and may be discarded by intermediate switches if the network becomes congested.

### 7.3.3 Excess Burst Rate

As Excess Burst Rate is also available; it defines the maximum data rate the service provider's network will attempt to sustain. Also note that all EBR traffic is flagged Discard Eligible.

## 7.4 The Frame Relay Subsystem

Frame Relay on the **BRICK** consists of 5 SNMP system tables contained in the **BRICK**'s **fr** group. An overview of these tables is shown below. The full description of each SNMP object is contained on the following pages.

### 7.4.1 Overview: Frame Relay System Tables

| Variable              | Meaning                                                                                                                                                                                                                                                                             |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>frGlobals</b>      | Global settings for Frame Relay on the <b>BRICK</b> . Currently only contains the <b>frTrapState</b> object which is used to enable/disable <b>frDLCIStatusChange</b> traps on the <b>BRICK</b> . (This trap indicates that the state of a particular Virtual Circuit has changed.) |
| <b>frDlcmiTable</b>   | Contains parameters for each DLCM (Data Link Connection Management) interface for each instance of frame relay service on the <b>BRICK</b> .                                                                                                                                        |
| <b>frCircuitTable</b> | Contains information for each Data Link Connection Identifiers and corresponding virtual circuits.                                                                                                                                                                                  |
| <b>frErrTable</b>     | Used to store important status messages reported for interfaces configured with Local Management Interface.                                                                                                                                                                         |
| <b>frMprTable</b>     | Contains Multiprotocol Routing over Frame Relay interfaces (MPFR) on the <b>BRICK</b> . These interfaces are Virtual interfaces since they do not necessarily map to a single hardware interface. MPFR interfaces may be used by higher level protocols.                            |

Table 7-1: Frame Relay System Tables

| <b>biboAdmSyslogMessage</b>            | <b>-Level</b> |
|----------------------------------------|---------------|
| Attach link <ifindex> failed           | debug         |
| Attach link <ifindex>                  | debug         |
| Bind link <ifindex> failed             | debug         |
| Link <ifindex> bound; starting LMI     | debug         |
| Be exceeded - packet discarded         | debug         |
| Want open ifc <ifindex>                | debug         |
| Unknown ARP protocol <proto>           | debug         |
| No license                             | info          |
| DLCI out of range: <dlci>              | notice        |
| No more than 256 interfaces allowed    | error         |
| Create: illegal index <ifindex>        | error         |
| Create: index <ifindex> already exists | error         |

**Table 7-2: biboAdmSyslogMessage**

## 7.4.2 Frame Relay Setup Tool Menus

Several menus have been added to Setup Tool to allow for easy configuration of Frame Relay on the **BRICK**. An overview of the menu structure is shown below. Individual submenus are described in detail on the following pages.

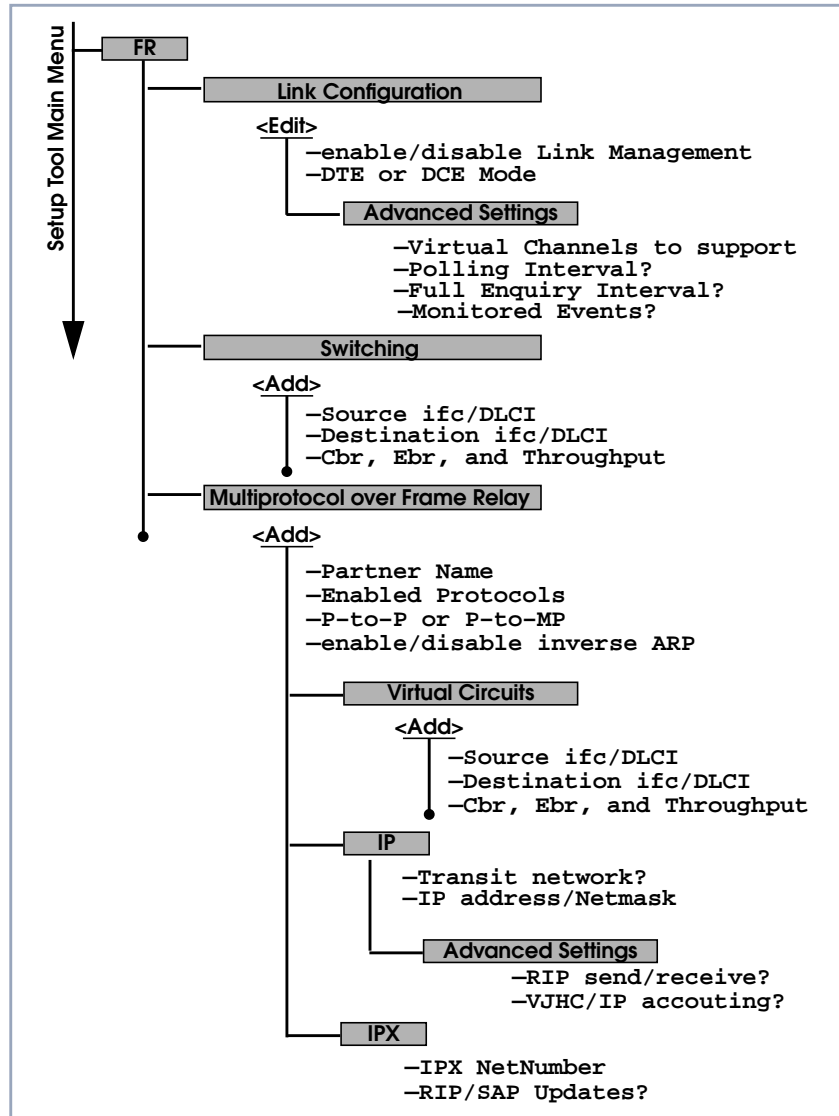


Figure 7-4: Setup Tool Menu Structure

### 7.4.3 Setup Tool Menus

Frame Relay on the **BRICK** can be configured from Setup Tool using the three menus available here.

|                                                                               |                          |
|-------------------------------------------------------------------------------|--------------------------|
| BRICK Setup Tool                                                              | BinTec Communications AG |
| [FRAME RELAY]: Frame Relay Configuration                                      | MyBRICK                  |
| Link Configuration<br>Switching<br>Multiprotocol over Frame Relay<br><br>EXIT |                          |
| Press <Ctrl-n>, <Ctrl-p> to scroll through menu items, <Return> to enter      |                          |

| Field                                 | Meaning                                                                 |
|---------------------------------------|-------------------------------------------------------------------------|
| <b>Link Configuration</b>             | contains the settings relative to the layer 2 of Frame Relay interface. |
| <b>Switching</b>                      | lists settings for each Frame Relay Virtual Circuit.                    |
| <b>Multiprotocol over Frame Relay</b> | lists all existing MPFR interfaces configured on the <b>BRICK</b> .     |

Table 7-3: **FR** ► **FRAME RELAY CONFIGURATION**

► Go to **FR** ► **LINK CONFIGURATION**.

This menu lists the available links that may be configured as the transport layer of a Frame Relay interface. Use the menu shown below (First select the link and press **Enter**) to edit link's settings.

|                                                                                      |                          |
|--------------------------------------------------------------------------------------|--------------------------|
| BRICK Setup Tool                                                                     | BinTec Communications AG |
| [FRAME RELAY][LINK][EDIT]: Frame Relay Link Configuration                            | MyBRICK                  |
| <p>Link<br/>Line Management<br/>Mode</p> <p>Advanced Settings</p> <p>SAVE CANCEL</p> |                          |
| Press <Ctrl-n>, <Ctrl-p> to scroll, <Return> to edit/select                          |                          |

| Field                  | Meaning                                                                                                                                            |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Link</b>            | Shows the link that is currently being edited.                                                                                                     |
| <b>Line Management</b> | Determines whether or not link management is being performed on this link. Currently, the method described in Q.933 is supported.                  |
| <b>Mode</b>            | Defines the mode (DTE or DCE) the <b>BRICK</b> operates at for this connection. Note that one side of the link must operate as DTE and one as DCE. |

Table 7-4: **FR** ► **LINK CONFIGURATION**

► Go to **ADVANCED SETTINGS**.

This menu can be used to configure special settings relating to line management for Frame Relay interfaces on the **BRICK**. Some options only apply to **BRICK** operating in DTE or DCE mode.



| BRICK Setup Tool                                                       |        | BinTec Communications AG |
|------------------------------------------------------------------------|--------|--------------------------|
| [FRAME RELAY][LINK][EDIT][ADVANCED]: Advanced Link Configuration BRICK |        |                          |
| Supported Virtual Channels                                             | 250    |                          |
| Polling Interval                                                       | 10     |                          |
| Full Enquiry Interval                                                  | 6      |                          |
| Idle Interval                                                          | 15     |                          |
| Error Threshold                                                        | 3      |                          |
| Monitored Events                                                       | 4      |                          |
| OK                                                                     | CANCEL |                          |
| Enter integer range 1...250                                            |        |                          |

| Field                             | Meaning                                                                                                                                                                                                         |
|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Supported Virtual Channels</b> | This field can be used to control how many Virtual Channels this Link supports; a maximum of 250 (default) VCs are possible.                                                                                    |
| <b>Polling Interval</b>           | When set for DTE mode (client) and q933a line management is enabled this field determines the number of seconds between successive status enquiry messages sent out by the <b>BRICK</b> . (Default 10 seconds). |
| <b>Full Enquiry Interval</b>      | When set for DTE mode (client) and q933a line management is enabled this field determines the number of status enquiry intervals that pass before issuing a full status enquiry message (default 6 intervals).  |
| <b>Idle Interval</b>              | When set for DCE mode (server) and line management is enabled this field defines the number of seconds within a status enquiry messages should be received (default 15 seconds).                                |

| Field                   | Meaning                                                                                                                                                                                                                                                                                                                                  |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Error Threshold</b>  | When line management is enabled, this field defines the maximum number of unanswered Status Enquiries the <b>BRICK</b> accepts before declaring the interface down (default 3 messages).                                                                                                                                                 |
| <b>Monitored Events</b> | When line management is enabled this field defines the number of status polling intervals over which the error threshold (previous field) is counted. For example, if within <b>MonitoredEvents</b> number of events the station receives <b>ErrorThreshold</b> number of errors, the interface is marked as down (default 4 intervals). |

Table 7-5: **FR** ➤ **LINK CONFIGURATION** ➤ **ADVANCED SETTINGS**

➤ Go to **SWITCHING**.

This menu is used to configure frame relay switching functionality on the **BRICK**. When used as a Frame Relay switch this menu can be used to configure routes, or mappings (i.e., from incoming interface/DLCI to outgoing interface DLCI).

Frame Relay routes can be added, removed, or changed here.

| Source    |      | Destination |        | Bc | Be   | Throughput |
|-----------|------|-------------|--------|----|------|------------|
| Interface | DLCI | Interface   | DLCI   |    |      |            |
|           | ADD  |             | DELETE |    | EXIT |            |

➤ Select **ADD** to create a new Frame Relay route.

- Select **DELETE** to remove a Frame Relay route entry that has been tagged (using the spacebar) for deletion.
- Select **EXIT** to accept the list of Frame Relay routes and return to the previous menu.  
To edit a Frame Relay route, highlight the entry and then enter **Return**. When adding or changing an entry the following information must be provided.

| Field                                                   | Meaning                                                                                                               |
|---------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| <b>Source Interface</b>                                 | Use the spacebar and scroll through the list of Frame Relay interfaces to select the source interface for this route. |
| <b>Source DLCI</b>                                      | Defines the DLCI of the source interface for this route.                                                              |
| <b>Destination Interface</b>                            | Use the spacebar to scroll through the list of Frame Relay interfaces and select the destination interface.           |
| <b>Destination DLCI</b>                                 | Use the spacebar to scroll through the list of Frame Relay interfaces and select the destination interface.           |
| <b>Committed Burst Rate</b><br>(Abbreviated <b>Bc</b> ) | This field defines the maximum amount of data (in bits) to transfer under normal conditions.                          |
| <b>Excess Burst Rate</b><br>(Abbreviated <b>Be</b> )    | This field defines the maximum amount of uncommitted data (in bits) to attempt deliver.                               |
| <b>Throughput</b>                                       | This field defines the physical throughput for this interface (and defaults to <b>ifSpeed</b> ).                      |

Table 7-6: **FR** ➤ **SWITCHING**

- Go to **MULTIPROTOCOL OVER FRAME RELAY**.  
This menu lists Multiprotocol Routing over Frame Relay interfaces on the **BRICK**. MPFR interfaces can be added, removed, or changed here.

|                                                      |        |                          |  |
|------------------------------------------------------|--------|--------------------------|--|
| BRICK Setup Tool                                     |        | BinTec Communications AG |  |
| FRAME RELAY][MPR]: Frame Relay Multiprotocol Routing |        | MyBRICK                  |  |
| Interface Name                                       | Type   |                          |  |
| ADD                                                  | DELETE | EXIT                     |  |

| Field                 | Meaning                                                                                        |
|-----------------------|------------------------------------------------------------------------------------------------|
| <b>Interface Name</b> | Identifies the interface name (taken from the <b>ifDescr</b> object from the <b>ifTable</b> ). |
| <b>Type</b>           | Specifies whether the interface is a point-to-point, or point-to-multipoint interface.         |

Table 7-7: **FR** ► **MULTIPROTOCOL OVER FRAME RELAY**

- ADD** ► Go to **ADD**.  
 This menu is used to create (or change) MPFR (Multi-Protocol routing over Frame Relay) interfaces on the **BRICK**.

|                                                            |  |                          |  |
|------------------------------------------------------------|--|--------------------------|--|
| BRICK Setup Tool                                           |  | BinTec Communications AG |  |
| [FRAME RELAY][MPR][ADD]: Configure Frame Relay MPR Partner |  | MyBRICK                  |  |
| Partner Name                                               |  |                          |  |
| Interface Type                                             |  | multipoint               |  |
| Inverse ARP                                                |  | enabled                  |  |
| Virtual Circuits>                                          |  |                          |  |
| IP>                                                        |  |                          |  |
| IPX>                                                       |  |                          |  |
| SAVE                                                       |  | CANCEL                   |  |
| Enter string, max length = 25 chars                        |  |                          |  |

| Field                 | Meaning                                                                         |
|-----------------------|---------------------------------------------------------------------------------|
| <b>Partner Name</b>   | Define a unique name to identify this MPFR partner.                             |
| <b>Interface Type</b> | Determines the interface type as being either “multipoint” or “point to point”. |
| <b>Inverse Arp</b>    | Enables/disables inverse ARP over this interface.                               |

Table 7-8: **FR** ➤ **MULTIPROTOCOL OVER FRAME RELAY** ➤ **MULTIPROTOCOL ROUTING**

➤ Go to **VIRTUAL CIRCUITS**.

This menu should only be used by sites receiving multiple DLCIs from their Frame Relay service provider. Depending on the number of DLCIs and type of service being received use this menu to define the appropriate data rates.

| Source Interface |      | Destination Interface |        | BC | Be | Throughput |
|------------------|------|-----------------------|--------|----|----|------------|
| Interface        | DLCI | Interface             | DLCI   |    |    |            |
|                  | ADD  |                       | DELETE |    |    | EXIT       |

| Field                       | Meaning                                                                                  |
|-----------------------------|------------------------------------------------------------------------------------------|
| <b>Source Interface</b>     | Using the spacebar, scroll through the list of Frame Relay interfaces.                   |
| <b>Source DLCI</b>          | Defines the DLCI used on this interface.                                                 |
| <b>Committed Burst Rate</b> | The maximum amount of data that is guaranteed to be transferred by the service provider. |
| <b>Excess Burst Rate</b>    | The amount of additional data that is uncommitted by the service provider.               |
| <b>Throughput</b>           | The physical throughput of this interface.                                               |

Table 7-9: **FR** ► **MULTIPROTOCOL OVER FRAME RELAY** ► **VIRTUAL CIRCUITS**

- IP** ► Go to **IP**.  
This is where you configure the IP settings for this remote MPFR partner.



The settings used in this menu are the same as those used in the **WAN PARTNER** ► **ADD** ► **IP** menu described in the User's Guide but only apply to this MPFR partner.

- IPX** ► Go to **IPX**.  
This is where you configure the IP settings for this remote MPFR partner.



The settings used in this menu are the same as those used in the **WAN PARTNER** ➤ **ADD** ➤ **IPX** menu described in the User's Guide but only apply to this MPFR partner.

## 7.5 Example Configuration using Setup Tool

### 7.5.1 Frame Relay over ISDN Lines

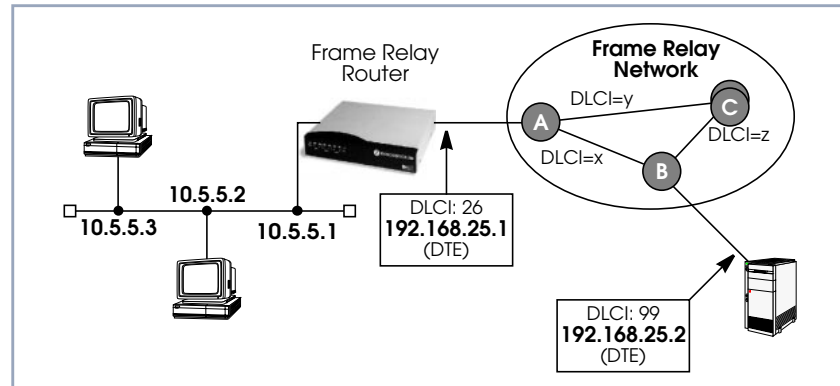


Figure 7-5: Scenario: Frame Relay over ISDN Lines

**Requirements** Frame Relay requires a separate license to be installed on the **BRICK**.

- After installing your license verify the Frame Relay is listed as “valid” in Setup Tool’s License menu (or the Status field for the frame\_relay entry in the **biboAdmLicInfoTable** shows valid\_license).

#### Step 1

- Define the physical interface** ➤ In Setup Tool’s main menu select the ISDN interface where the Frame Relay service is being received.



```

BRICK Setup Tool BinTec Communications AG
[WAN][ADD]: WAN Interface MyBRICK

Result of autoconfiguration: Euro ISDN, point to multipoint
ISDN Switch Type autodetect on bootup
D-channel dialup
B-channel dialup
B-channel dialup

Incoming Call Answering>
Advanced Settings>

 SAVE CANCEL

Use <Space> to select

```

- You should verify the **Result of autoconfiguration** field is correct. If this interface is a leased line or it was not properly detected set the Switch Type and D/B channel fields appropriately here and **SAVE** the settings.

### Step 2

#### Configure a new WAN Partner

- Create a new interface in the **WAN PARTNER** ➤ **ADD** menu. This step defines the (physical) link to the next switch in the Frame Relay network (host A shown above).

|                                       |                          |
|---------------------------------------|--------------------------|
| BRICK Setup Tool                      | BinTec Communications AG |
| [WAN][ADD]: Configure WAN Partner ( ) | MyBRICK                  |
| Partner Name                          | FRprovider               |
| Encapsulation                         | Frame Relay              |
| Encryption                            | none                     |
| Calling Line Identification           | no                       |
| WAN Numbers>                          |                          |
| PPP>                                  |                          |
| Advanced Settings>                    |                          |
| IP>                                   |                          |
| IPX>                                  |                          |
| BRIDGE>                               |                          |
| Use <Space> to select                 |                          |

- After defining a partner name select the **Encapsulation** *Frame Relay* and configure no other protocol. Under **WAN Numbers** select the ISDN port (from step 1) to use and **SAVE** the settings.

### Step 3

#### Configure the Frame Relay Link Settings

- Go to the **FR** ➤ **LINK CONFIGURATION** menu and select the physical link (partner name) you configured in the previous step and press enter to set the desired parameters. It is very important that you set the **Mode** field to *dte* here if the **BRICK** is operating as a Frame Relay router.

```

BRICK Setup Tool BinTec Communications AG
[FRAME RELAY][LINK][EDIT]: Frame Relay Link Configuration MyBRICK

Link FRprovider
Line Management none
Mode dte

Advanced Settings>

 SAVE CANCEL

Use <Space> to select

```

- Optionally, you can define whether Link Management should be performed for this link. If Link management is to be performed on this link, several options are available via the Advanced Settings sub-menu that control how often various LMI packets to send to the server (DCE) and the intervals at which these enquiries are sent.

#### Step 4

#### Configure the Multi-Protocol Routing Interface

- Go to the **MULTIPROTOCOL OVER FRAME RELAY** menu and select **ADD** to create a new MPFR (Multi-Protocol routing over Frame Relay) partner interface. This step will define the virtual interface to the end-system (host at IP address 192.168.25.2 in the diagram above) IP packets will be routed to/from.



When enabling protocols to route over Frame Relay please note that at current, only IP over Frame Relay has been tested on the **BRICK**.

|                                                            |                |                          |  |
|------------------------------------------------------------|----------------|--------------------------|--|
| BRICK Setup Tool                                           |                | BinTec Communications AG |  |
| [FRAME RELAY][MPR][ADD]: Configure Frame Relay MPR Partner |                | MyBRICK                  |  |
| Partner Name                                               | FRpartner      |                          |  |
| Interface Type                                             | point to point |                          |  |
| Inverse Arp                                                | disabled       |                          |  |
| Virtual Circuits>                                          |                |                          |  |
| IP>                                                        |                |                          |  |
| IPX>                                                       |                |                          |  |
|                                                            | SAVE           | CANCEL                   |  |
| Enter string, max length = 25 chars                        |                |                          |  |

**Step 5****Select Frame Relay Interface**

➤ Go to **VIRTUAL CIRCUITS** ➤ **ADD** to select the interface to use for the Frame Relay partner.

|                                                                                 |       |                          |  |
|---------------------------------------------------------------------------------|-------|--------------------------|--|
| BRICK Setup Tool                                                                |       | BinTec Communications AG |  |
| [FRAME RELAY][MPR][ADD][Switching][ADD]: Configure Frame Relay Virtual Circuits |       | MyBRICK                  |  |
| Source Interface                                                                | xi2   |                          |  |
| Source DLCI                                                                     | 16    |                          |  |
| Committed Burst Rate                                                            | 64000 |                          |  |
| Excess Burst Rate                                                               | 0     |                          |  |
| Throughput                                                                      | 64000 |                          |  |
|                                                                                 | OK    | CANCEL                   |  |
| Use <Space> to select                                                           |       |                          |  |

The most important setting, however, is the following (see [table 7-9, page 278](#) for the description of the Virtual Circuit parameters):

| Field                   | Meaning                                                                               |
|-------------------------|---------------------------------------------------------------------------------------|
| <b>Source Interface</b> | In this field one of the WAN Partners with Frame Relay encapsulation can be selected. |

Table 7-10: **FR** ► **MULTIPROTOCOL OVER FRAME RELAY** ► **ADD** ► **VIRTUAL CIRCUITS** ► **ADD**

### Step 6

#### Configure IP settings for MPFR Interface

- In the **IP** submenu configure the IP settings for the remote Frame Relay end station (192.168.25.2 in our example diagram). A transit network is optional. Select **SAVE** to ensure your Frame Relay setup is saved to a configuration file.

|                                                      |                          |
|------------------------------------------------------|--------------------------|
| BRICK Setup Tool                                     | BinTec Communications AG |
| [FRAME RELAY][MPR][IP]: IP Configuration (FRpartner) | MyBRICK                  |
| IP Transit Network                                   | no                       |
| Partner's LAN IP Address>                            | 192.168.25.2             |
| Partner's LAN IP Netmask>                            | 255.255.255.0            |
| Advanced Settings>                                   |                          |
| SAVE                                                 | CANCEL                   |
| Enter string, max length = 25 chars                  |                          |

