Extended Feature Reference

Version 1.2
Document #71050A

February 1999

NOTE

The information in this manual is subject to change without notice.

This manual provides a complete description of all the complex, separately licensable features available for the BinTec BIANCA/BRICK and BinGO! routers. The information included in this manual is compatible with software version 4.9.

While every effort has been made to ensure the accuracy of all information in this document, 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 and the BinTec logo are registered trademarks of BinTec Communications AG.

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

- lb -am

BinTec Communications AG February 1999

CONTENTS

Contents

Introduction

OSPF

RADIUS

Token Authentication Firewall

Virtual Private Networking

X.25

Frame Relay

1

INTRODUCTION

What's covered

How to contact BinTec Communications

Ways to contact BinTec	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 GERMANY
WWW	http://www.BinTec.de

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 you BRICK or BinGO router.
- Release notes for upgrading your system software.
- Windows software and UNIXTools applications.

About your User Documentation

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

This document describes extended features available on BIANCA/ BRICK and BinGO! routers 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.

What’s covered in this guide

Chapter 1 Introduction is this chapter.

Chapter 2 OSPF describes using the Open Shortest Path First interior routing protocol on your BinTec router.

Chapter 3 RADIUS descirbes using your BinTec router as a RADIUS Client.

Chapter 4 Token Authentication Firewall describes Token Authentication Firewall support on your BinTec router.

Chapter 5 Virtual Private Networking describes using your BinTec router to implement Virtual Private Networking.

Chapter 6 X.25 describes operating your BinTec router in an X.25 environment.

Chapter 7 Frame Relay describes using your BinTec router as Frame Relay router.

Conventions used in this guide

To help you locate and interpret information easily, this manual uses the following visual clues and typographic conventions.

Visual Clues	
✍	Lets you know what information you’ll need before you start to configure a feature.

Visual Clues

❗	Marks the beginning of a list of steps required to configure a feature.
❓	References to information in other sections or documents that may be helpful.
⚠	Points out important information such as safety precautions and common pitfalls.

Typographic Conventions

`Bold constant width` type represents characters or text that you must type in, exactly as shown.

Bold italic type represents special system table names.

Text enclosed in a box like this SYSTEM represents a submenu or menu command found in Setup Tool.

2

OSPF

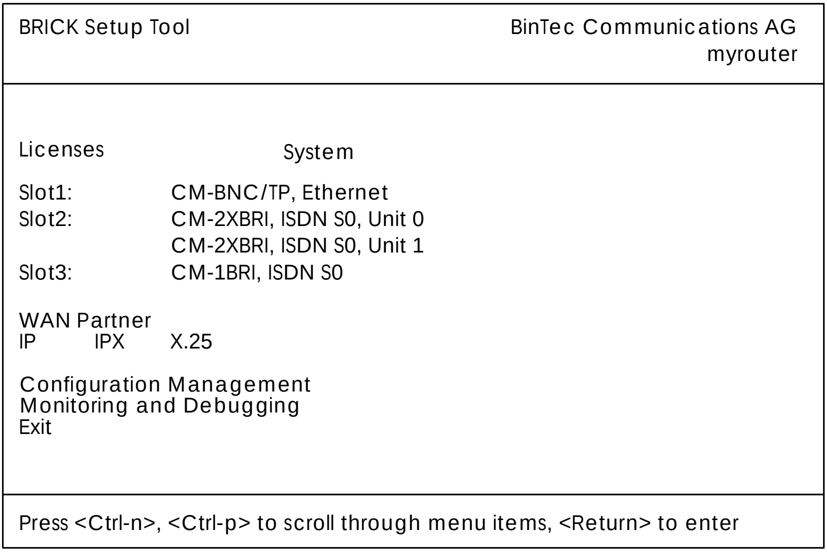
What's covered

In this chapter we'll describe the Setup Tool menus and settings you'll see while using Setup Tool to configure the OSPF protocol on your router.

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

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.

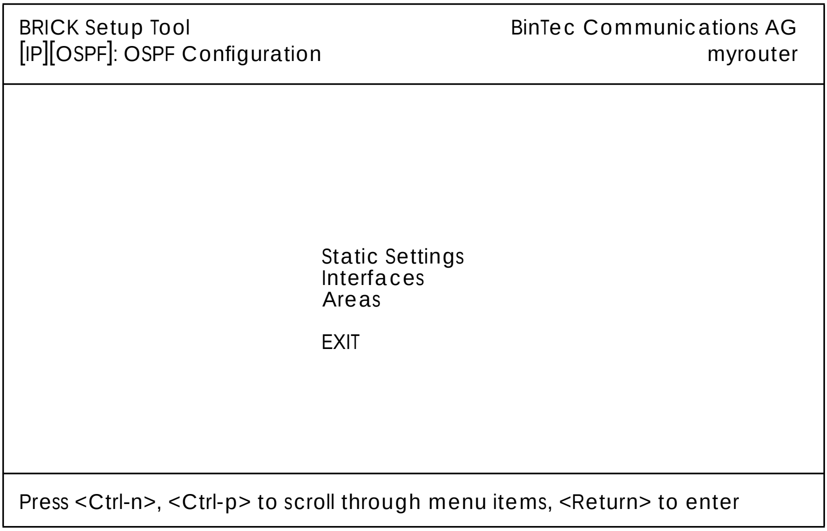


```
BRICK Setup Tool                                BinTec Communications AG
                                                                myrouter


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
```

IP → OSPF → This is the starting point for all OSPF settings.

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                                  myrouter




                            Static Settings
                            Interfaces
                            Areas

                            EXIT




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

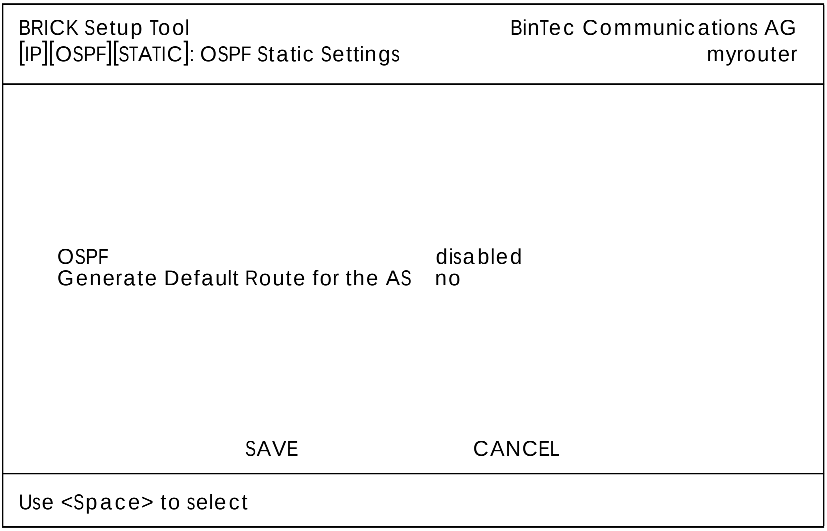
STATIC SETTINGS contains global OSPF parameters. This is where OSPF is enabled on the router.

INTERFACES lists all OSPF capable router interfaces and is used for configuring interface-specific settings.

AREAS lists all known OSPF areas and used for adding/configuring area-specific settings.

IP → OSPF → STATIC SETTINGS

This menu contains global settings for the OSPF protocol.



```
BRICK Setup Tool                                    BinTec Communications AG
[IP][OSPF][STATIC]: OSPF Static Settings                            myrouter




    OSPF                                  disabled
    Generate Default Route for the AS     no




                     SAVE                           CANCEL

Use <Space> to select
```

OSPF = 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 yes the router advertises a default route over all active OSPF interfaces (See the “Admin Status” field in the **IP → OSPF → INTERFACES** menu.).

Note: Special consideration should be made when 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.

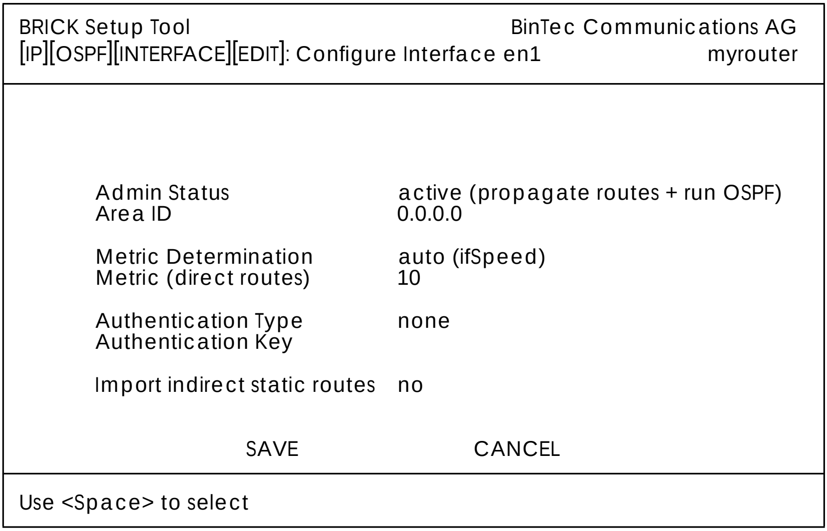
Select **SAVE** to accept the settings and return to the previous menu.

Select **CANCEL** to discard all changes made since the last SAVE and return to the previous menu.

IP → OSPF → INTERFACES

This menu lists the router interfaces OSPF can be configured for. 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 hit enter. The fields shown in the resulting EDIT menu shown below can be configured separately for each interface.



```
BRICK Setup Tool                                      BinTec Communications AG
[IP][OSPF][INTERFACE][EDIT]: Configure Interface en1                 myrouter



      Admin Status                    active (propagate routes + run OSPF)
      Area ID                         0.0.0.0

      Metric Determination            auto (ifSpeed)
      Metric (direct routes)          10

      Authentication Type             none
      Authentication Key

      Import indirect static routes   no

                       SAVE                       CANCEL

Use <Space> to select
```

Admin Status = The status of an OSPF interface defines whether routes and/or OSPF protocol packets are propagated over the interface.

If OSPF hasn't been enabled yet only the Admin Status field is displayed (in which case changes are irrelevant).

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:

Active   OSPF is running over this interface.

Passive	OSPF is not running over this interface. OSPF protocol packets are neither sent or received over the interface, however this interface may be included in other Router Links.
Off	OSPF is not running over this interface and this interface is not included in Router Links.

Note: ⚠ 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.

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.

Determination	Meaning
auto	The metric = the value of the base metric which is based on the bandwidth (***ifSpeed***) of the interface.
fixed	The metric defined (configurable) in the following field is always used (no adjustment).
auto + adjust[1]	When the dial-up interface is in the up state, the metric = <*base metric value*> – 10. Otherwise, metric = <*base metric value*>.
fixed + adjust[a]	When the dial-up interface is in the up state the metric = <*base metric value*> – 10. Otherwise metric = <*base metric value*>.

[1] Only valid for Dial-up interfaces.

Metric = Identifies the base metric value, or cost of this interface. For auto determination values (see above) the actual metric used is adjusted starting a base metric value which is a simple function of the bandwidth of the physical medium. All interfaces (except leased line interfaces) use the function.

$$\text{Base Metric Value} = \frac{\mathbf{1000{,}000{,}000}}{\langle \textit{bandwidth in bps} \rangle}$$

This results in 10 for ethernet, 6 for token ring, and 1562 for dialup ISDN interfaces (1 B-Channel). Note that 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).

For **fixed determination** values (see previous field) the base metric value can be configured here.

Authentication Type = 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.

By default this is set to none. With simple, Key is transmitted as a text string in each packet. With md5, Key is used to create (verify) an encrypted digest which is sent with each packet.

Authentication Key = A text string to use in connection with the Authentication Type set above.

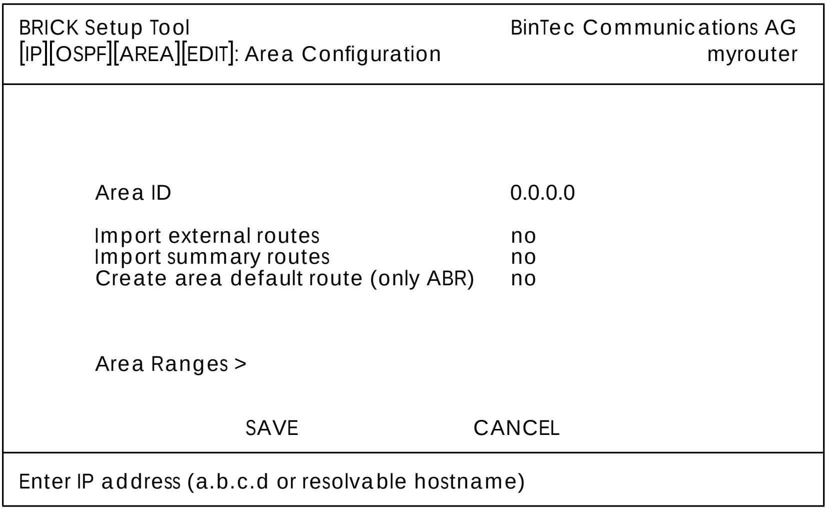
Import indirect static routes = If set to no (default) only direct routes for this interface are propagated over active OSPF interfaces (See the Admin Status field). When set to yes, indirect static routes are also propagated over active interfaces and are contained in external advertisements.

Note: ⚠ Although practical for sites using WAN interfaces without transfer networks caution should be made to avoid routing loops when importing indirect static routes.

IP → OSPF → AREAS

This menu lists the OSPF Areas known to the router. 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 hit enter.



```
BRICK Setup Tool                                  BinTec Communications AG
[IP][OSPF][AREA][EDIT]: Area Configuration                        myrouter



       Area ID                                    0.0.0.0

       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)
```

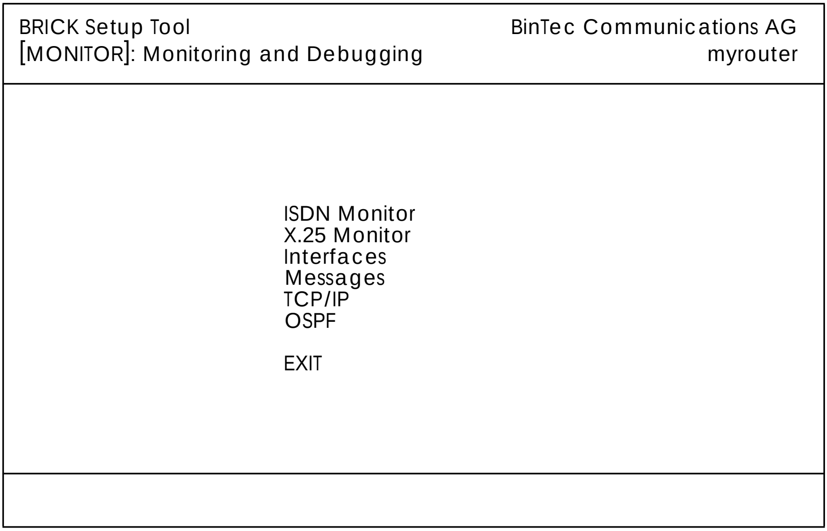
Area ID = Identifies the OSPF Area this entry corresponds to. The backbone area is 0.0.0.0.

Import external routes = Specifies whether external routes should be imported for this area. When set to no, this Area is defined as an OSPF Stub Area.

Area Ranges = This submenu specifies IP Address ranges for route condensation among areas.

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                                BinTec Communications AG
[MONITOR]: Monitoring and Debugging                             myrouter




                    ISDN Monitor
                    X.25 Monitor
                    Interfaces
                    Messages
                    TCP/IP
                    OSPF

                    EXIT
```

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.

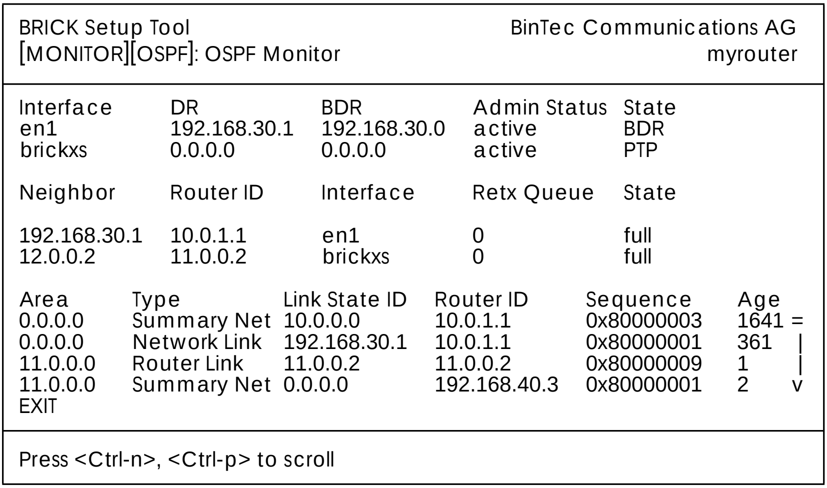
TCP/IP menu lets you monitor IP traffic by protocol.

OSPF menu lets you monitor OSPF related information.

Select **EXIT** to return to the main menu.

MONITORING AND DEBUGGING → OSPF

The OSPF monitor is divided horizontally in three sections and displays information relating to OSPF Interfaces, Neighbours, and Areas.



```
BRICK Setup Tool                                         BinTec Communications AG
[MONITOR][OSPF]: OSPF Monitor                                            myrouter

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       0x80000003     1641 =
0.0.0.0    Network Link  192.168.30.1   10.0.1.1       0x80000001     361  |
11.0.0.0   Router Link   11.0.0.2       11.0.0.2       0x80000009     1    |
11.0.0.0   Summary Net   0.0.0.0        192.168.40.3   0x80000001     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.)

Interface = The router interface the entry corresponds to.

DR = The Designated Router’s IP address on this interface (A DR is not shown for Point-To-Point interfaces).

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 on page 9).

State = The OSPF status (***ospfIfState***) of the interface shown here may be:

down	OSPF is not running on this interface.
wait	The initial phase of OSPF where DR and BDR are determined.

PTP The interface is a Point-To-Point interface.
No DR or BDR is shown.

DR The router is the Designated Router for this interface.

BDR The router is the Backup Designated Router for this interface.

DRother Another router is the DR/BDR for this interface.

Neighbour Section

The Neighbour section lists the OSPF neighbour routers that have been identified via the HELLO protocol.

Neighbor = The neighbour router's address on this interface.

Router ID = The neighbour 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 neighbour. This is the number of advertisements that need to be sent to (and acknowledged from) this neighbour.

State = The state of OSPF with this neighbour router may be:

init The initial phase. A HELLO packet was received from this neighbour.

twoWay Bidirectional communication with the neighbour. Transmitted HELLO packets have been accepted by the neighbour router (parameters are correct).

EXstart The exchange of Database Description Packets between the router and neighbour has begun.

exchangeActively exchanging Database Description Packets with the neighbour router.

loading The router and the neighbour router are now exchanging Link State Advertisements.

full The router and neighbour routers' Link State Database are now synchronized.

LSDB Section

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

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.

Overview of the OSPF Protocol

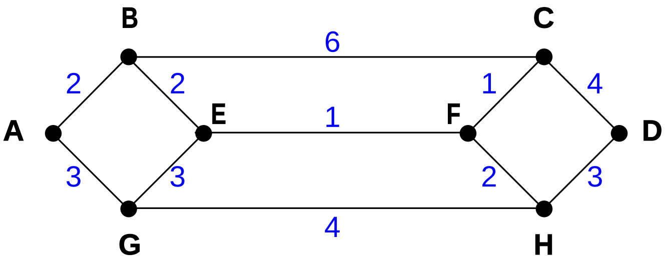
OSPF (Open Shortest Path First), is an interior routing protocol that is often used by larger network installations as an alternative to RIP. It was originally designed to address some of the limitations of RIP (when used in larger networks). Some of the problems (with RIP) that OSPF addresses include:

- **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 initialisation 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.

Shortest Path Routing

With RIP, routes are measured and selected according to number of hops it takes for a packet reach it’s 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.



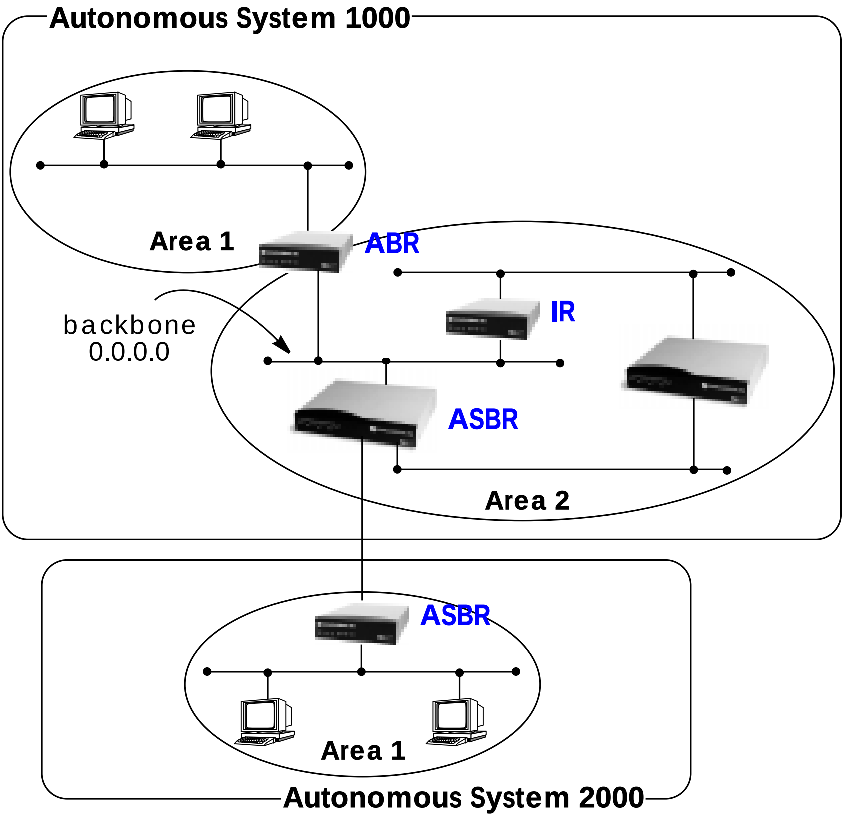
In OSPF each link has a cost associated with it (typically some fixed number divided by the bandwidth of the link). Routes are calculated and selected according to the least cost of the overall path a packet will travel. Thus in shortest-path routing the best path is also the fastest path (theoretically),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

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 of 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 Router Types) in an AS must have a physical connection to the backbone.

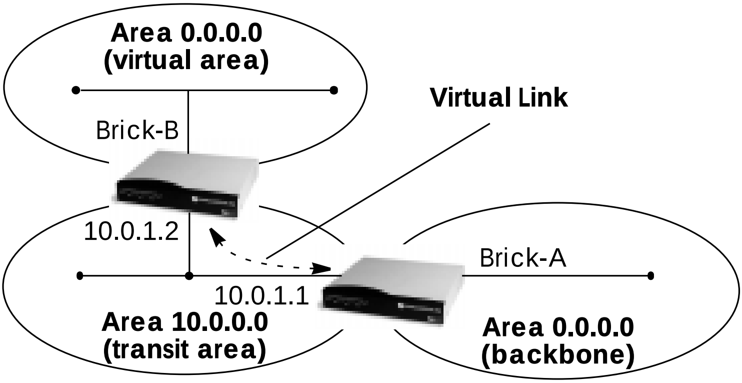


Any of the routers shown above could additionally be the Designated Router or Backup Designated Router for its respective network.

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's not possible to physically connect all areas to the backbone. By configuring a "Virtual Link" between two area border routers a remote area an 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.



Router Types

The location of a router's interfaces 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 synchronise 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.

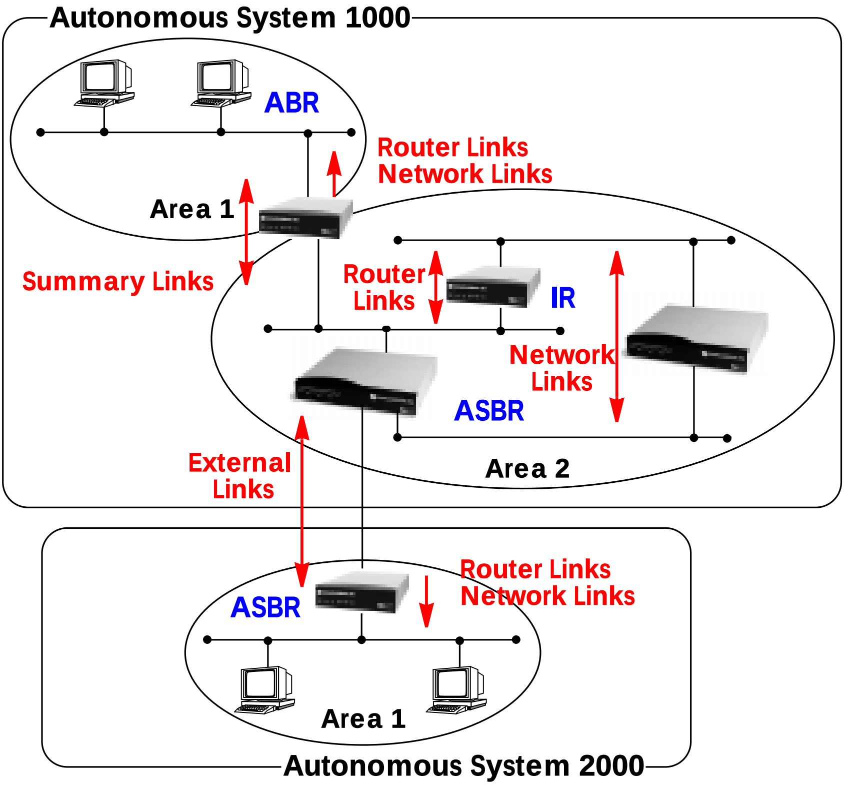
Link State Advertisement Types

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 DR (or BDR). **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.

LSA Type	Purpose:
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.



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.

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:

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

Neighbor Identification

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

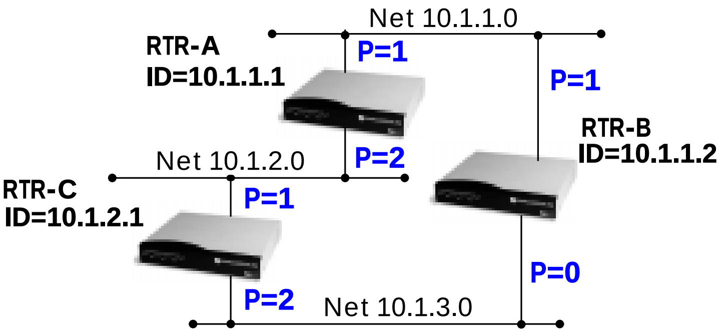
1. Share a common network.
2. Are using the same Area Number for that segment.
3. Are using the same Authentication for the segment.
4. Are using the same parameters (HELLO interval, etc.).

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

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.



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

Building up the LSD and the STP

Link-State Advertisements, contain information about a routers 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.

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 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.

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.

1. 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.
2. 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.

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

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:

1. The exchange of HELLO packets between neighbours is suppressed once the BRICK has synchronized its LSDB with that neighbour (A dial-up connection is initially opened to synchronize the database.).
2. Link State advertisements are only flooded to neighbour 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).

Note: This feature should only be used if all routers in the AS support this feature (RFC 1793) since some routers don't acknowledge the DC-bit (or use it differently). This could result in unwanted ISDN connections or connections.

Note: 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.

Example OSPF Installation

A typical network installation showing how OSPF could be put to use is shown in the diagram on the following page. 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 povided for each router.

Area 11.0.0.0 (stub area)

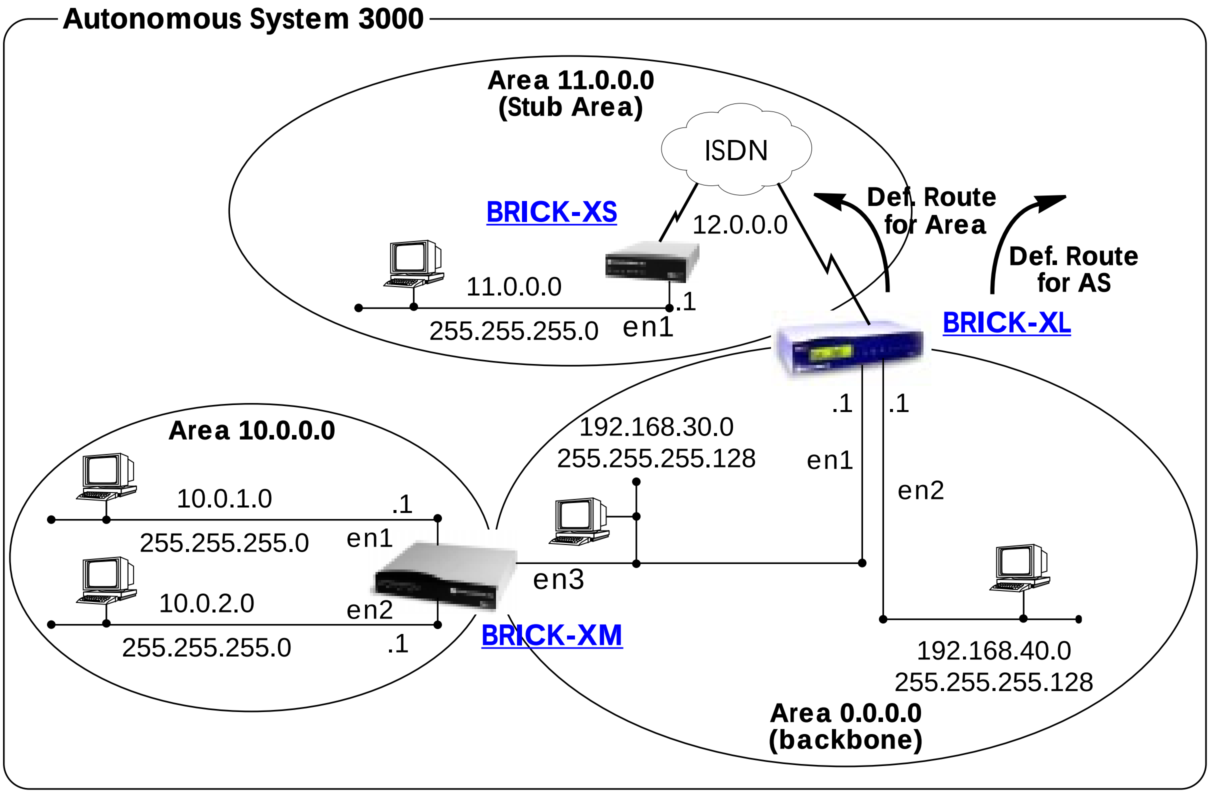
- Since the remote LAN in Area 11.0.0.0 is linked to the backbone via an ISDN dialup link this area is configured as a stub area. This means that external routing information advertisements won't flow into this area. The default route for this area is provided by the router BRICK-XL.
- Because OSPF on 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-XL 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.


Autonomous System 3000
Area 11.0.0.0
(Stub Area)
ISDN
BRICK-XS
12.0.0.0
Def. Route
for Area
Def. Route
for AS
11.0.0.0
.1
255.255.255.0
en1
BRICK-XL
.1
.1
192.168.30.0
255.255.255.128
en1
en2
Area 10.0.0.0
10.0.1.0
.1
255.255.255.0
en1
en3
10.0.2.0
en2
255.255.255.0
.1
BRICK-XM
192.168.40.0
255.255.255.128
Area 0.0.0.0
(backbone)

Configuration Overview

All BRICKs:

1. A valid OSPF license must be installed. This can be added to the ***biboAdmLicenseTable*** or from Setup Tool's LICENSES → menu.
2. OSPF must be enabled by setting ***ospfAdminStat*** to `enabled`, or from Setup Tool's IP → OSPF → STATIC SETTINGS → menu.

BRICK-XL Overview (details):

1. Create the dial-up partner interface to BRICK-XS.
2. Have BRICK-XL advertise the default route for the AS.
3. Create the Area entry for Area 11.0.0.0.
4. Assign the new dialup partner interface to Area 11.0.0.0 and set the interface to active.
5. Verify ethernet interfaces en1 and en2 are assigned to Area 0.0.0.0 and set both interfaces to active.

BRICK-XS Overview (details):

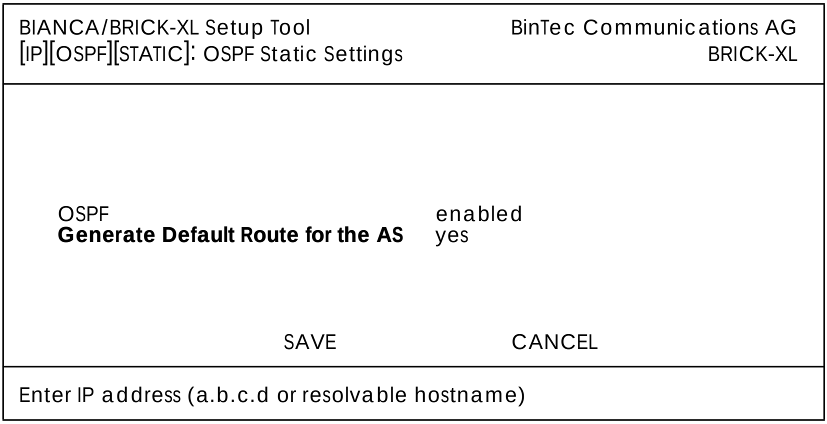
1. Create the dial-up partner interface to BRICK-XL.
2. Create the Area entry for Area 11.0.0.0.
3. Assign the ethernet interface (en1) to Area 11.0.0.0 and set the interface to active.
4. Assign the new dial-up interface to Area 0.0.0.0 and set the interface to active.

BRICK-XM Overview (details):

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

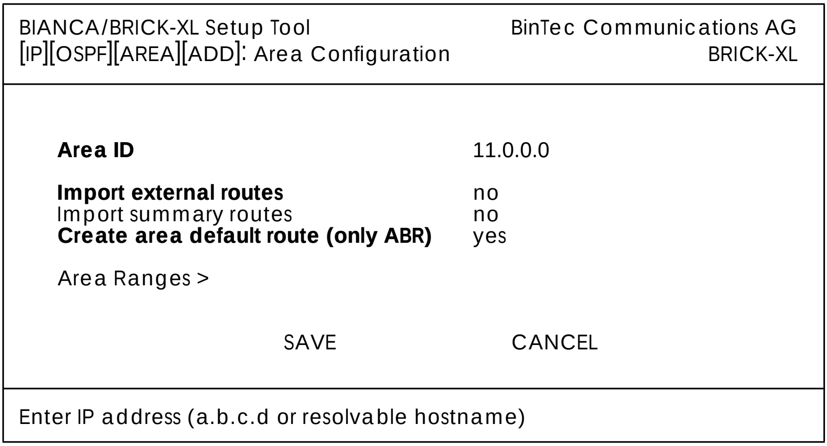
Configuration Steps for BRICK-XL

1. Assuming an OSPF license is installed and OSPF has been enabled the partner interface to BRICK-XS should be created. Note that our example uses a transfer network (network 12.0.0.0).
2. Since BRICK-XL should advertise the default route for the AS set this field to yes in IP → OSPF → STATIC SETTINGS →.



```
BIANCA/BRICK-XL Setup Tool                      BinTec Communications AG
[IP][OSPF][STATIC]: OSPF Static Settings                        BRICK-XL



   OSPF                                 enabled
   Generate Default Route for the AS    yes


                  SAVE                    CANCEL

Enter IP address (a.b.c.d or resolvable hostname)
```

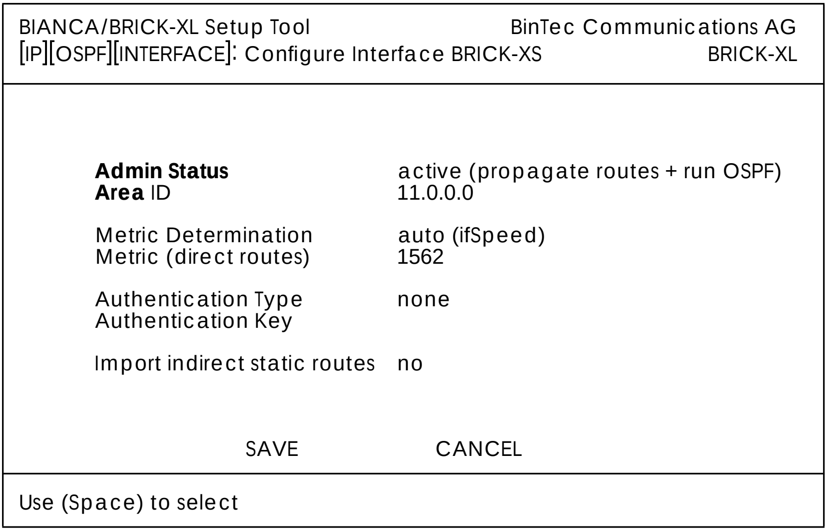
3. In the IP → OSPF → AREAS → menu create an entry for Area 11.0.0.0. Define this area as a Stub Area and have BRICK-XL generate the default route for this area.



```
BIANCA/BRICK-XL Setup Tool                      BinTec Communications AG
[IP][OSPF][AREA][ADD]: Area Configuration                       BRICK-XL

   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)
```

4. In the IP → OSPF → INTERFACES → menu locate the dialup interface entry created in step 1 and hit enter to edit the settings.

   Set the Admin Status to active and assign it to Area 11.0.0.0 (or the area created in step 3) and select SAVE .



```
BIANCA/BRICK-XL Setup Tool                          BinTec Communications AG
[IP][OSPF][INTERFACE]: Configure Interface BRICK-XS                 BRICK-XL



      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
```

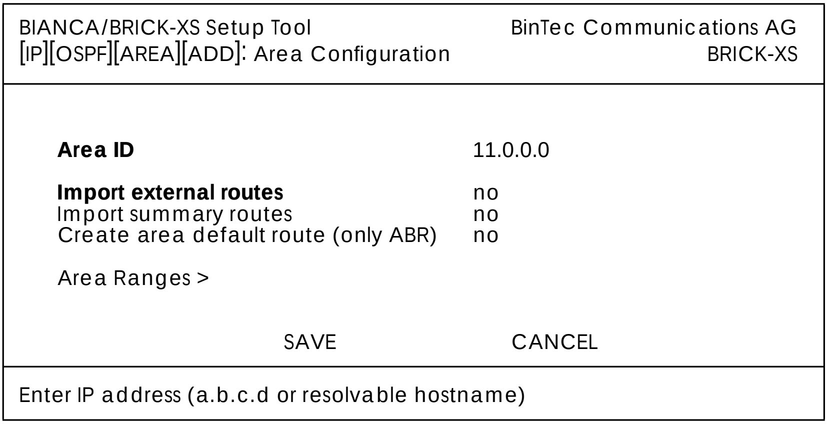
   By default, dial-up interfaces are set to passive in the Admin Status field.

5. 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 step 2) and select SAVE

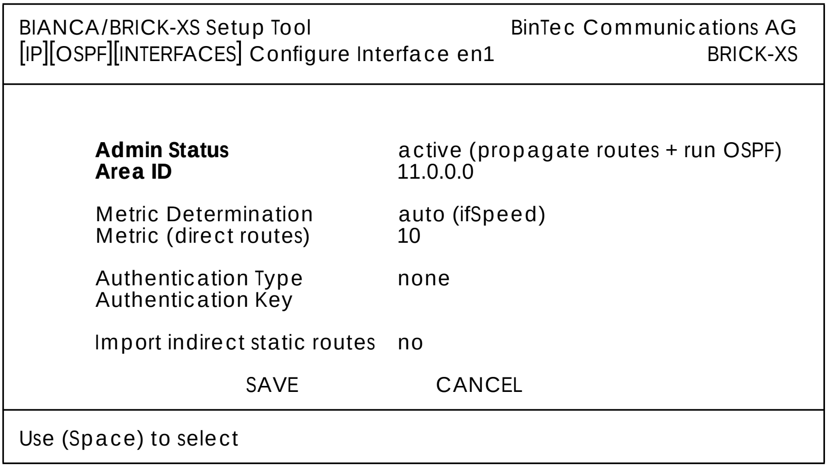
Configuration Steps for BRICK-XS

1. Assuming an OSPF license is installed and OSPF has been enabled the dial-up partner interface to BRICK-XL should be created. In our example a transfer network (12.0.0.0) is used.
2. In the IP → OSPF → AREAS → menu create Area 11.0.0.0. and define it as a Stub Area.



```
BIANCA/BRICK-XS Setup Tool                          BinTec Communications AG
[IP][OSPF][AREA][ADD]: Area Configuration                           BRICK-XS


   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)
```

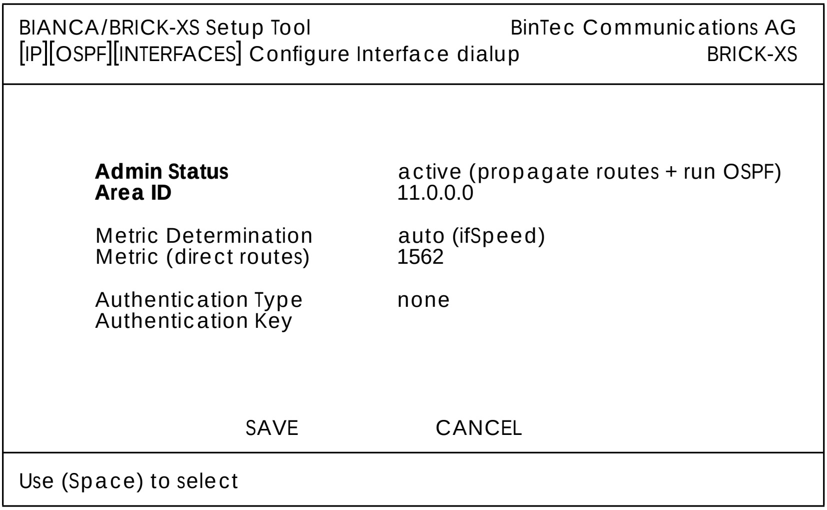
3. In the IP → OSPF → INTERFACES → menu assign the ethernet interface (en1) to Area 11.0.0.0 and make sure the Admin Status is set to active.



```
BIANCA/BRICK-XS Setup Tool                          BinTec Communications AG
[IP][OSPF][INTERFACES] Configure Interface en1                      BRICK-XS


      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 static routes  no

                  SAVE                  CANCEL

Use (Space) to select
```

4. 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 step 2).

   Set the Admin Status for the dialup interface to active and select SAVE.

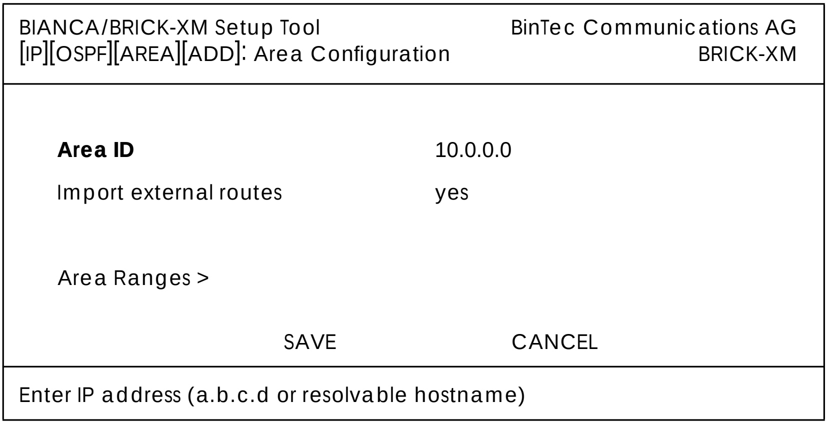


```
BIANCA/BRICK-XS Setup Tool                          BinTec Communications AG
[IP][OSPF][INTERFACES] Configure Interface dialup                   BRICK-XS


      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
```

Configuration Steps for BRICK-XM

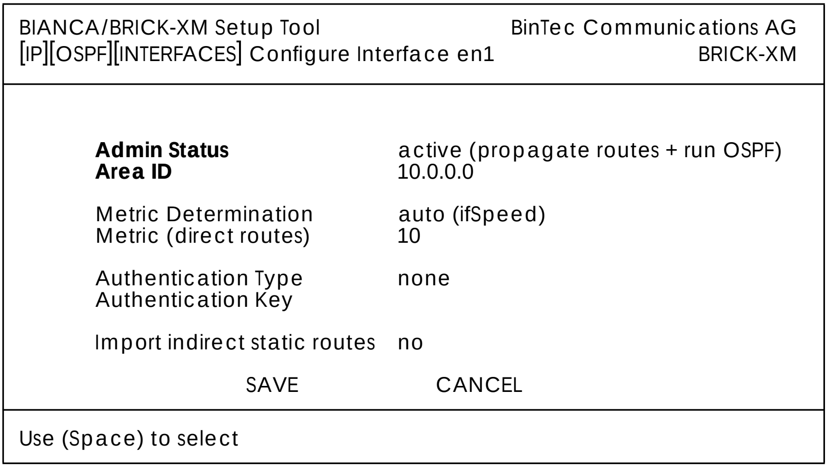
1. An OSPF license must already be installed and OSPF should been enabled IP → OSPF → STATIC SETTINGS → menu.

   Then create an area entry for Area 10.0.0.0 in the IP → OSPF → AREAS → menu.



```
BIANCA/BRICK-XM Setup Tool                      BinTec Communications AG
[IP][OSPF][AREA][ADD]: Area Configuration                       BRICK-XM


   Area ID                          10.0.0.0

   Import external routes           yes


   Area Ranges >


                      SAVE                 CANCEL

Enter IP address (a.b.c.d or resolvable hostname)
```

2. 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.



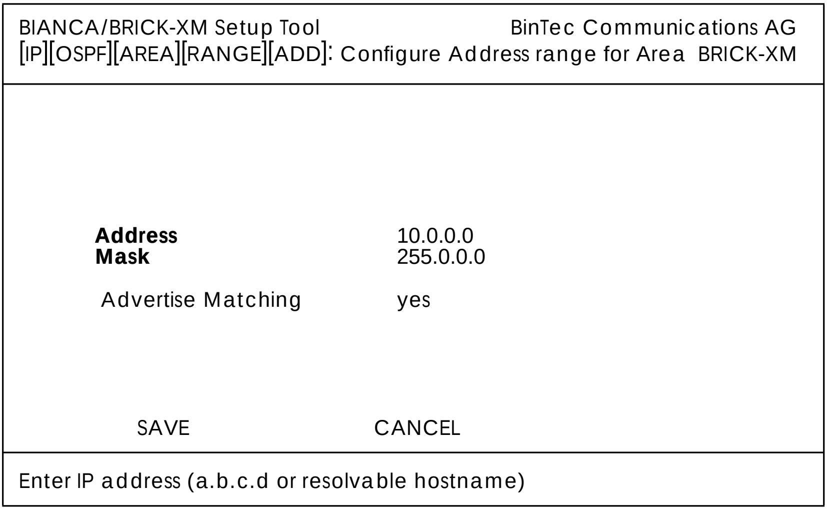
```
BIANCA/BRICK-XM Setup Tool                      BinTec Communications AG
[IP][OSPF][INTERFACES] Configure Interface en1                  BRICK-XM


      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 static routes  no

                   SAVE               CANCEL

Use (Space) to select
```

3. 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.

4. Return to the IP → OSPF → AREA → menu and scroll to the Area ID entry for the backbone and hit 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.*.*.*.



```
BIANCA/BRICK-XM Setup Tool                    BinTec Communications AG
[IP][OSPF][AREA][RANGE][ADD]: Configure Address range for Area  BRICK-XM



      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.

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 oher 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)
    RetransInterval(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***.

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).

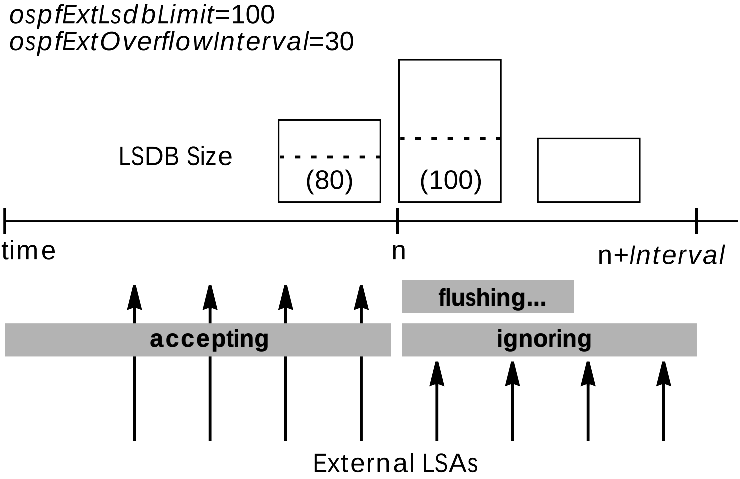
Once the limit is reached the BRICK goes into Overflow State. In Overflow State two things happen:

1. The BRICK begins to flush all external advertisements generated locally.
2. The BRICK ignores all new external advertisements.

NOTE: 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 yc 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.



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.

Note: Until a neighbour 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 synchronised, the HELLO protocol is then suppressed.

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		ipImportDstProto	
	default_route		✓[1]
	direct		
	static		✓[2]
	rip	–	
	ospf	✓[3]	–

1. ***ipImportSrcProto***=default_route ***ipImportDstProto***=ospf
   This entry forces an external Link State Advertisement to be generated that defines a default route for the Autonomous System.
2. ***ipImportSrcProto***=static ***ipImportDstProto***=ospf
   With this entry statically configured indirect routes will be propagated via OSPF as external LSAs.
3. ***ipImportSrcProto***=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 triggered an immediate broadcast of the entire routing table.

The remaining fields of ***ipImportTable*** allow for further control of how (and what) routing information is imported.

- ***ipImportMetric1***
  The metric in the context of the destination protocol the imported routes should get. If sset to -1 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 0.0.0.0, 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=X.X.0.0 and Mask=255.255.0.0 then addresses X.X.0.0 through X.X.255.255 are valid.
- ***ipImportEffect***
  Defines the effect of this entry. If set to “import”, importation from ***SrcProto*** to ***DstProto*** takes place. If set to “doNotImport” importation is prevented.
- ***ipImportIfIndex***
  Specifies the interface index of the interface for which the entry should be valid. If set to 0 the entry is valid for all interfaces.

3

RADIUS

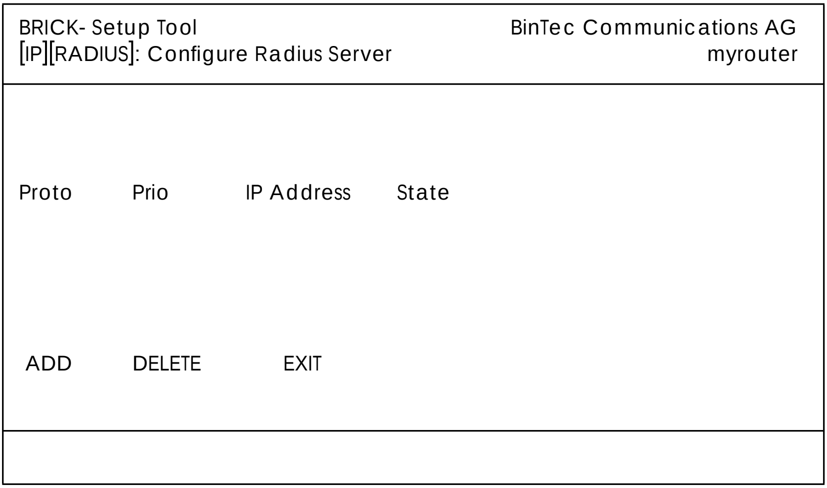
What's covered

In this chapter we'll cover all of the menus and settings you'll see while using Setup Tool to configure your router as a RADIUS Client for authentication and accounting.

Setup Tool Menus

IP → RADIUS SERVER

This menu lists all RADIUS Servers currently configured on the router.



```
BRICK- Setup Tool                                    BinTec Communications AG
[IP][RADIUS]: Configure Radius Server                                myrouter



Proto      Prio      IP Address    State




 ADD       DELETE        EXIT
```

IP → RADIUS SERVER →

This menu lists all the RADIUS Servers currently configured. You can add, edit, or delete list entries in the usual fashion.

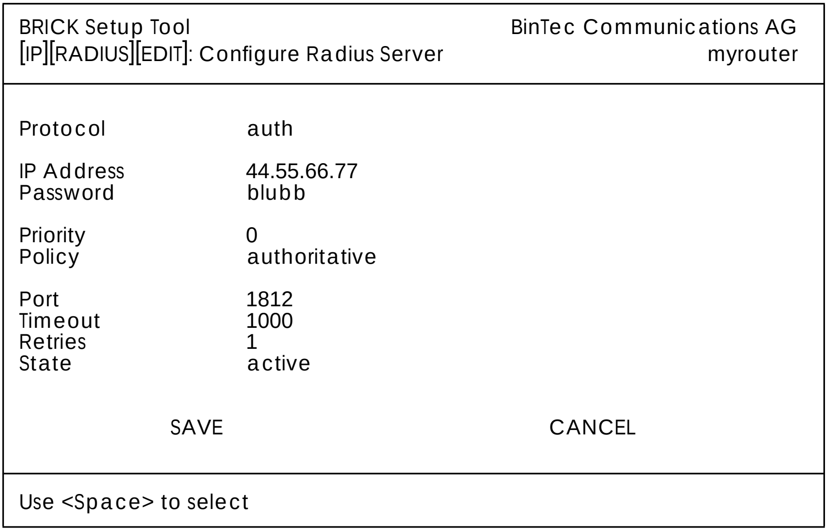
For each Radius Server you can configure the following parameters:

Protocol = Use this RADIUS Server for authentication purposes (**auth**) or for accounting ISDN connections (**acct**).
When you configure a RADIUS Server for accounting, the BRICK transmits Start and Stop Radius packets for each ISDN connection to this server.
Default value: auth

IP Address = IP Address of the RADIUS Server.

Password = Shared secret between RADIUS Server and BRICK.

Priority = 0 … 7. When there are several RADIUS Server entries, the server with the lowest priority entry is used first. If there is no reply from this server, the server with the next lowest priority entry is used,



```
BRICK Setup Tool                                  BinTec Communications AG
[IP][RADIUS][EDIT]: Configure Radius Server                       myrouter

Protocol                auth

IP Address              44.55.66.77
Password                blubb

Priority                0
Policy                  authoritative

Port                    1812
Timeout                 1000
Retries                 1
State                   active

            SAVE                              CANCEL

Use <Space> to select
```

and so forth, i.e. servers with *Priority*=**0** have the highest priority.
Default value: 0

Policy = can be set to **authoritative** or **non-authoritative**. If set to authoritative, a negative answer to a request will be accepted. This is not necessarily true when set to **non-authoritative**, where the next radius server will be asked until there is finally an **authoritative** server configured.
Default value: authoritative

Port = TCP port to use for RADIUS data. According to RFC 2138 the default ports are 1812 for authentication (was 1645 in older RFCs) and 1813 for accounting (1646 in older RFCs).
Default value: 1812

Timeout = 50 … 50000, number of milliseconds to wait for an answer to a request.
Default value: 1000 (1 second)

Retries = number of retries if a request is not answered. If after *Retries* attempts still no answer was received, the server *State* is set to **inactive**. The BRICK then tries to contact the Server every 20 seconds, and once the Server replies, the *State* is changed to **active** again.
Default value: 1

State = the state of the RADIUS Server. In normal operation mode this is either **active** (server answers requests) or **inactive** (server does not answer; see *Retries* above). You can also set State=**disabled**, to temporarily disable requests to a certain RADIUS Server.
Default value: active

RADIUS Overview

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 Packets

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.).
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.

RADIUS Server Files (UNIX)

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 attributes default behaviour.
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 1. define requirements for authenticating callers (password, PPP ID, Calling Line) and, 2. define the type of connections to establish if the user was 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.

Standard RADIUS Attributes

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 included on the Companion CD (the file is also available from our WWW server).

The following standard RADIUS attributes are available.

RADIUS Attribute	Type	R / A	Remark
User-Name	string	REQ	User name, mandatory inband: PPP partner name outband: PPP partner telephone number
User-Password	string	REQ	Password for PAP authentication
CHAP-Password	string	REQ	Password for CHAP authentication
NAS-Identifier	string	REQ	sysName of the BRICK
Service-Type	integer	ANS	Framed (for PPP) Callback-Framed (for PPP with Callback)
Framed-Protocol	integer	ANS	inband: PPP outband: PPP, X25, X25-PPP, IP-HDLC, IP-LAPB, MPR-LAPB MPR-HDLC, FRAME-RELAY, X31-BCHAN, X75-PPP, X75BTX-PPP, X25-NOSIG, X25-PPP-OPT
Framed-IP-Address	ipaddr	ANS	Partner IP address
Framed-IP-Netmask	ipaddr	ANS	Partner IP netmask
Framed-Routing	integer	ANS	None, RIPv1-Broadcast, RIPv1-Listen, RIPv1-Broadcast-Listen
Framed-Compression	integer	ANS	None, Van-Jacobson-TCP-IP
Framed-Route	string	ANS	You can create a route of the format ‹ipaddr›[/‹netmask bits›] ‹gateway› [‹metric1›...‹metric5›] e.g.: 192.2.3.4/24 193.141.54.1 1

RADIUS Attribute	Type	R / A	Remark
Idle-Timeout	integer	ANS	Shorthold
Port-Limit	integer	ANS	Number of B channels (== MaxConn)
Reply-Message	string	ANS	outband: ifDescr is set to this name (instead of using the telephone number)
Callback-Number	string	ANS	telephone number for Callback

The following RADIUS attributes are *not* yet applicable to your BRICK:

Acct-Authentic	CHAP-Challenge	Login-IP-Host
Acct-Delay-Time	Callback-Id	Login-LAT-Group
Acct-Input-Octets	Called-Station-Id	Login-LAT-Node
Acct-Input-Packets	Calling-Station-Id	Login-LAT-Service
Acct-Output-Octets	Filter-Id	Login-Port
Acct-Output-Packets	Framed-AppleTalk-Link	Login-Service
Acct-Session-Id	Framed-AppleTalk-Network	NAS-Port-Type
Acct-Session-Time	Framed-AppleTalk-Zone	Proxy-State
Acct-Status-Type	Framed-IPX-Network	Session-Timeout
Acct-Terminate-Cause	Framed-MTU	Termination-Action
		Vendor-Specific

BinTec Vendor Extensions

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

The following options are available at the moment:

Option	Type	Mode
BinTec-biboPPPTable	string	static
BinTec-ipExtIfTable	string	static
BinTec-ipRouteTable	string	dynamic
BinTec-ipExtRtTable	string	dynamic
BinTec-biboDialTable	string	dynamic
BinTec-ipNatPresetTable	string	dynamic

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> = "*variable*1=*value*1 … *variable*n=*value*n"

A few lines from a RADIUS setup 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.

Partner Recognition via CLID

To identify RADIUS partners outband by their CLID (calling line identification, i.e. ISDN telephone number) there has to be a corresponding entry in the RADIUS database, e.g.

```
9119732123    Service-Type = Framed,
              Framed-Protocol = IP-HDLC,
              Reply-Message = "partner1"
```

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 this number comes in a new PPP entry is generated with *Encapsulation*=**ip_hdlc** and *ifDescr*=**partner1**.

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.

At the moment it is not possible to use both inband and outband RADIUS authentication at the same time for one connection.

Channel Bundling

You can now bundle several B channels to achieve a higher data throughput using the *Port-Limit* option.

Note: ⚠ Certain RADIUS servers handle the setup of further B channels for a connection incorrectly.
This can result in very high charges!
So before using channel bundling for RADIUS make sure your RADIUS server is capable of handling it correctly.

RADIUS Table Entries

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

Default RADIUS UDP Port

The default UDP port used for RADIUS authentication is 1645.

4

TOKEN AUTHENTICATION FIREWALL

What’s covered

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

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

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.

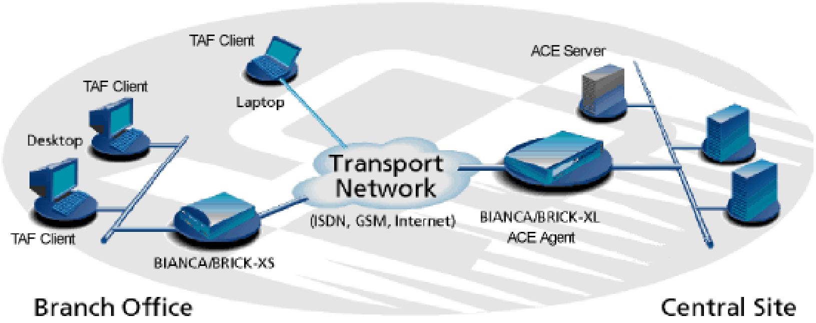


Figure 1: TAF Clients, ACE Agent and ACE Server

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 authentification. 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.

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 2: Token Card

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.

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.

Configuration of TAF

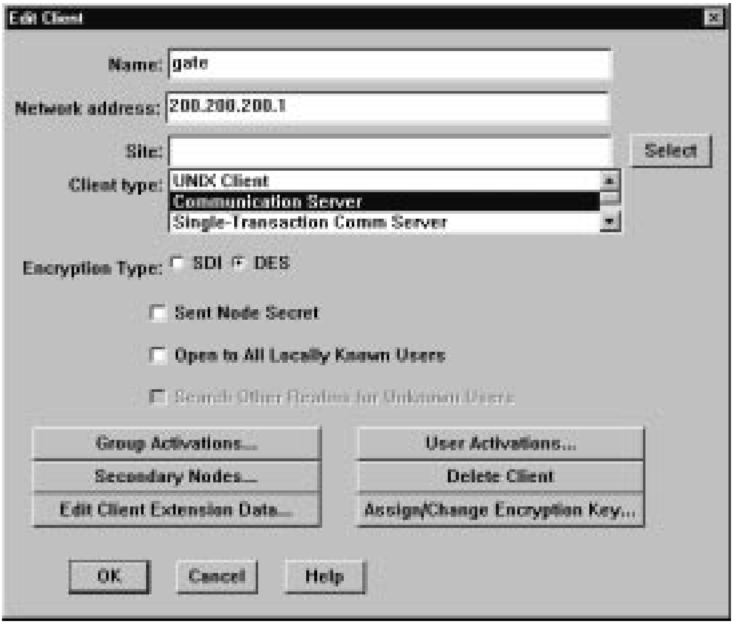
Configuring the ACE/Server

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.



Edit Client

Name: gate

Network address: 200.200.200.1

Site:

Client type: UNIX Client
Communication Server
Single-Transaction Comm Server

Encryption Type: ○ SDI ● DES

□ Sent Node Secret

□ Open to All Locally Known Users

□ Search Other Realms for Unknown Users

Figure 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 check box in the dialog shown above will appear selected (for detailed information see *"Node Secret" on page 65*).

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

If you haven’t 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 above, we recommend to refer to the ACE/Server’s manuals for detailed information on the configuration of the ACE/Server.

Configuring the BRICK (ACE/Agent)

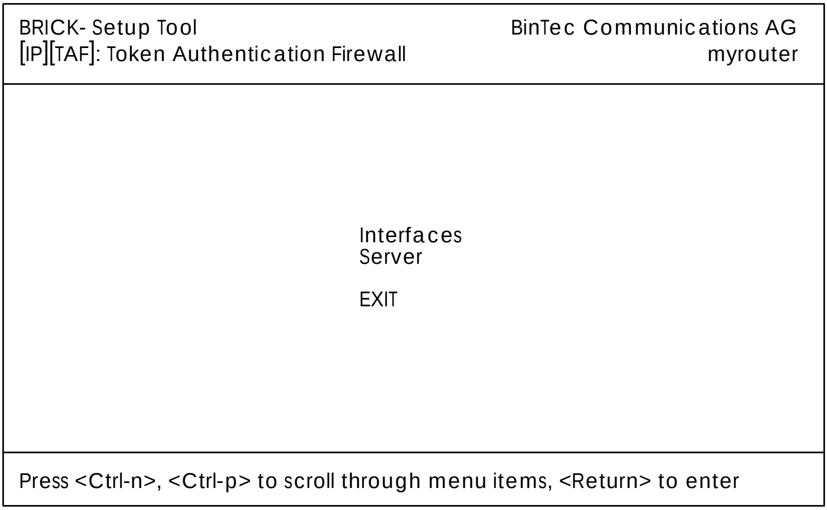
In the following the TAF configuration of the BRICK is described in detail.

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

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                               myrouter

                              Interfaces
                              Server

                              EXIT

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

The INTERFACES menu is used to enable / disable SecurID support separately for each BRICK interface.

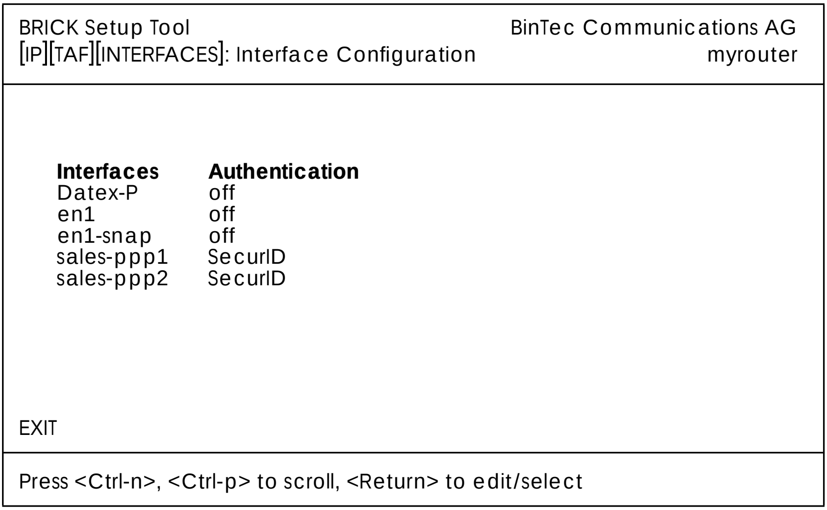
The SERVER menu is used for configuring SecurID Server relevant settings on the BRICK. These settings must correspond to the parameters configured on the ACE / Server.

IP → **TOKEN AUTH. FIREWALL** → **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.

Note: ⚠ 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”.

.



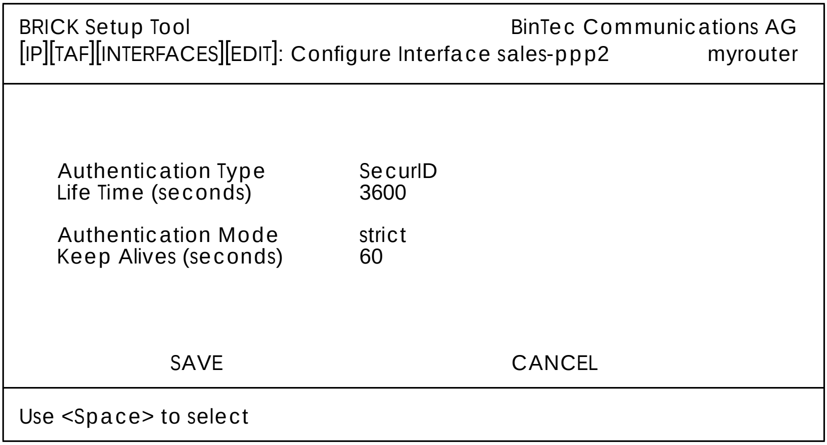
```
BRICK Setup Tool                                  BinTec Communications AG
[IP][TAF][INTERFACES]: Interface Configuration                    myrouter



   Interfaces          Authentication
   Datex-P             off
   en1                 off
   en1-snap            off
   sales-ppp1          SecurID
   sales-ppp2          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 hit the <Enter> key. In the resulting menu ensure that Authentication is set to “SecurID” and select **SAVE** .

IP → TOKEN AUTH. FIREWALL → INTERFACES → EDIT

This menu is used for configuring interface specific settings for Token Authentication Firewall.



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



   Authentication Type          SecurID
   Life Time (seconds)          3600

   Authentication Mode          strict
   Keep Alives (seconds)        60



            SAVE                            CANCEL

Use <Space> to select
```

Authentication Type = This field is used to enable/disable TAF for the respective interface. By default Authentication Type is disabled (**off**). Setting to **SecurID** enables TAF for the interface.

Life Time (seconds) = The time in seconds to allow 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 **strict** each source IP address must be authenticated separately. If set to **loose** all source IP addresses are allowed if at least one IP address was successfully authenticated on this interface.
Default value: **strict**

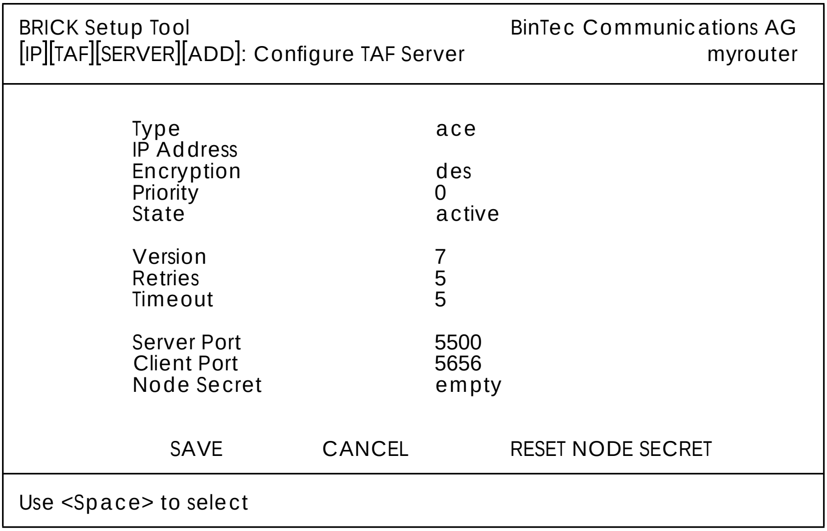
Keep Alives (seconds) = The interval in seconds after which a new keep-alive request is sent to the BRICK by the ACE/Server.
Keep-alive packets will never cause a new connection to be set up, nor will they affect the shorthold mechanism.

IP → TOKEN AUTH. FIREWALL → 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.

NOTE: 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                         myrouter

                 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          RESET NODE SECRET

Use <Space> to select
```

Type = The type of authentication server. Currently ACE/Server (**ace**) 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 des (Data Encryption Standard) or sdi (Security Dy-

namics proprietary) encryption.
Default value is **DES**.

Priority = The authentication server with the lowest priority value is the first used for requests. Use the value 0 for the master server and the value 1 for the slave server.

State = Either active or disabled.

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 **-** 6.
Default value is **5**.

Timeout = The time in seconds to wait for a reply from the authentication server before retrying. Valid range is 1 -20.
Default value is **5**.

Server Port = The port number to use for communication between the BRICK and the authentication server.
By default port **5500** is used.

Client Port = The port number to use for communication with TAF Clients.
Default port is **5656**.

Node Secret = Indicates whether the Node Secret has already been received by the BRICK (**received**) or not (**empty**).
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" check box on the ACE/Server is automatically selected. See *"Configuring the ACE/Server" on page 58*.

You can use RESET NODE SECRET to momentarily clear the Node Secret on the BRICK. When the "Sent Node Secret" check box on the ACE/Server is cleared, the ACE/Server will transmit a new Node Secret at the next communication.
Whenever the BRICK receives a new Node Secret form the ACE/Serv-

er the ***tafServerTable***, where the Node Secret is stored, is saved to the flash ROM.

TAF Commands on the BRICK

makekey Command

`makekey` [**`-g`**]

The makekey command can be used to show the current public key (stored on the *biboAdmPublicKey* variable), or—when invoked with the **`-g`** option—to generate a new pair of keys (public and private).

You will only need to use **`makekey -g`** once before configuring TAF for a WAN partner for the first time.

shtaf Command

`shtaf`

The **shtaf** 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).

If the authentication was successful, it will give you a normal BRICK login prompt. After logging in to the BRICK you can terminate *shtaf* by typing **exit**.

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 *sdinfo* 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 (see p. 64).

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** (see page 66). 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** (see page 66) 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 won't 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 page 63).

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.

biboAdmSyslogMessage (and 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
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 wrong coded.)*	Notice
Tafd: decryption error 0x‹type› *(The message from the client was wrong coded.)*	Err
Tafd: encryption error 0x‹type› *(The message from the client was wrong coded.)*	Err

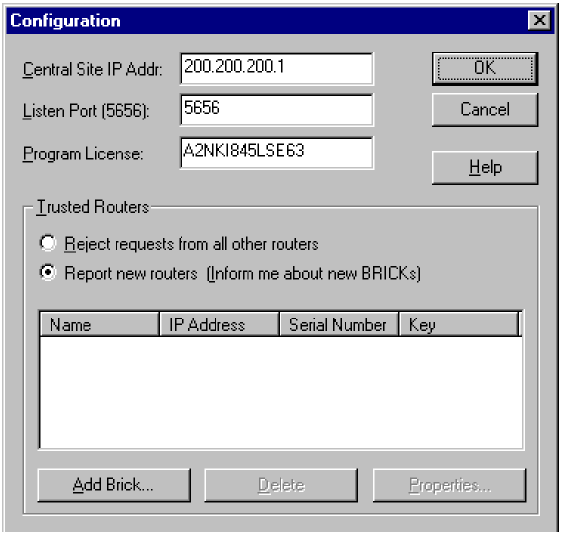
biboAdmSyslogMessage ***(and Meaning)***	***~Level***
Tafd: no key for encryption *(You have to call "makekey -g" to generate a new key.)*	Err
Tafd: Request for token authentication ignored - no key available *(You have to call "makekey -g" to generate a new key.)*	Err
Tafd: TAF server unreachable *(The ACE/Server is unreachable/does not answer/ is not working.)*	Err
Tafd: No TAF License	Err
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

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 respectively can be downloaded from BinTec's Web Server at http://www.bintec.de (Section: FTP Server).You can install it together with BRICKware on the TAF client PC.

When 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 to reinstall all components of BRICKware (including TAF).

The TAF Login program will automatically be installed in your Autostart menu (eventually you must 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.



Central Site IP Addr: 200.200.200.1

Listen Port (5656): 5656

Program License: A2NKI845LSE63

Trusted Routers

○ Reject requests from all other routers

● Report new routers (Inform me about new BRICKs)

Name	IP Address	Serial Number	Key

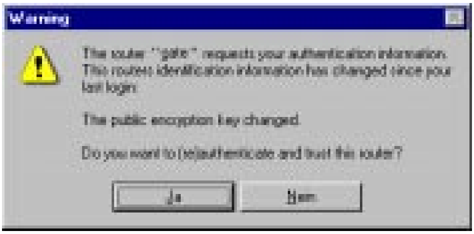
Figure 4: TAF Configuration

Repeat this procedure on each PC you want to use for TAF authentication purposes. Each PC needs his 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.


Warning

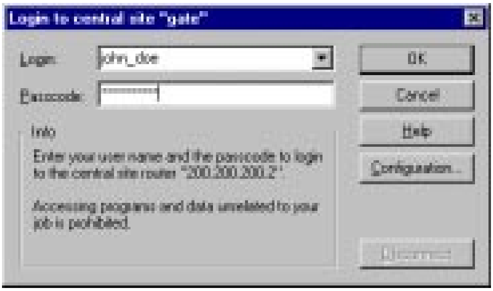
The router ""gate"" requests your authentication information. This routers identification information has changed since your last login.

The public encryption key changed.

Do you want to (re)authenticate and trust this router?

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

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.


Login to central site "gate"

Login: john_doe

Passcode: **********

Info

Enter your user name and the passcode to login to the central site router "200.200.200.2".

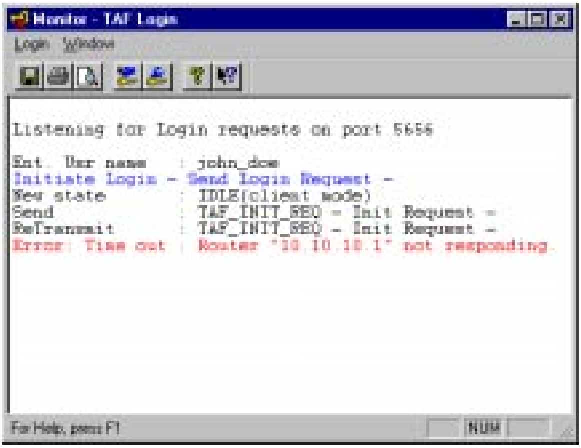
Accessing programs and data unrelated to your job is prohibited.

Figure 6: TAF Login

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

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**.



```
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.
```

Figure 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

What’s covered

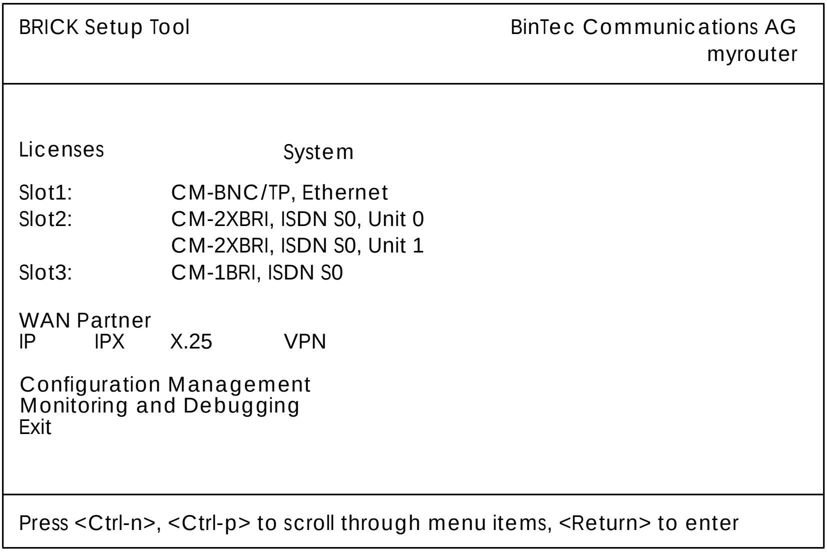
In this chapter we’ll cover the Setup Tool menus and set tings you’ll see while using configure the Virtual private networking support on your router.

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

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

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
                                                                myrouter

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        VPN

Configuration Management
Monitoring and Debugging
Exit

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

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

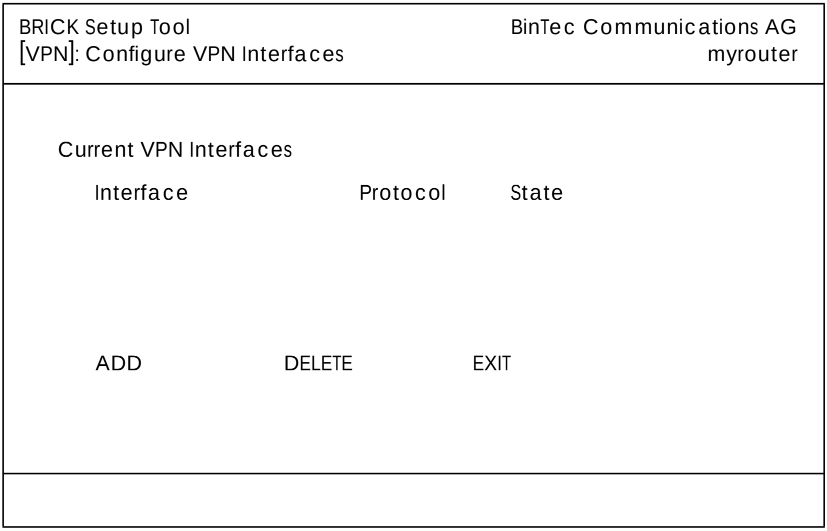
VPN →

The VPN menu lists the current Virtual Private Networking partner interfaces configured on the router.

Select ADD to add a new VPN interface.

Select DELETE to delete a VPN interface that has been marked (using the spacebar) for deletion.

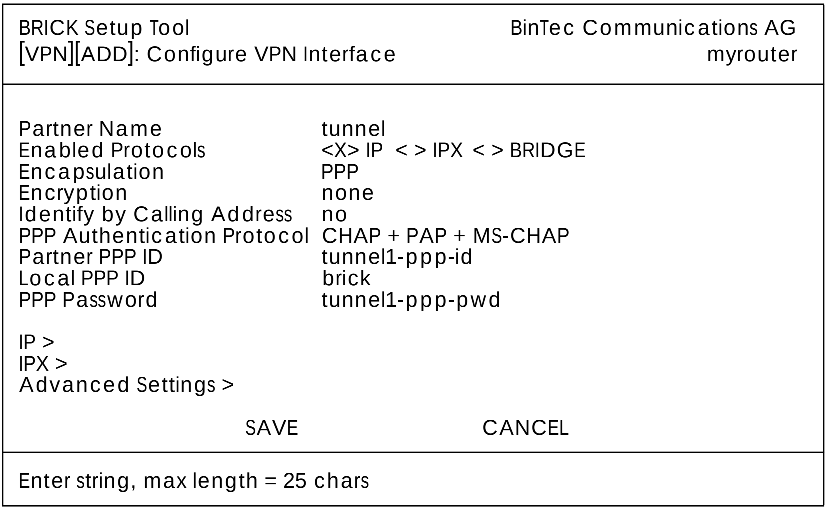
Select EXIT to return to accept the configured list of partners ad return to the main menu.



```
BRICK Setup Tool                                   BinTec Communications AG
[VPN]: Configure VPN Interfaces                                    myrouter


   Current VPN Interfaces

      Interface                    Protocol      State




      ADD              DELETE             EXIT
```

VPN → **ADD** →

Use this menu to create Virtual Private Networking interfaces.



```
BRICK Setup Tool                                BinTec Communications AG
[VPN][ADD]: Configure VPN Interface                             myrouter


Partner Name                   tunnel
Enabled Protocols              <X> IP  < > IPX  < > BRIDGE
Encapsulation                  PPP
Encryption                     none
Identify by Calling Address    no
PPP Authentication Protocol    CHAP + PAP + MS-CHAP
Partner PPP ID                 tunnel1-ppp-id
Local PPP ID                   brick
PPP Password                   tunnel1-ppp-pwd

IP >
IPX >
Advanced Settings >

                    SAVE                   CANCEL

Enter string, max length = 25 chars
```

Partner Name

= The partner name assigned to this virtual interface.

Enabled Protocols

= The protocols that may be routed over this interface.

Encapsulation

= The type of encapsulation to use; currently PPP must be used.

Identify by Calling Address

= This allows the BRICK to verify this VPN partner by its “calling IP Address”. This is the IP address the VPN partner can be reached at on the Internet (i.e., an official IP address).

PPP Authentication Protocol

= 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.

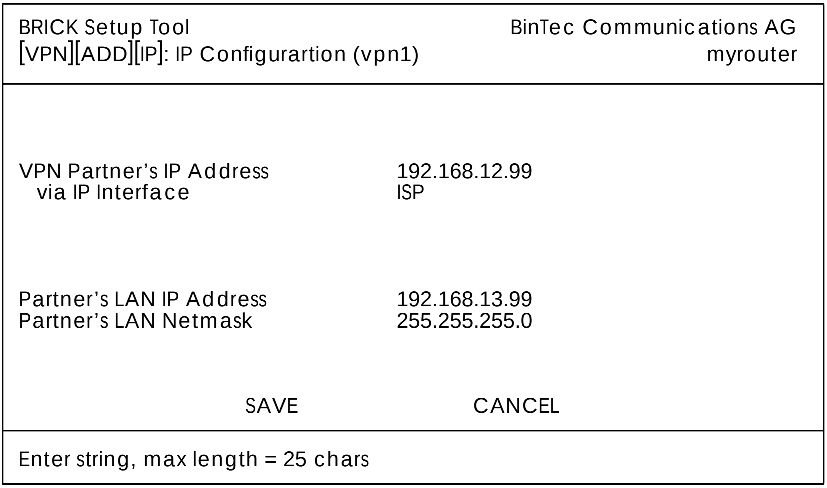
VPN → ADD → IP →

The VPN IP submenu defines IP address settings for the VPN partner interace.

Note: 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 Configurartion (vpn1)                              myrouter



VPN Partner’s IP Address               192.168.12.99
  via IP Interface                     ISP



Partner’s LAN IP Address               192.168.13.99
Partner’s LAN Netmask                  255.255.255.0


                        SAVE                    CANCEL

Enter string, max length = 25 chars
```

VPN Partner’s IP Address

= The VPN partner’s IP address where it can be reached at 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.

Partner’s LAN IP Address

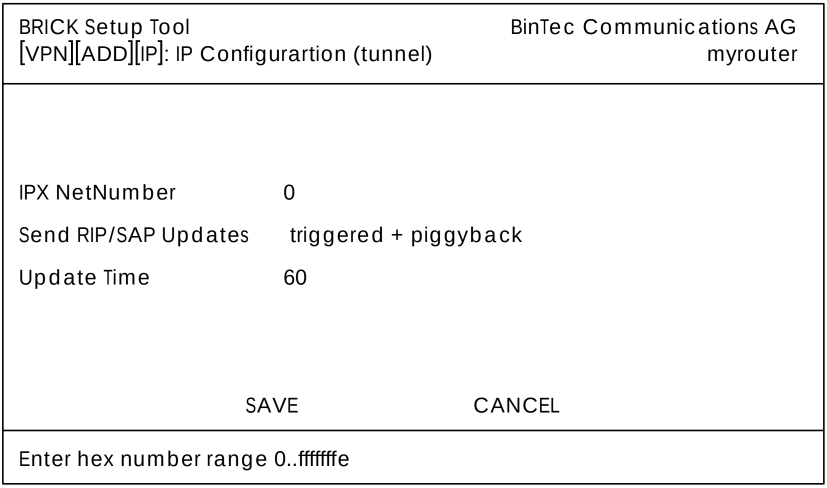
= The VPN partner’s LAN address.

Partner’s LAN Netmask

= The netmask the partner uses on it’s LAN. If left blank, a standard netmask for the respective network class will be used.

VPN → ADD → IPX →

The VPN IPX submenu defines IPX relevant settings for VPN partner interfaces that support IPX.



```
BRICK Setup Tool                                   BinTec Communications AG
[VPN][ADD][IP]: IP Configurartion (tunnel)                         myrouter



IPX NetNumber             0

Send RIP/SAP Updates       triggered + piggyback

Update Time               60



                      SAVE                    CANCEL

Enter hex number range 0..ffffffe
```

IPX NetNumber

= The IPX network number of the network link (the PPTP link). This is required by some IPX routers.

Send RIP/SAP Updates

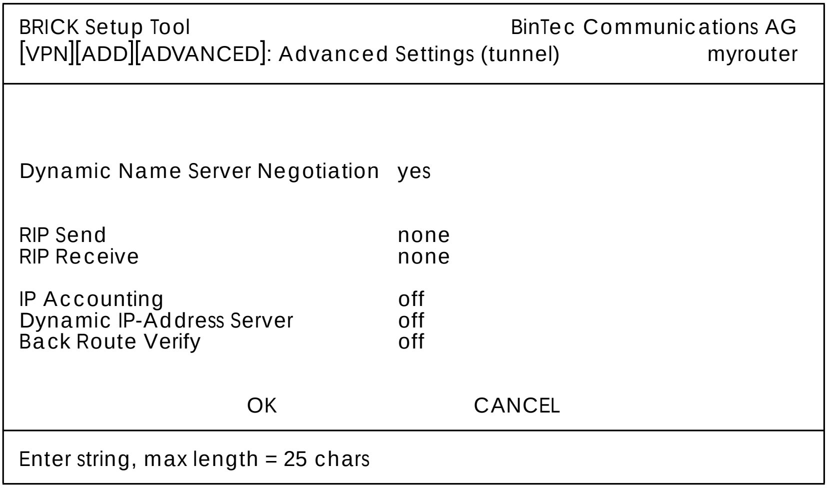
= Determines how often RIP and SAP packets are tranmitted to this VPN partner. The possible options are the same as those defined in the menu, see chapter 4 of the *User's Guide* for additional information.

Update Time

= Determines how often (in seconds) periodic updates are sent to this VPN partner.

VPN → ADD → ADVANCED SETTINGS

The settings defined here are similar to the [WAN PARTNERS][ADVANCED SETTINGS] menu but apply specifically to an VPN partner interface.



```
BRICK Setup Tool                                  BinTec Communications AG
[VPN][ADD][ADVANCED]: Advanced Settings (tunnel)                  myrouter



Dynamic Name Server Negotiation  yes


RIP Send                         none
RIP Receive                      none

IP Accounting                    off
Dynamic IP-Address Server        off
Back Route Verify                off


                OK                    CANCEL

Enter string, max length = 25 chars
```

Dynamic Name Server Negotiation

= Defines whether (and how) the name server's address is configured.

RIP Send/Receive

= Defines the which version of RIP packets to exchange with this partner.

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.

Dynamic IP-Address Server

= Defines whether or not the BRICK should assign this partner an available IP address from the IP address pool.

Back Route Verify

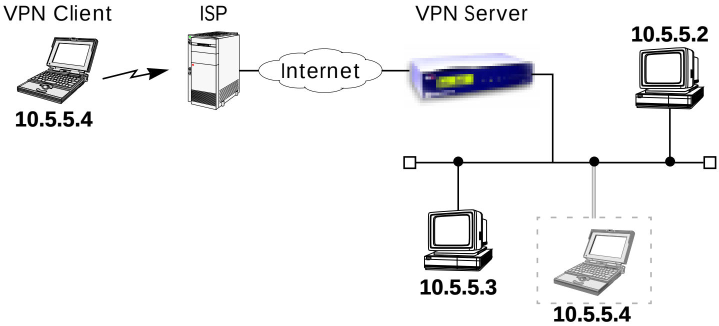
= When enabled the BRICK verifies that the return route for all packets received from this partner interface uses the same interface the packet arrived on.

Overview of Virtual Private Networking

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.



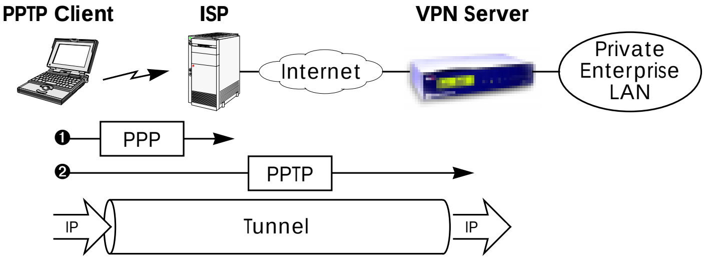
The protocol that makes VPN possible is the Point-to-Point Tunnelling Protocol or PPTP. PPTP is an IETF standard described in RFC 1171.

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 tunnelled 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 an ends.

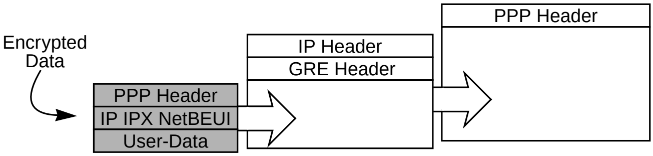
Scenario 1. PPTP Client–to–VPN Server



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.

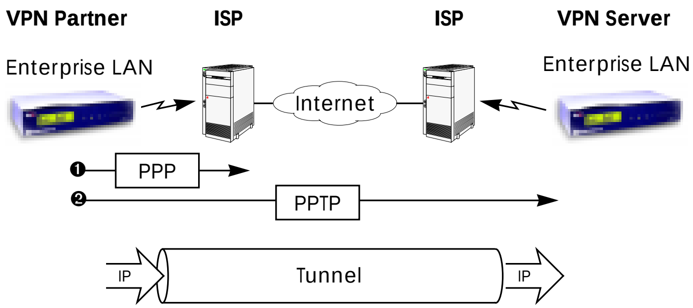
To hosts on the Private Enterprise LAN the remote PPTP Client appears as if it were directly connected to the LAN.

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



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.

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. 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.

Note: ⚠ 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 dialoge 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 (Scenario 2: LAN–to–LAN VPN).

Note: ⚠ The following limitation currently exists when combining compression + encryption for a PPTP link with Windows based hosts.

When the **Enable software compression** option is enabled in the **Server Types** tab (see Step 5.) Windows PPTP Clients offer EITHER MS-STAC Compression OR MPPE 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 the [VPN][ADD] menu on page 90).

Virtual Private Networking Examples

Example Client-to-LAN Configuration

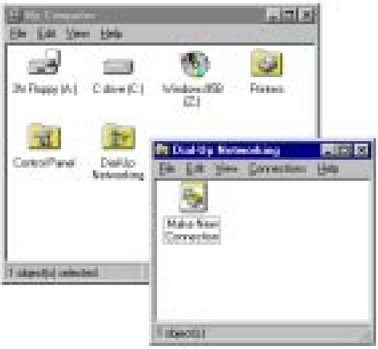
The Virtual Private Network shown in Scenario 1 on page 83 would be configured as follows.

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.

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



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

3.Here you will need to enter the ISP's telephone number.

4.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.

5.Click the **Server Types** tab.

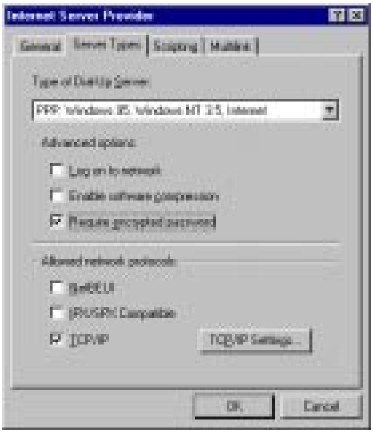
– In the **Type of Dial-Up Server:** field select:
"PPP: Windows 95, Windows NT, Internet"

– In the **Advanced options:** box

Disable	"Log on to network"
Disable	"Enable software compression"
Enable	"Require encrypted password"

– In the **Allowed network protocols:** box

Disable	"NetBEUI"
Disable	"IPX"
Enable	"TCP / IP"



Type of Dial-Up Server:
PPP: Windows 95, Windows NT 3.5, Internet

Advanced options:
☐ Log on to network
☐ Enable software compression
☑ Require encrypted password

Allowed network protocols:
☐ NetBEUI
☐ IPX/SPX Compatible
☑ TCP/IP

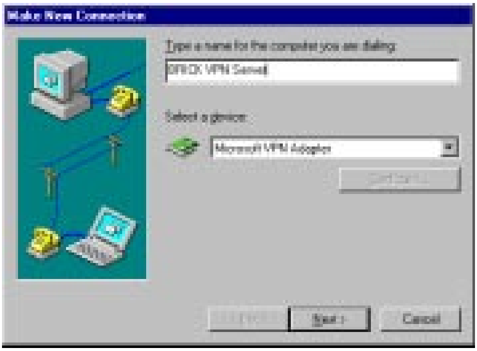
6.Click the **TCP/IP Settings...** button. Verify the IP address, name service, and compression settings are consistent with those required by the ISP and click **OK**.

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

7.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.

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



Type a name for the computer you are dialing:
BRICK VPN Server

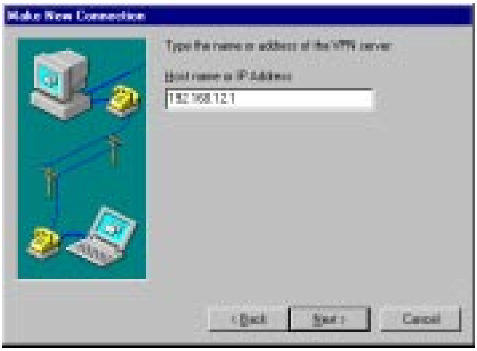
Select a device:
Microsoft VPN Adapter

2.In the **Type a name for the computer you are dialing:** field specify a name for your BRICK VPN Server.

3.From the **Select a device:** drop menu select the device “Microsoft VPN Adapter” and click **Next>**.

In the dialogue shown below enter the official IP address of the BRICK VPN Server.

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



Type the name or address of the VPN server:

Host name or IP Address:
192.168.12.1

4.Click **Next>** and the **Finish**. A new icon for the BRICK VPN Server will be added to the Dial-Up Networking folder.

5.In the Dial-Up Networking folder right-click the new BRICK VPN Server icon and select **Properties** to verify the connection settings.

6.Click the **Server Types** tab.

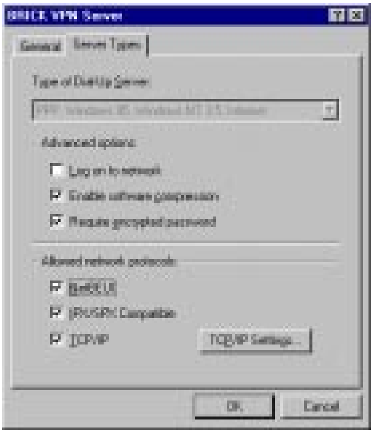
– In the **Type of Dial-Up Server:** field select:
“PPP: Windows 95, Windows NT, Internet”

– 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”

– 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.



BRICK VPN Server

General | Server Types

Type of Dial-Up Server:
PPP: Windows 95, Windows NT 3.5, Internet

Advanced options:
□ Log on to network
☑ Enable software compression
☑ Require encrypted password

Allowed network protocols:
☑ NetBEUI
☑ IPX/SPX Compatible
☑ TCP/IP

7.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 page 90).

8.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 on page 92.

Configure BRICK VPN Server

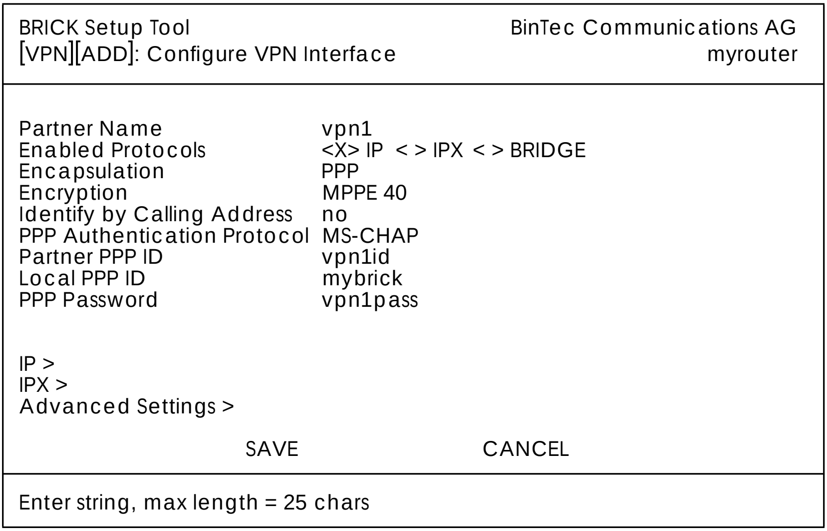
Requirements: A separate VPN license must be installed before the BRICK will support VPN connections. A VPN license can be purchased from BinTec Communications directly or from your local distributor.

Configure Link to the Internet Service Provider.

1.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

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



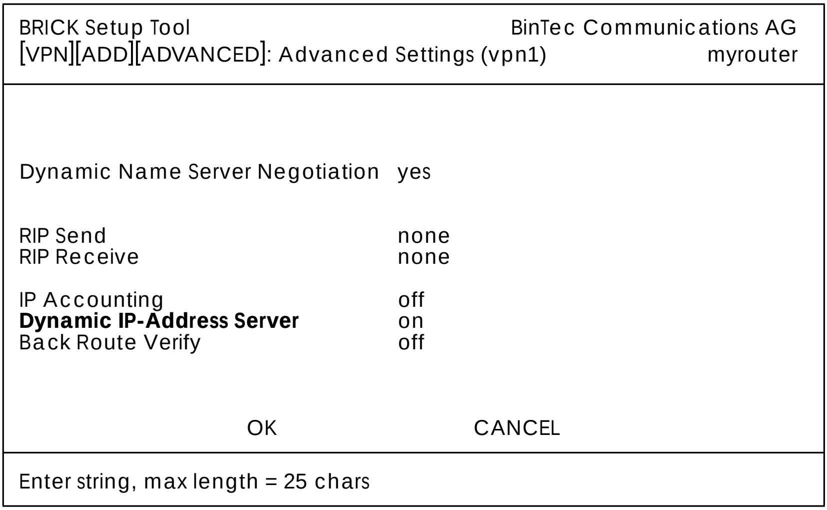
```
BRICK Setup Tool                                    BinTec Communications AG
[VPN][ADD]: Configure VPN Interface                                 myrouter


Partner Name                    vpn1
Enabled Protocols               <X> IP  < > IPX  < > BRIDGE
Encapsulation                   PPP
Encryption                      MPPE 40
Identify by Calling Address     no
PPP Authentication Protocol     MS-CHAP
Partner PPP ID                  vpn1id
Local PPP ID                    mybrick
PPP Password                    vpn1pass


IP >
IPX >
Advanced Settings >

                    SAVE                   CANCEL

Enter string, max length = 25 chars
```

- In the **Encryption** field you may select MPPE encryption (40 bit or 128 bit session-key) or none.
- Disable (“no”) the **Identify by Calling Address** option. This option can not be used since the BRICK will assign the PPTP client an IP address at connect time.
- In the **PPP Authentication Protocol** field select which authentication to use.

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

- The **Partner PPP ID** and **PPP Password** fields define the values the VPN Partner must enter in the **User name** and **Password:** fields when establishing the VPN Connection.

2.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 in the **ADVANCED SETTINGS** sub menu must be enabled.



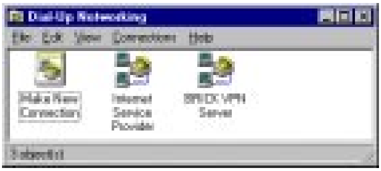
```
BRICK Setup Tool                                      BinTec Communications AG
[VPN][ADD][ADVANCED]: Advanced Settings (vpn1)                        myrouter



Dynamic Name Server Negotiation   yes


RIP Send                          none
RIP Receive                       none

IP Accounting                     off
Dynamic IP-Address Server         on
Back Route Verify                 off



                     OK                    CANCEL

Enter string, max length = 25 chars
```

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

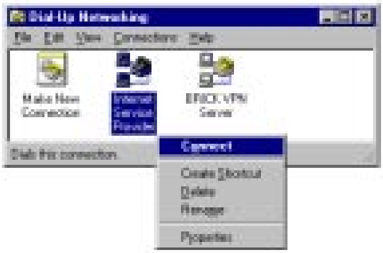
3.So that the BRICK can assign the PPTP client an IP address, make sure there are available IP addresses defined in the **IP** → **Dynamic IP Addresses** menu.

Connecting to the BRICK VPN Server

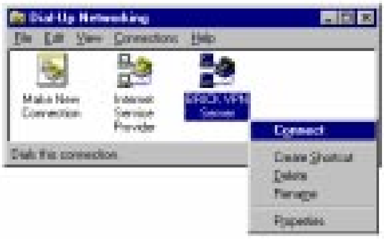
1.Open the Dial-Up Networking folder by double-clicking **My Computer**, and then **Dial-Up Networking**.



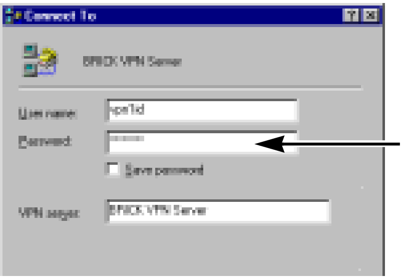
2.Right-click the Internet Server Provider icon, select **Connect** and enter the user / password assigned by the ISP.



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



4. In the **Connect To** window shown below enter the PPP ID and PPP Password settings configured on the BRICK (see page 91) in the **User name** and **Password:** fields.



Connect To

BRICK VPN Server

User name: vpn1id

Password: ******

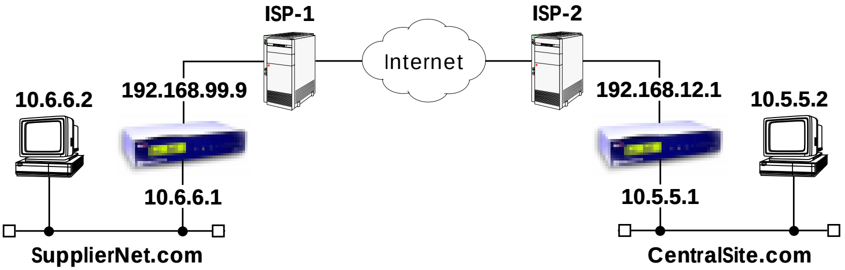
□ Save password

VPN server: BRICK VPN Server

(PPP Password)
vpn1pass

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.



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.

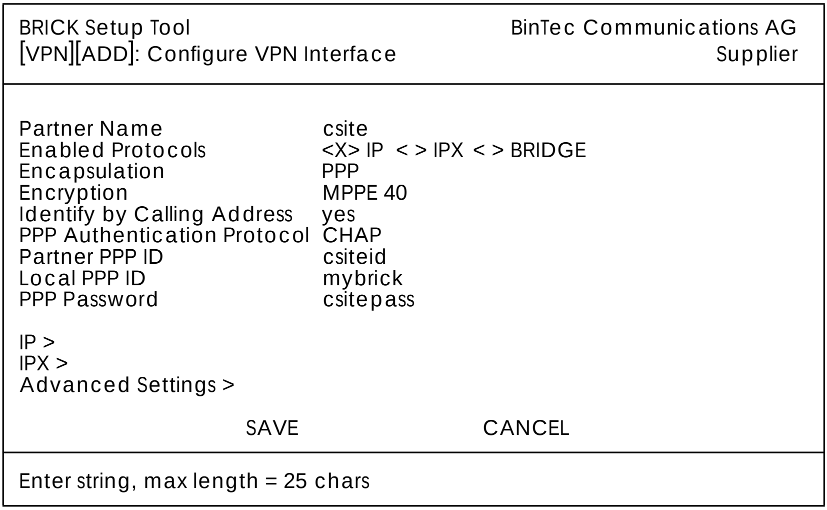
Configuration on SupplierNet BRICK

1. 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".
2. The link to the ISP-1 can be setup as a standard dial-up / leased PPP interface in the WAN PARTNER menu.
3. Configure the VPN Partner interface in the VPN menu. The VPN Partner interface for the BRICK-XL 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 encryption (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 **PPP Authentication Protocol** field select which authentication to use.

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

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

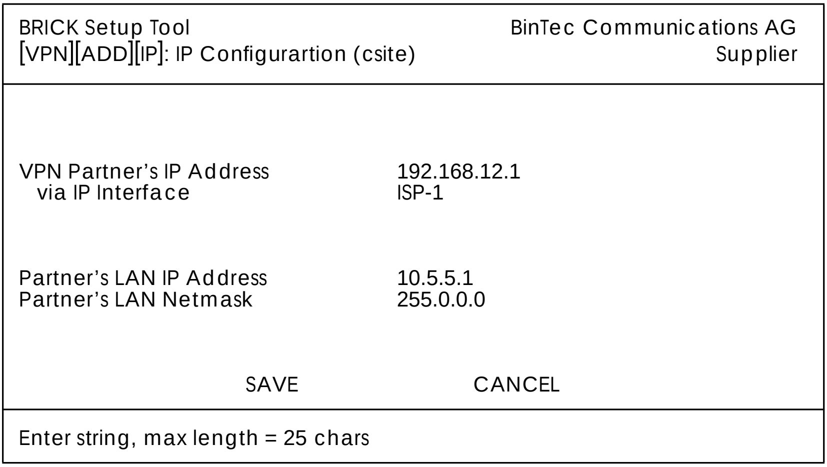


```
BRICK Setup Tool                                    BinTec Communications AG
[VPN][ADD]: Configure VPN Interface                                 Supplier

Partner Name                  csite
Enabled Protocols             <X> IP  < > IPX < > BRIDGE
Encapsulation                 PPP
Encryption                    MPPE 40
Identify by Calling Address   yes
PPP Authentication Protocol   CHAP
Partner PPP ID                csiteid
Local PPP ID                  mybrick
PPP Password                  csitepass

IP >
IPX >
Advanced Settings >

                  SAVE                  CANCEL

Enter string, max length = 25 chars
```

4.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 Configurartion (csite)                          Supplier


VPN Partner’s IP Address             192.168.12.1
  via IP Interface                   ISP-1


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
```

5.In the ADVANCED SETTINGS sub menu the **Dynamic IP Address Server** option must be set to “off”. Other options available there apply to the VPN interface and are described in chapter 4 of your *User’s Guide* under the [WAN PARTNERS][ADVANCED SETTINGS] section.

Configuration on Central Site BRICK

1.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”.

2.The link to the ISP-2 can be setup as a standard dial-up / leased PPP interface in the WAN PARTNER menu.

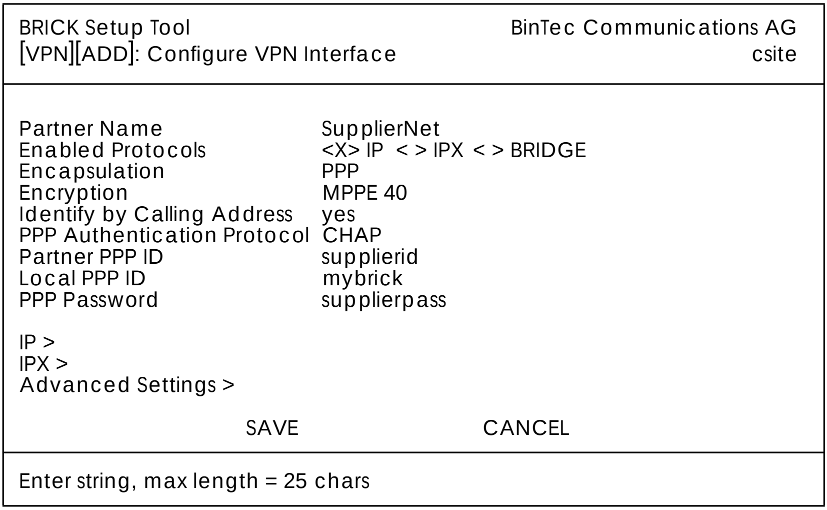
3.Configure the VPN Partner interface in the VPN menu. The VPN Partner interface for the BRICK-XL on SupplierNet.com could be configured as follows.

- Define a partner name (`SupplierNet`) and enable one or more protocols to support on the link.

- In the **Encryption** field you may select MPPE encryption (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 **PPP Authentication Protocol** field select which authentication to use.

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

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

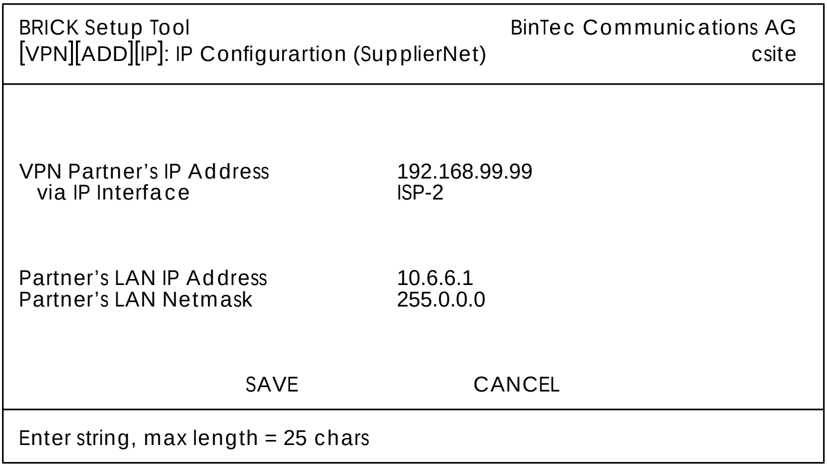


```
BRICK Setup Tool                                    BinTec Communications AG
[VPN][ADD]: Configure VPN Interface                                    csite

Partner Name                    SupplierNet
Enabled Protocols               <X> IP  < > IPX  < > BRIDGE
Encapsulation                   PPP
Encryption                      MPPE 40
Identify by Calling Address     yes
PPP Authentication Protocol     CHAP
Partner PPP ID                  supplierid
Local PPP ID                    mybrick
PPP Password                    supplierpass

IP >
IPX >
Advanced Settings >

                 SAVE                       CANCEL

Enter string, max length = 25 chars
```

4.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 Configurartion (SupplierNet)                      csite


VPN Partner's IP Address          192.168.99.99
  via IP Interface                ISP-2


Partner's LAN IP Address          10.6.6.1
Partner's LAN Netmask             255.0.0.0


                  SAVE                   CANCEL

Enter string, max length = 25 chars
```

5.In the ADVANCED SETTINGS sub menu the **Dynamic IP Address Server** option must be set to "off". Other options available there apply to the VPN interface and are described in chapter 4 of your *User's Guide* under the [WAN PARTNERS][ADVANCED SETTINGS] section.

6

X.25

What’s covered

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

Then we’ll cover all of the menus and settings you’ll 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.

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.

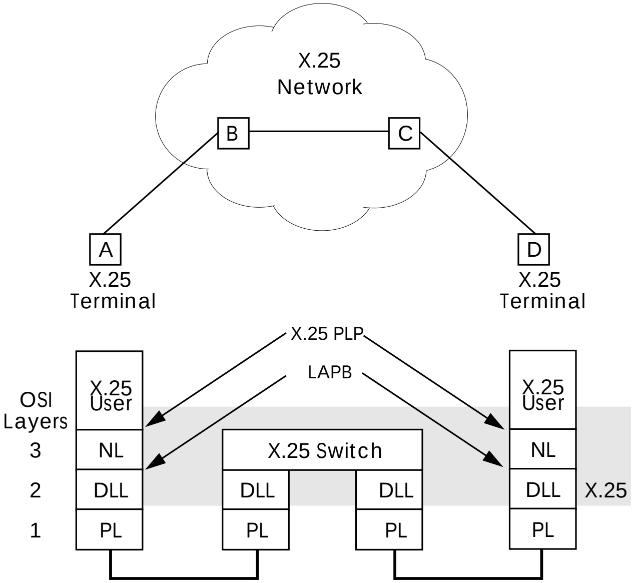
Packet Switched

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 “piggyback” confirmation messages onto other packets. Also, in X.25, packets always arrive in sequence at the receiving station.



Call Setup

Before data can be exchanged among X.25 partners an X.25 call must be setup. 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.

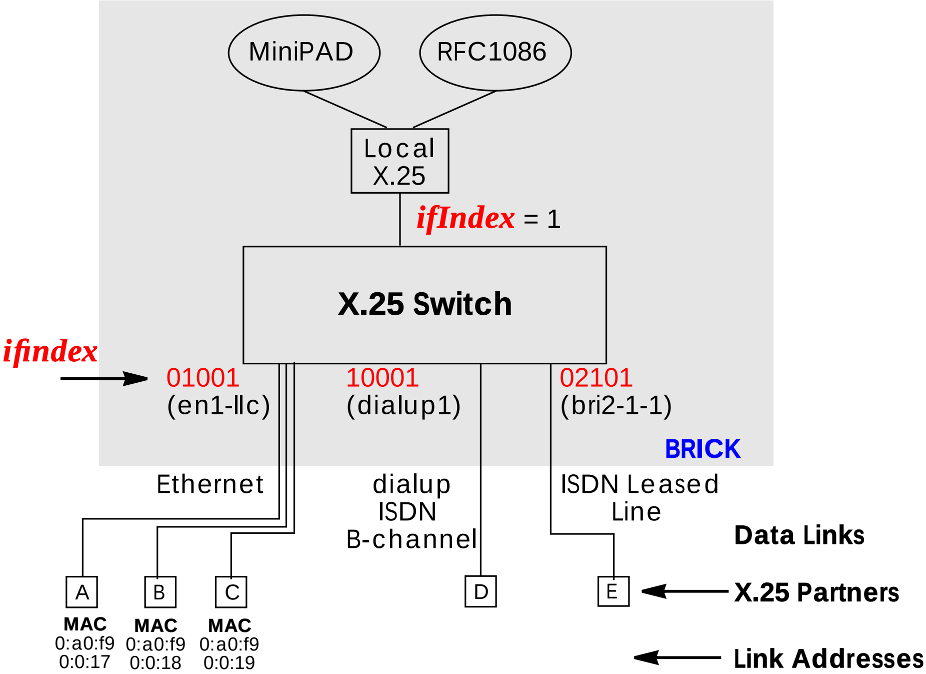
If an X.25 network lies between two end stations, the VC numbers used by each end station may be different. For example, if hosts A and D in the diagram above are communicating, the VC number used for the A–B connection may be different from the one used for C–D.

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.

Data Links and Virtual Circuits

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.

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



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 resp. 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.

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 resp. in the ***biboPPPTable***, 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)

X.25 Addressing Schemes

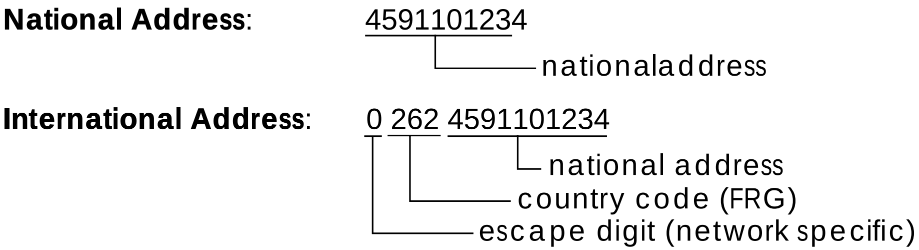
As in TCP/IP networks, each host in an X.25 network must be uniquely identified before communication between them is possible. However, there is one important difference. In TCP/IP, each data packet contains the source/destination addresses and is routed individually (packets can take different paths). 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 Addresses (X.213)

Standard X.25 Addressing (X.121)

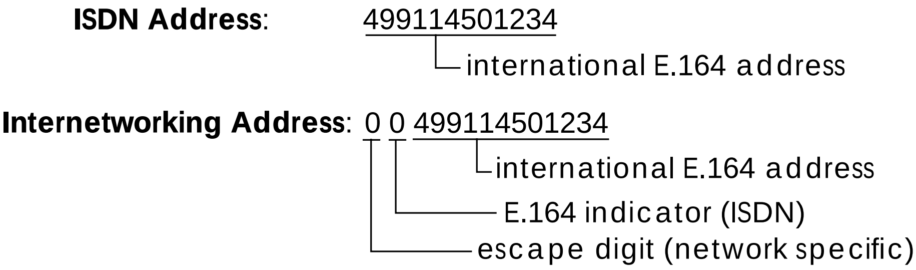
The X.121 addressing scheme is the oldest and most common format used in X.25 networks. X.121 addresses consist of up to 15 digits and may begin with a leading escape digit (normally a 0). If the leading 0 is present, it is assumed to be an international address, otherwise a national address is assumed. For example:[1]



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,

[1] Note that spaces in the example addresses are used only for added readability.

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:


ISDN Address:
499114501234
international E.164 address
Internetworking Address:
0 0 499114501234
international E.164 address
E.164 indicator (ISDN)
escape digit (network specific)

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.[1]

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[2]. The first two digits have special meanings and specify the Type of Address (TOA) and Numbering Plan Identification (NPI) respectively.

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

[1] The BRICK supports extended addresses and differentiates between standard and extended addresses using a leading @ in the ~Addr field.

[2] Most implementations are currently using less than 42 digits.

For example, the following addresses are characterized according to their TOA and NPI digits:[1]

A national X.121 address	@2 3 4591101234
An international X.121 address	@1 3 262 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 conformant (indicated by a leading X) and Non-OSI conformant (indicated by a leading N).

Some example NSAP addresses are as follows:[2]

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.

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.

[1] Spaces in the example addresses are used only for added readability.

[2] Note that spaces in the example addresses are used only for added readability.

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 *Data Links and Virtual Circuits* on page 102) 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	

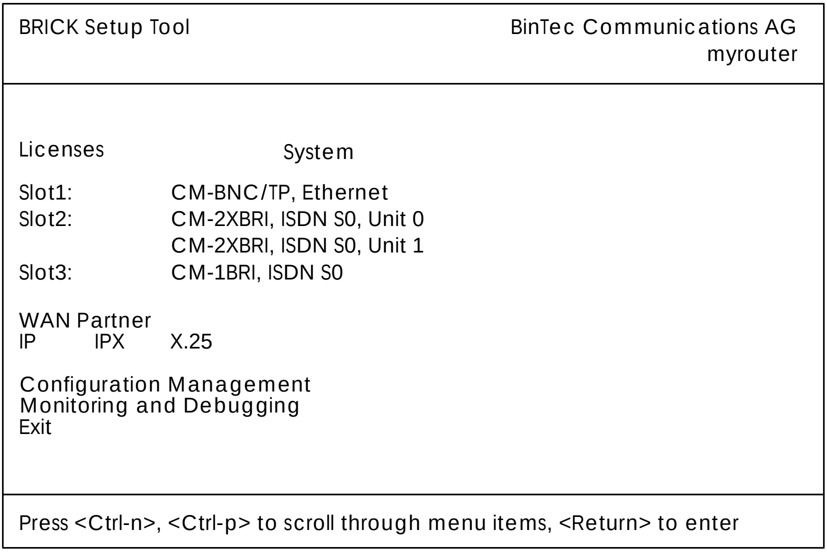
Here, the first two entries route all calls between partners A and D. The third and fourth entries provide routes for all calls between partners B and E. The last two entries specify routes for calls originating from partner C. Any calls to an X.25 destination address beginning with 0, 1, 2, 3, or 4 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.

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.

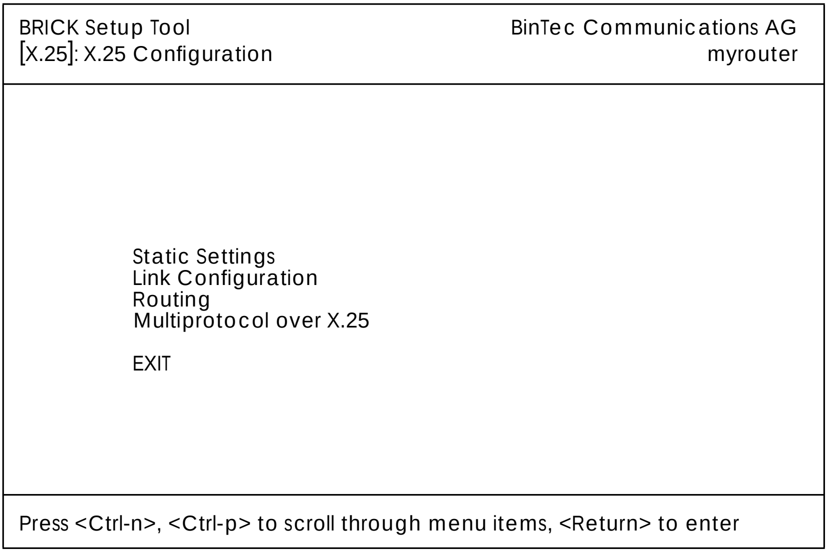


```
BRICK Setup Tool                                 BinTec Communications AG
                                                                 myrouter


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
```

X.25 This is the point where our exploration of Setup Tool begins.

X.25 →

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



```
BRICK Setup Tool                                    BinTec Communications AG
[X.25]: X.25 Configuration                                          myrouter



                  Static Settings
                  Link Configuration
                  Routing
                  Multiprotocol over X.25

                  EXIT


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

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.

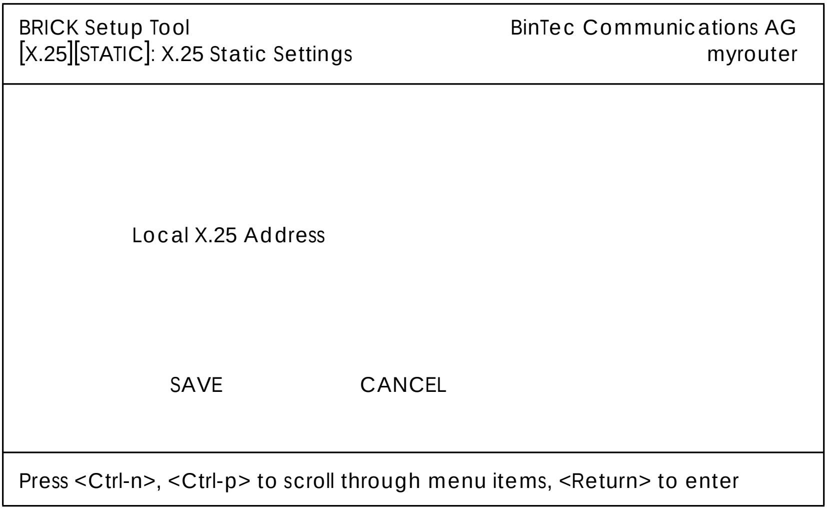
ROUTING contains the router's X.25 routing table.

MULTIPROTOCOL OVER X.25 is used to configure the Multiprotocol Routing over X.25 (MPX25) feature.

Select EXIT to return to the main menu.

X.25 → 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                               myrouter



         Local X.25 Address



            SAVE                  CANCEL


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

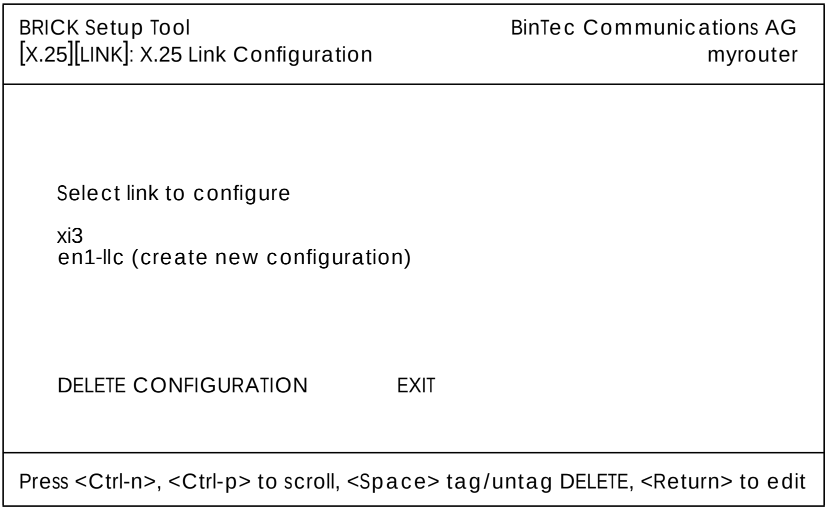
Local X.25 Address = The router's official X.25 address. Setting this variable is only required if the router is not directly connected to an official X.25 data network. When connected directly, the router ascertains its X.25 address automatically.

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.

X.25 → 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 - 3 on a BRICK-XM, 1 - 6 on a BRICK-XL), the system creates an initial link using xi1 through xi3 (xi6).
• X.31 interfaces	If you're 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                                 myrouter


     Select link to configure

     xi3
     en1-llc (create new configuration)


     DELETE CONFIGURATION           EXIT

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

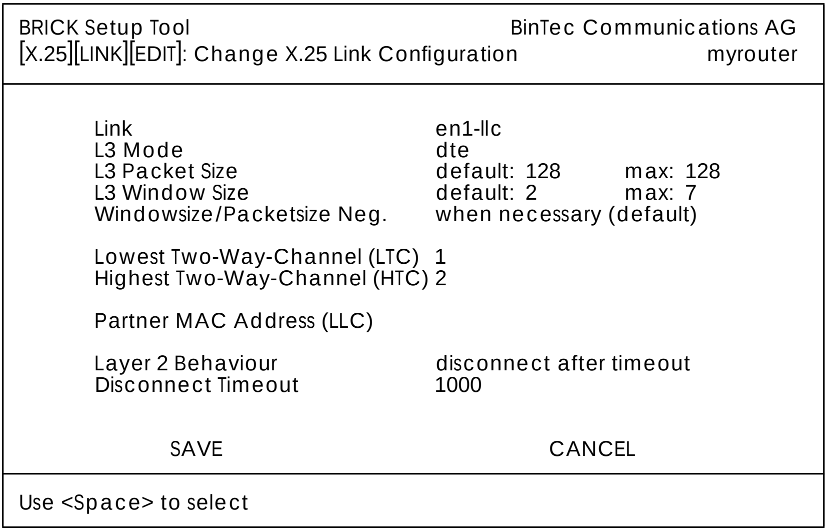
Before an X.25-compatible interface can be used, its link characteristics must first be set.

To edit an X.25 link highlight the entry and then enter <Return>.

To remove an X.25 link, tag the entry for deletion (spacebar) and select DELETE CONFIGURATION .

X.25 → **LINK CONFIGURATION** → **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                  myrouter


        Link                            en1-llc
        L3 Mode                         dte
        L3 Packet Size                  default:  128        max:  128
        L3 Window Size                  default:  2          max:  7
        Windowsize/Packetsize Neg.      when necessary (default)

        Lowest Two-Way-Channel (LTC)    1
        Highest Two-Way-Channel (HTC)   2

        Partner MAC Address (LLC)

        Layer 2 Behaviour               disconnect after timeout
        Disconnect Timeout              1000


                SAVE                                CANCEL

Use <Space> to select
```

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 *default* and *maximum* values for Packet size (128, …, 4096 bytes) and Window size (2 - 127).

Windowsize/Packetsize Neg. = Decides whether window / packet-size negotiation is made for this X.25 link. The possible values are ***never***, ***always*** and ***when necessary***, where ***when necessary*** is the default value. The value ***never*** means no negotiation. When a call arrives that does not correspond to the default size, the call is cleared. ***Always*** means negotiations are always made and when ***when necessary*** 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.

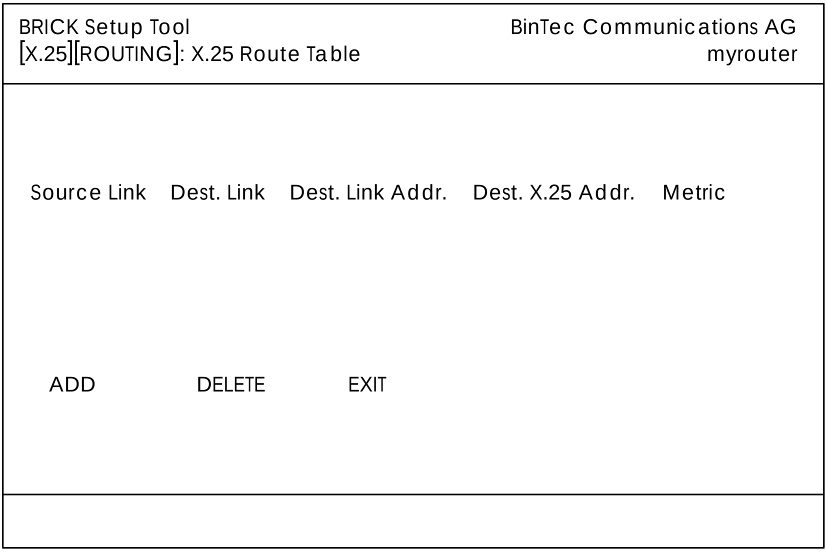
Partner MAC Address (LLC) = Used when configuring a link for a partner on the LAN and specifies the host's MAC or hardware address.

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.

X.25 → 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.



```
BRICK Setup Tool                                  BinTec Communications AG
[X.25][ROUTING]: X.25 Route Table                                 myrouter



 Source Link   Dest. Link   Dest. Link Addr.   Dest. X.25 Addr.   Metric





   ADD              DELETE           EXIT
```

To edit an X.25 route, highlight the entry and then enter <Return>.

Select ADD to create a new X.25 route.

Select DELETE to remove an X.25 route entry that has been tagged (using the spacebar) for deletion.

Select EXIT to accept the list of X.25 routes and return to the previous menu.

X.25 → **ROUTING** → **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.

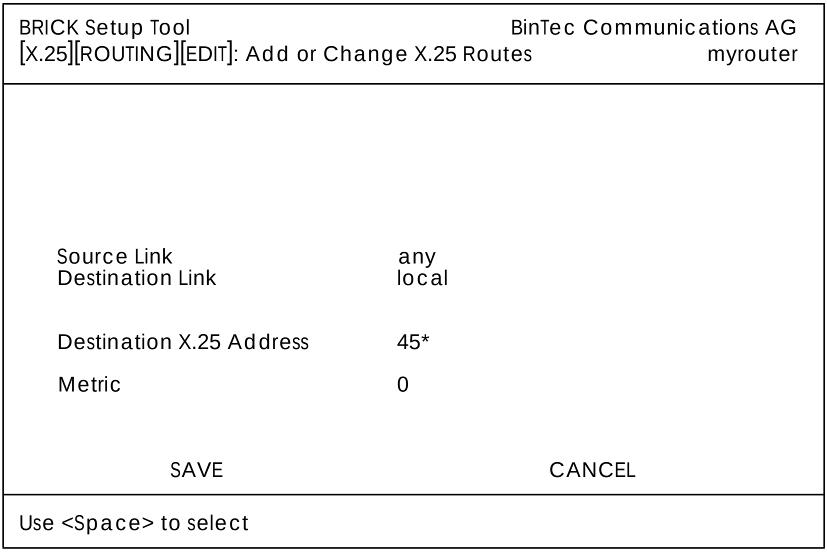
You must define the destination link where the X.25 packets will be routed by specifying these two parameters. Standard wildcard characters can also be used in the Destination Address parameter.

{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

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 Setup Tool                                   BinTec Communications AG
[X.25][ROUTING][EDIT]: Add or Change X.25 Routes                   myrouter




    Source Link                        any
    Destination Link                   local


    Destination X.25 Address           45*

    Metric                             0


            SAVE                                CANCEL

Use <Space> to select
```

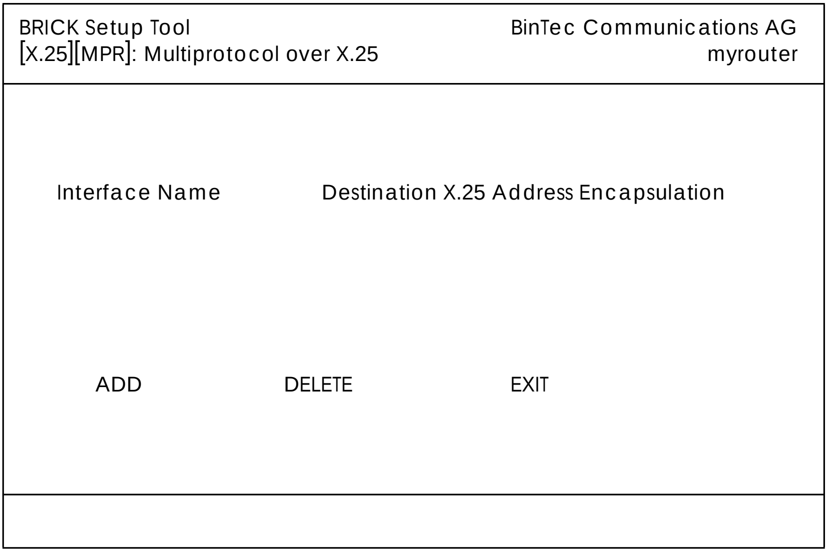
SAVE immediately saves route to memory and returns to the previous menu.

CANCEL discards entries made here and returns to previous menu.

X.25 → **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.

Note: ⚠ 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                               myrouter


   Interface Name         Destination X.25 Address Encapsulation



      ADD              DELETE              EXIT
```

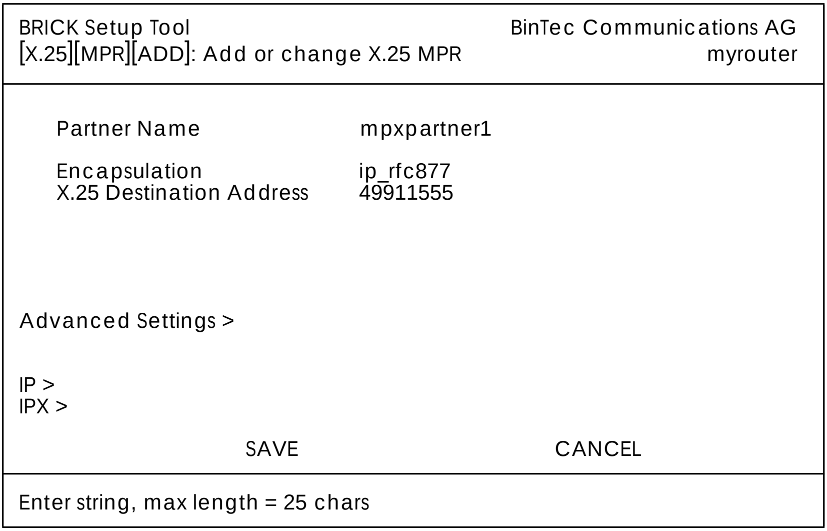
Select **ADD** to create a new MPX25 link.

Select **DELETE** to remove an MPX25 link tagged for deletion.

Select **EXIT** to accept the list of MPX25 links and return to the previous menu.

X.25 → **MULTIPROTOCOL OVER X.25** → **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                            myrouter

    Partner Name                    mpxpartner1

    Encapsulation                   ip_rfc877
    X.25 Destination Address        49911555



Advanced Settings >


IP >
IPX >
                  SAVE                          CANCEL

Enter string, max length = 25 chars
```

Partner Name = Enter a unique name to identify this MPX25 partner.

Encapsulation = Here you select the type of encapsulation / protocol to use. Note that the remote MPX25 partner must be configured to use the same encapsulation.

Encapsulation	Protocol		
ip_rfc877	IP		
ip			
mpr		IPX	Bridge
ipx			

When selecting ***ip_rfc877*** or ***ip***, you must define the IP settings in the IP Submenu (see below).

When selecting ***mpr***, 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 ***mpr*** is selected and needs no configuration.

When selecting ***ipx***, you must define the IPX settings in the IP menu (see below).

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. {00}4991155 matches both 004991155 and 4991155 for incoming calls, outgoing calls are placed using 004991155.

X.25 → **MULTIPROTOCOL OVER X.25** → **ADD** → **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.

Note: 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.

X.25 → **MULTIPROTOCOL OVER X.25** → **ADD** → **IPX** →

This is where you configure the IPX settings for the remote MPX25 partner. This menu is only available if IPX or ***mpr*** has been enabled.

Note: The settings used in this menu are the same as those used in the **WAN PARTNER** → **ADD** → **IPX** → menu but only apply to this MPX25 partner.

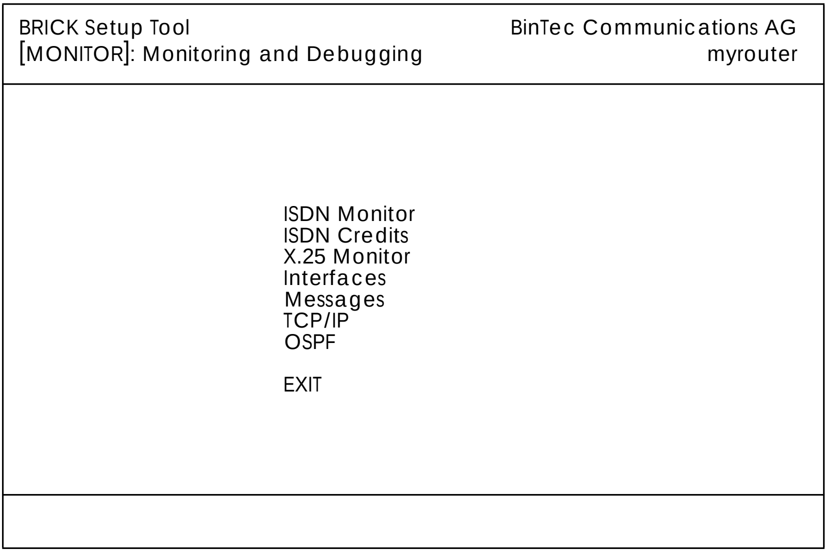
X.25 → **MULTIPROTOCOL OVER X.25** → **ADD** → **ADVANCED SETTINGS** →

This menu can be used to configure advanced features.

Note: 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 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                                   BinTec Communications AG
[MONITOR]: Monitoring and Debugging                                myrouter

                          ISDN Monitor
                          ISDN Credits
                          X.25 Monitor
                          Interfaces
                          Messages
                          TCP/IP
                          OSPF

                          EXIT
```

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.

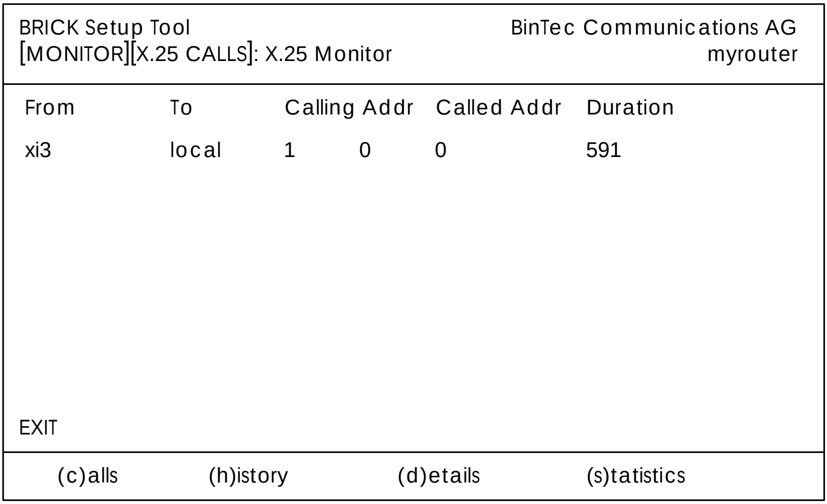
OSPF menu lets you monitor OSPF related information.

Select EXIT to return to the main menu.

MONITORING AND DEBUGGING → X.25 MONITOR

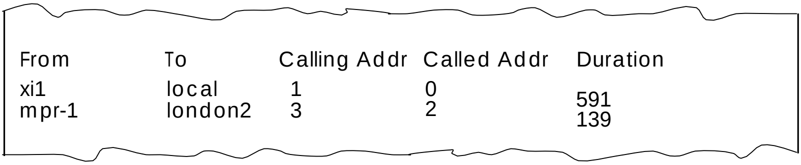
The X.25 Monitor menu initially display all active X.25 connections. These calls include leased and dialup connections made through X.25 public networks or over ISDN.

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.



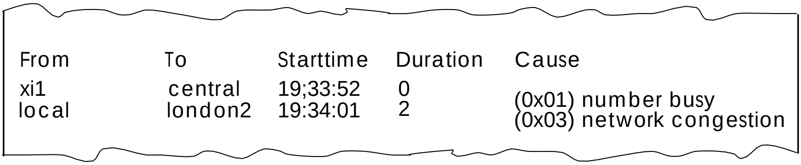
```
BRICK Setup Tool                                  BinTec Communications AG
[MONITOR][X.25 CALLS]: X.25 Monitor                               myrouter

 From          To         Calling Addr  Called Addr  Duration

 xi3           local      1       0     0            591










EXIT

    (c)alls        (h)istory         (d)etails          (s)tatistics
```

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



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

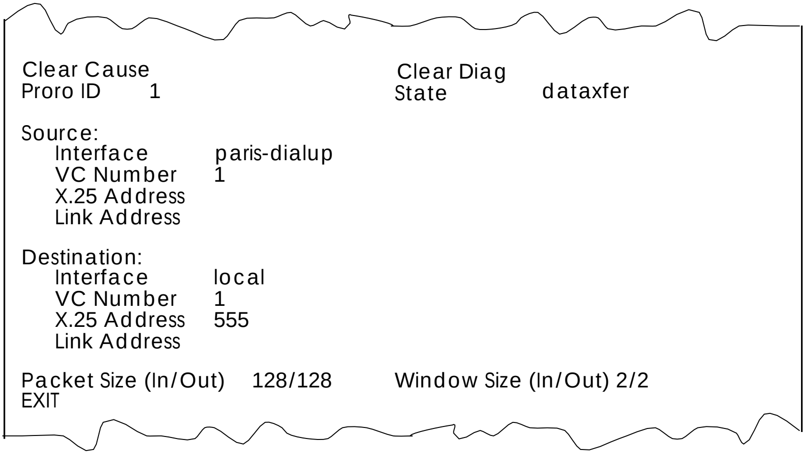
The **(h)istory** listing shows a list of completed X.25 connections (both incoming and outgoing) since the last system reboot.



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

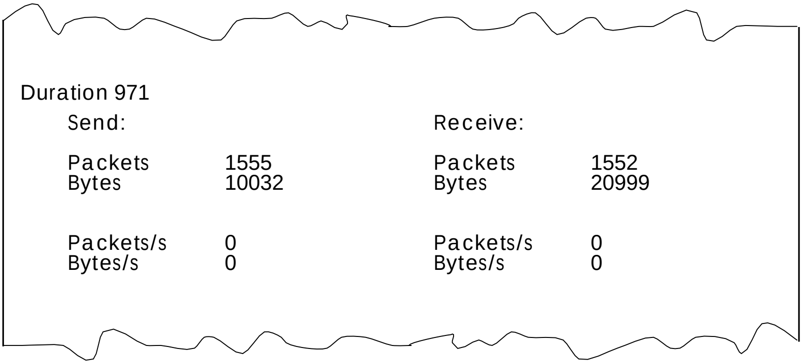
For completed calls, you can display additional information about the call. Select a call from the list, then enter “d” to see a detailed listing.

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
```

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
```

X.25 Features

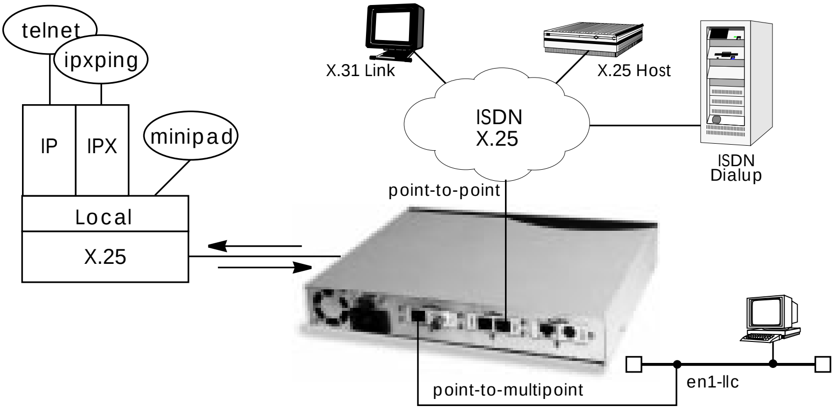
The following pages describe configuring some of the most common X.25 features on the router such as:

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 instance. For example, when routing IP traffic over X.25 links (see Multiprotocol routing configuration on page 137).



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.

Configure it

LICENSES →

Verify License

Verify your X.25 license is valid. You should find "X25 (valid)".

X.25 → LINK CONFIGURATION →

Configure the X.31 Link

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 highlight the link and press <Return> to define the characteristics of this data link.

L3 Mode	dte
L3 Packet Size	default:128 max:128
L3 Window Size	default:2 max:7
Windowsize / Packetsize Neg.	when necessary (default)
Lowest Two-Way-Channel	1
Highest Two-Way-Channel	2
Layer 2 Behaviour	always active

X.25 → ROUTING →

Create Route for Incoming Calls

Next, 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[1] interface. The result: PAD calls are given to the PAD subsystem, calls containing IP data go to the IP subsystem, etc.

Source Link	x31d<*slot*>-<*unit*>-<*TEI*>
Destination Link	local
Destination X.25 Address	<*router's ISDN telno*>

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

X.25 → ROUTING →

Create Route for Outgoing Calls

Create an X.25 route for outgoing calls. This route says that all calls from the local[1] interface are routed to the X.31 link.

Source Link	local
Destination Link	x31d<*slot*>-<*unit*>-<*TEI*>
Destination X.25 Address	<*leave empty*>

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 it's 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.

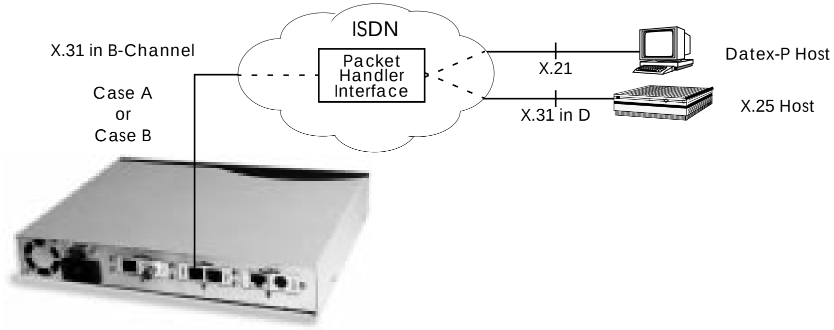
You can also connect to the Deutsche Telekom's Traffic Generator service to verify data transfers are possible over the X.31 link. This can be done with:

```
minipad 026245911029003
```

[1] See page 123 for information on the router's special local interface.

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 S_0 interface. In both scenarios the router accesses X.25 hosts through the Packet Handler Interface (PHI) provided by the ISDN carrier.



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 it

WAN PARTNER → ADD →

Configure WAN Partner

First, configure the PHI as a new WAN partner.

Partner Name	phi
Encapsulation	X31 B-channel

Under WAN NUMBERS → set your PHI's ISDN number if your carrier supports Case A. For Case B you don't need to configure the number.

Number	*<PHI's telephone number>*
Direction	both

X.25 → LINK CONFIGURATION →

Configure the Link

Next, 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't be established, verify with you carrier.

L3 Mode	dte
L3 Packet Size	default:128 max:128
L3 Window Size	default:2 max:7
Windowsize / Packetsize Neg.	when necessary (default)
Lowest Two-Way-Channel	1
Highest Two-Way-Channel	2
Layer 2 Behaviour	disconnect when idle

X.25 → ROUTING → ADD →

Route for Incoming Calls

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[1] interface.

Source Link	*<interface name for PHI>*
Destination Link	local
Destination X.25 Address	*<router's ISDN telephone number>*

X.25 → ROUTING → ADD →

Route for Outgoing Calls

Create another route for outgoing calls. This route says that all calls from the local[1] interface are routed to the PHI.

Source Link	local
Destination Link	*<interface name for your PHI>*
Destination X.25 Address	*<leave empty>*

[1] See page 123 for information on the router's special local interface.

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

You can use the CM-X21 communications module to connect networks over a public (or private) X.25 data network.

Before you begin

Before you start you're going to need the following information.

- The number of Virtual Channels, and the Window and Packet sizes assigned by your X.25 network service provider.
- Your router's official X.25 address.
- The remote partner's official X.25 address.
- Decide what types of traffic will be routed over this interface.

Configure it

CM-X21, X.21 →

Configure Hardware Interface

First, we need to configure the hardware interface.

Layer 1 Mode	dte
Layer 2 State	auto

WAN PARTNER → ADD →

Edit WAN Partner

Locate the appropriate X.21 entry to configure, (X.21 partner entries have the format: xi<*slot number*>) and enable X.25.

Encapsulation	X. 25

X.25 → LINK CONFIGURATION →

Configure Data Link

Locate the X.21 entry for the WAN partner you just configured. If you didn't change the partner name you should see an xi<*slot number*> link depending on where your X.21 module is installed.

L3 Mode	dte
L3 Packet Size	<*Packet assigned by network*>
L3 Window Size	<*Win Size assigned by network*>
Windowsize/Packetsize Neg.	when necessary (default)
Lowest Two-Way-Channel	<*LTC assigned by network*>
Highest Two-Way-Channel	<*HTC assigned by network*>
Layer 2 Behaviour	always active

X.25 → ROUTING → ADD →

Route for Incoming Calls

Next, create a route for incoming calls. This will allow calls arriving on the X.21 link that are addressed to the router's X.25 address to be given to the local[1] interface.

Source Link	xi<*slot number*>
Destination Link	local
Destination X.25 Address	<*router's X.25 address*>

X.25 → ROUTING → ADD →

Route for Outgoing Calls

Create another route for outgoing calls. This route says that all calls from the local[1] interface are routed over the X.21 link.

Source Link	local
Destination Link	xi<*slotnumber*>
Destination X.25 Address	<*leave empty*>

More Info

Depending on how you've set up X.25 routing, you can test your X.25 configurations using minipad. See minipad on page 170. 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 channels available, you can also place a call to your own router's X.25 address with

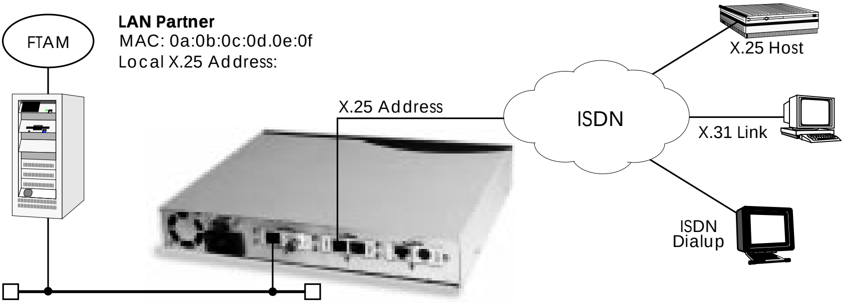
minipad <*your router's X.25 address*>

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 .

[1] See page 123 for information on the router's special local interface.

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.



Before you begin

Before you start you're going to 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 it

X.25 → STATIC SETTINGS →

Configure X.25 Local Address

First, verify the router's local X.25 address is configured.

X.25 Local Address	*<router's X.25 Address>*

X.25 → LINK CONFIGURATION →

Create LAN Host Link

We need to 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)

Highlight the entry and enter <Return> to configure the link. For ethernet links the following settings should be acceptable.

L3 Mode	dce
L3 Packet Size	1024 bytes
L3 Window Size	5
Windowsize / Packetsize Neg.	when necessary (default)
Lowest Two-Way-Channel	1
Highest Two-Way-Channel	4095
Partner MAC address (LLC)	<*LAN Partner's MAC address*>
Layer 2 Behaviour	disconnect when idle

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.

X.25 → ROUTING → ADD →

Edit X.25 Routing Table

Here we 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[1] interface.

Source Link	en1<*slot*>-llc
Destination Link	local
Destination X.25 Address	<*router's X.25 address*>

X.25 → ROUTING → ADD →

Edit X.25 Routing Table

Now we'll 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.

Source Link	local
Destination Link	en<*slot*>-llc
Destination Link Address	<*LAN Partner's MAC address*>
Destination X.25 Address	<*LAN Partner's X.25 address*>

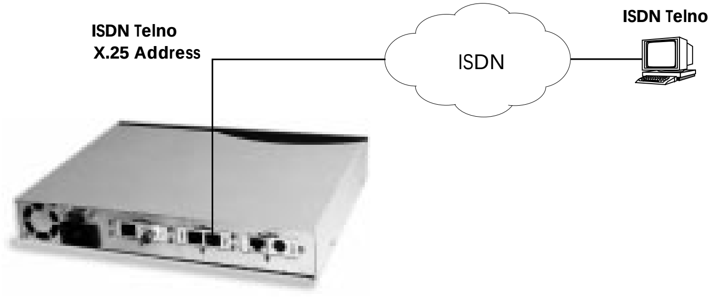
More Info

Depending on how you've set up X.25 routing, you can test your X.25 configurations using minipad. See minipad on page 170.

[1] See page 123 for information on the router's special local interface.

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.



Before you begin

Before you start you're going to need the following information.

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

Configure it

X.25 → STATIC SETTINGS → **Configure X.25 Local Address**

Verify the router's X.25 address is set here.

WAN PARTNER → ADD → **Edit WAN Partner**

Create a new WAN partner interface and enable X.25 traffic.

Encapsulation	X. 25

Under WAN NUMBERS → set the partner's ISDN number.

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

Note: ⚠ 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.

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.

Before you begin

Before you start you're going to need the following information.

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

Configure it

X.25 → STATIC SETTINGS → **Configure X.25 Local Address**

Verify the router's X.25 address is set here. (optional)

WAN PARTNER → ADD → **Edit WAN Partner**

Create a new WAN partner interface and enable X.25 without configuration.

Encapsulation	X. 25 No Configuration, No Signalling

The now 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 like described above 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                           -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.

Note: **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 generating 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 above.

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

- 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 *dstLinkAddr*	Meaning
X	X.25 address
N	NSAP

- Rule

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

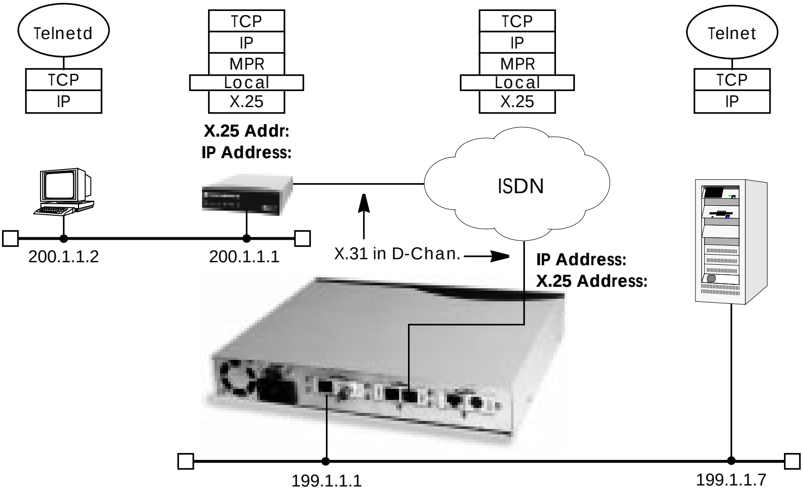
Part of *dstLinkAddr*	Meaning
.	take over one digit
%	delete one digit
*	take over the remaining digits
0-9	insert digits

Examples:

Rule	X.25 Address/NSAP	ISDN Number/MAC Address/IP 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

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'll 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 on page 124.)



Before you begin

Before you start you're going to need the following information.

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

Configure it

X.25 → **MULTIPROTOCOL OVER X.25** → **ADD** → **New MPX25 Partner**

Create a new MPX25 interface for the remote X.25 partner. Here's where we define the types of traffic (IP, IPX, and Bridge) to transport over this link. For our example above, we're only routing IP.

Encapsulation	*<one of: ip_rfc877 | ip | mpr>*
X.25 Destination Address	*<MPX25 partner's X.25 address>*

Note: 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).

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

So under **IP** → set.

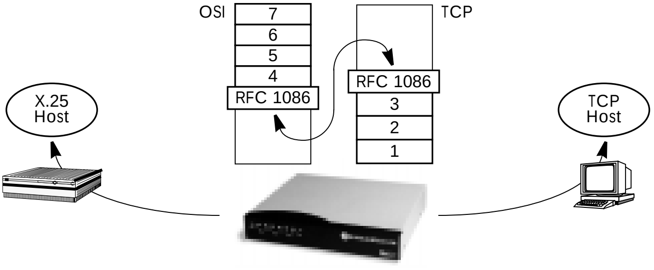
IP Transit Network	yes
Local ISDN IP Address	*<router's IP address>*
Partner's ISDN IP Address	*<MPX25 partner's IP address>*
Partner's LAN IP Address	*<optional>*
Partner's LAN Netmask	*<optional>*

More Info

Depending on how you've set up X.25 routing, you can test your X.25 configurations using minipad. See minipad on page 170.

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.



Depending on which side initiates the connection (see the examples under *More Info* shown on page 140) the router performs the appropriate protocol mappings as shown above.

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.

Configure it

LICENSES → **Verify License**

Verify your X.25 license. You should see “X.25(ok)” in this menu.

X.25 → ROUTING → ADD → **Route for outgoing calls**

X.25 routing must be configured so that incoming and outgoing calls can be established. Using the special *local* interface (see page 123) a minimal X.25 routing setup could be used as follows.

Source Link	local
Destination Link	<*X.25 interface name*[1]>

[1] Use an available X.25 compatible interface name here. By default interfaces for ISDN: x31d-<*slot #*>-<*unit #*>-<*TEI*> and X.21 modules: xi<*slot #*> are available.

X.25 → ROUTING → ADD →

Route for incoming calls

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.

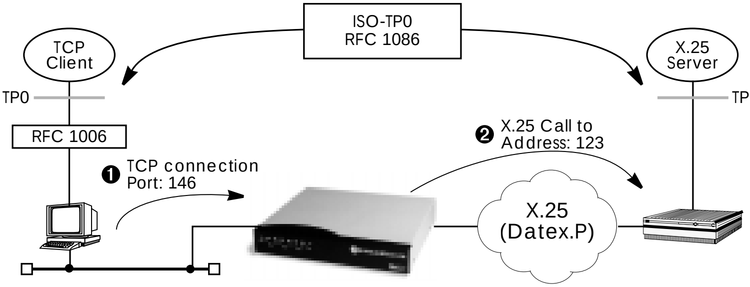
Source Link	*<X.25 interface name>*
Destination Link	local

More Info

Two common uses for this mechanism are as follows. For more detailed reference please refer to RFCs 1006 and 1086 respectively.

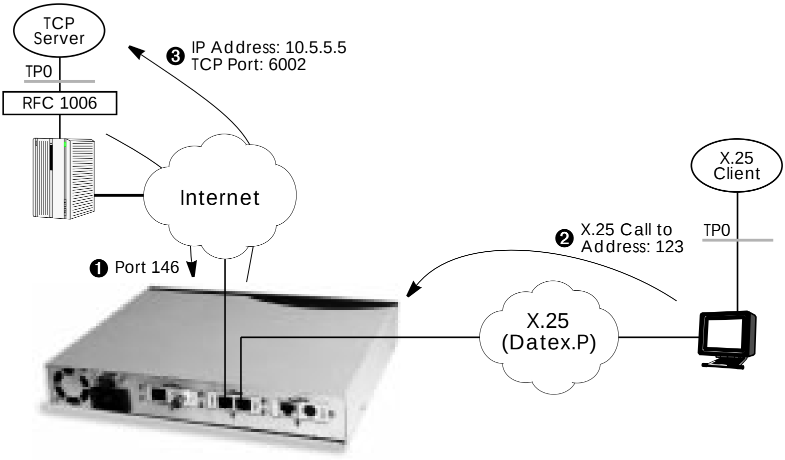
TCP Client requests connection to X.25 Server

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 TP0 packets can begin to be exchanged between the two endpoints.



X.25 Client requests connection to TCP Server

Here the TCP-Server must first initate 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).



Note: ⚠ 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.

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're going to 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 it

Configure Hardware Interface

CM-1BRI, ISDN S0 → INCOMING CALL ANSWERING → ADD →

Here you create a new entry for incoming calls on the ISDN interface to be routed to the X.25 PAD.

Item	x25_pad
Number	*<X.25 PAD's MSN>*

Next you must add the X.25 PAD as a new WAN partner.

Edit 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.

WAN PARTNER → ADD →

Create a new WAN partner interface.

Partner Name	*<X.25 PAD's partner name>*
Encapsulation	X.25 PAD

Create X.25 PAD Link

X.25 → LINK CONFIGURATION →

We need to create a new link for the X.25 PAD's partner. Select the appropriate link template from the list:

<*X.25 PAD's partner name*> (create new configuration)

Highlight the entry and enter <Return> to configure the link.

Now you can 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 **Windowsize/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.

First the routing configuration for a static routing between two X.25 partners (the X.25 PAD's partner and a remote X.25 host / partner).

X.25 → ROUTING → ADD →

Here we create an X.25 route that routes outgoing calls from the X.25 PAD to the remote X.25 network partner (X.25 host).

Source Link	<*X.25 PAD's partner name*>
Destination Link	<*X.25 network partner name*>

The partner used in the Destination Link must be configured before as an X.25 partner.

This second configuration is an example for connecting three X.25 partners, one of them the X.25 PAD's partner.

X.25 → ROUTING → ADD →

Source Link	<*X.25 PAD's partner name*>
Destination Link	<*X.25 network partner* name A>
Destination X.25 Address	1*

X.25 → ROUTING → ADD →

Source Link	*<X.25 PAD’s partner name>*
Destination Link	*<X.25 network partner* name B>
Destination X.25 Address	2*

The partners used in the Destination Links must be configured before as X.25 partners.

More Info

For further information on the X.25 PAD see “X.25 PAD” on page 145.

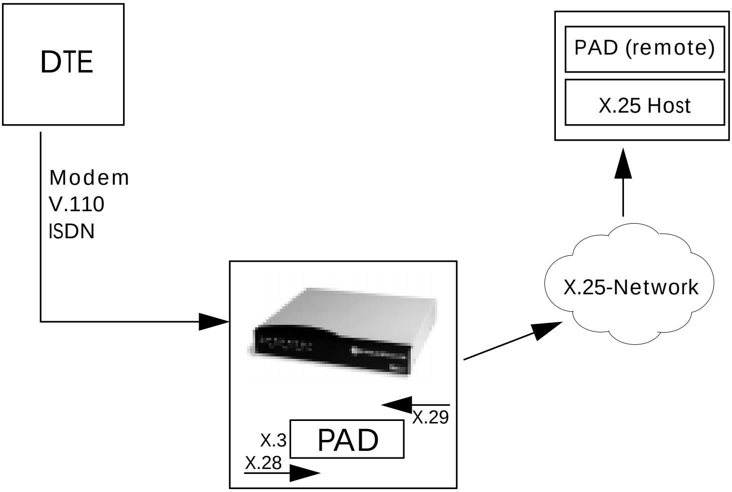
X.25 Utilities

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.



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” on page 159. 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 ***biboPPPTable*** 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

Number

The value of this parameter defines the unique number of the PAD Profile.

Possible values:

0-99	PadProfileTable numbers

The PadProfileTables 0, 90 and 91 (see below) are implemented in the BRICK.

State

This parameter describes the state of the profile.

Possible Values:

1	The Profile is valid. (**valid**)
2	The Profile is set to delete. (**delete**)

The default value is 1 resp. valid.

AutoCallDstAddr

When this parameter is set to a non-empty string, a call will automatically be established to this PAD address.

By default this variable is empty. To activate the autocall function the user must enter a value (valid X.25 address) for this variable in the ***x25PadProfileTable*** (described below) on the BRICK.

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

Number	Parameter	Description
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

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.

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 ecxception 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 IdleTimer

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*50ms 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.

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 line delete.

The default value is 0.

With the character defined, all characters entered since data were last forwarded are deleted.

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 that is to be used for output of the last line.

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, 32-126Decimal 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 ESC and ENQ.
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.

118 XCharDel

This parameter is a repetition of parameter 16.

The default value is 0.

119 XLineDel

This parameter is a repetition of parameter 17.

The default value is 0.

120 XLineDisp

This parameter is a repetition of parameter 18.

The default value is 0.

121 XForwardChar1 and 122 XForwardChar2

Allow the definition of up to two data forwarding characters in addition to parameter 3.

Possible values:

0 No additional data forwarding character

1-126 Decimal value of the character from the IA5 to be used as data forwarding character.

The default value for both parameters is 0.

123 XParity

Corresponds to parameter 21.

Possible values:

0 No parity bit checking or generation (**no_parity**)

125 XDelay

Defines how long data forwarding is to be delayed if it occurs simultaneously with a data input.

Possible values:

0 No delay of data forwarding. Only with full-duplex connections (parameter 2 is set to 1).

1-255 Number of seconds by which data forwarding is to be delayed.

The default value is 0.

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.

Each character entered resets the delay counter to 0. However, after input of an appropriate character, data forwarding starts immediately.

126 XLFInsert

This parameter is a repetition of parameter 13.

The default value is 0.

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 page 146) 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 or value 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 or value 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 or value 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>}

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 below.

Priority	PAD Function	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

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	**Profile**		
	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

Parameters	Profile		
	0	90	91
XForwardChar1	0	0	0
XForwardChar2	0	0	0
XParity	0	0	0
XDelay	0	0	0

Profile 0 is the initial profile set at the start of PAD (see page 23).

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 acknowledgement of this command.

<address>↵

Establishes a connection to the <address> (valid X.25 address) indicated after a physical connection has been established.

Also see the parameter “AutoCallDstAddr” on page 147.

^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 on page 149.

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, see the description of parameter 12 on page 153.

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 command	Valid before virtual call set-up	Valid after escaping from data transfer state
<address>	X	
PROF	X	X
SET	X	X
SET?	X	X

PAD command	Valid before virtual call set-up	Valid after escaping from data transfer state
PAR?	X	X
CLR		X
STAT	X	X
RESET		X
INT		X
RSET?		X
RPAR?		X
ICLR		X

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 profil 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 page 162). 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.

1 Escape

0 It is not possible to leave the data transfer state.

2 Echo

0 The echo mode is disabled; no echo. (**no_echo**)

3 ForwardChar

0 No data forwarding character assigned

4 IdleTimer

5 5*50ms= 250 ms

5 DevControl

0 No use of DC1 and DC3 (**no_use**)

6 SigControl

1 X.28 messages are transmitted to the DTE.

7 BrkControl

8 Data forwarding, switch to command mode

8 Discard

0 Normal data output to the DTE (**normal_data_delivery**)

9 CRPadding

0 No padding characters

10 LineFold

0 No automatic line folding

11 Speed

Detected automatically; internal value

12 FlowControl

0 No use of DC1 and DC3 for data flow control. (**no_use_DC1_DC3**)

13 LFInsert

0 No LF insertion

14 LFPadding

0 No padding characters

15 Edit

0 Editing not possible (**no_editing_user_data**)

16 CharDel

0 No editing

17 LineDel

0 No editing

18 LineDisp

0 No display

19 SigEdit

0 No editing PAD service signals

20 EchoMask

0 No echo mask

21 Parity

0 No parity bit checking or generation (**no_parity**)

22 PageWait

0 Page wait disabled

118 XCharDel

Repetition of parameter 16

119 XLineDel

Repetition of parameter 17

120 XLineDisp

Repetition of parameter 18

121 XForwardChar1

0 No additional data forwarding character

122 XForwardChar2

0 No additional data forwarding character

123 XParity

0 No parity bit checking or generation (**no_parity**)

125 XDelay

0 No delay of data forwarding; Only with full-duplex connections (parameter 2 is set to 1)

126 XLFInsert

Repetition of parameter 13

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 the section “How do I configure the routing for using an X.25 PAD?” on page 142.

minipad

`minipad` [**`-7`**] [**`-p`** *‹pktsz›*] [**`-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:

`-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.
Possible values for *‹outgocug›*: 0-9999.

`-b` *‹bcug›*
Bilateral Closed user group.
Possible values for *‹bcug›*: 0-9999.

‹x25address›
Either a standard X.121 address or an extended address.

Minipad is also useful for testing X.25 routes. To diasble X.25 connections to the minipad, *x25LocalPadCall* must be set to “dont_accept”.

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 devided up in following groups:

- Clear Causes
- Diagnostic Causes
- Restart Causes
- Reset Causes

Clear Causes

Clear causes are reported in the *ClearCause* field of the ***x25CallHistoryTable***

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

Diagnostic Causes

Diagnostic causes are reported in the *ClearDiag* field of the ***x25CallHistoryTable***

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 p4
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
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
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/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	transit network facility problem
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 surpassed
145	0x91	for interrupt
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
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

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.

1	0x01	local procedure error
3	0x03	network congestion
7	0x07	network operational

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.

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	0x1D	network out of order (PVC)

X.25 Syslog Messages

(**biboAdmSyslogSubject** = **x25**)

Note: The value <fd> used in X.25 system messages is an internal file number to discriminate between the different X.25 and TCP connections.

biboAdmSyslogMessage	***~Level***
ifc 1 vc <*vc*>: receive window exceeded, call cleared Protocol error in X.25 connection directly to BRICK (Interface 1).	err
ifc 1 vc <vc>: N(R) out of range, call cleared Protocol error in X.25 connection directly to BRICK (Interface 1).	err
Cannot rewrite call packet; Rule ... does not exist A rewriting rule has been referenced in ***x25RouteTable***, that is not defined in ***x25RewriteTable.***	err
Unable to route call to IFC ... (X.25 not supported) cannot use ifc ... for routing (ifc does not support X25) The specified target interface in an entry of the ***x25RouteTable*** does not support X.25.	err
source address too long (... bytes) The Link Layer Address (MAC) of a target interface specified in the ***x25RouteTable*** is longer than 20 Octets.	err
cannot use undefined ifc ... for routing The target interface of an entry in the ***x25RouteTable*** does not exist.	err

biboAdmSyslogMessage	***~Level***
channel misconfiguration (HIC) on ifc <*ifc*> channel misconfiguration (LTC) on <*ifc*> channel misconfiguration (HTC) on ifc <*ifc*> channel misconfiguration (LOC) on ifc <*ifc*> channel misconfiguration (HOC) on ifc <*ifc*> The channel specification of a link in the ***x25LinkPresetTable*** does not match the condition: LIC <= HIC < LTC <= HTC < LOC <= HOC	err
ifc=<*ifc*> [addr=...] vc=<*vc*> recv CALL <*SrcAddr*> -> <*DstAddr*> fac=<*fac*> 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 <*SrcAddr*> -> <*DstAddr*> fac=<*fac*> 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 <*SrcAddr*> -> <*DstAddr*> fac=<*fac*> 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 <*SrcAddr*> -> <*DstAddr*> fac=<*fac*> 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	***~Level***
ifc=<ifc> [addr=...] vc=<*vc*> recv CLEAR cause=<*causecode*> diag=<*diagcode*> 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*> send CLEAR cause=<*causecode*> diag=<*diagcode*> 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*> send RESET A X.25 RESET-REQUEST/INDICATION is being sent.	debug
ifc=<*ifc*> [addr=...] vc=<*vc*> recv RESET CONFIRM A X.25 RESET-RESPONSE/CONFIRM 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

biboAdmSyslogMessage	***~Level***
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
ifc=<*ifc*> [addr=...] vc=<*vc*> recv DIAG 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*> 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*> 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*> windowsize=<*incoming*>/<*outgoiung*> packetsize=<*incoming*>/<*outgoiung*> The call's incoming/outgoing parameters for windowsize and packetsize will used according to the given values (possibly after negotiation).	debug
ifc=<*ifc*> [addr=...] recv RESTART 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

biboAdmSyslogMessage	***~Level***
ifc=<*ifc*> [addr=...] send RESTART A RESTART packet is being sent over the given link.	debug
ifc=<*ifc*> [addr=...] recv RESTART CONFIRM A RESTART-CONFIRM packet has been received on the given link.	debug
ifc=<*ifc*> [addr=...] send RESTART CONFIRM A RESTART-CONFIRM packet is being sent over the given link	debug
ifc=<*ifc*> [addr=...] vc=<*vc*> recv ILLEGAL message An unknown message has been received on the given VC.	debug
ifc=<*ifc*> [addr=...] vc=<*vc*> invalid VC number	debug
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

biboAdmSyslogMessage	***~Level***
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 indiaction (SABM[E]) on the given link will be accepted and a UA frame being sent.	debug
ifc=<*ifc*> [addr=...] layer 2 resetted The reset request (SABM[e]) has been accepted by the peer and a UA frame has been received.	debug
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 Brick. 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

biboAdmSyslogMessage	***~Level***
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 Brick. Only the values 3 and 4 are allowed.	debug
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 Brick. Only the value 2 is allowed.	debug
txd[<*fd*>]: <*tcpaddr*>:<*port*> Connection to X.25 host addr=... failed The TCP host wanted to connect to the specified X.25 address but the Brick couldn't reach the X.25 host.	debug
txd[<*fd*>]: X.25 CALL_IND dest_addr=<*address*> An X.25 call indication for the specified X.25 address was received by the Brick.	debug

biboAdmSyslogMessage	***~Level***
txd[<*fd*>]: Connection failed - 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 addr=... An incoming X.25 connection was established	debug
txd[<*fd*>]: Connected to TCP <*tcpaddr*>:<*port*> The Brick opened an new TCP connection to the specified listening TCP host.	debug
txd[<*fd*>]: <*tcpaddr*>:<port> TCP <--> txd[<*fd*>] X.25 addr=... connected The Brick connected an incoming X.25 call to the specified TCP host.	debug
txd[<*fd*>]: Disconnect and close connection The Brick disconnects the TCP host and the X.25 host.	debug
txd[<*fd*>]: Received disconnect, cause=<*causecode*> diag=<*diagcode*> The Brick received a disconnect message from the X.25 connection. The cause and diagnostic codes of the X.25 clear indication message are shown.	debug
txd[<*fd*>]: Received disconnect The Brick 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

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 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.

Depending on which slots your communications modules are installed the LEDs for slots 1 through 6 (S1 ... S6) are as follows:

Modules		State	Meaning
	CM-X21	On	Sending or receiving a packet.

CM-X21Adapter

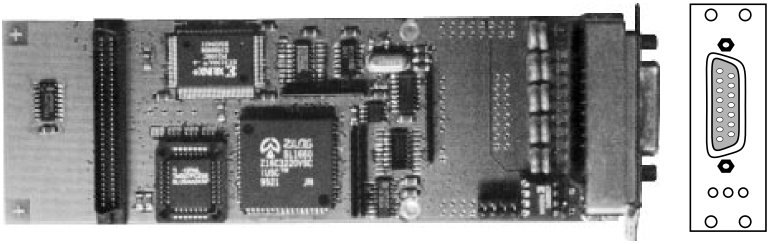


Figure 8: CM-X21 Adapter

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:

CM-X21 back plane LEDs:

Colour	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).

Note: ⚠ 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

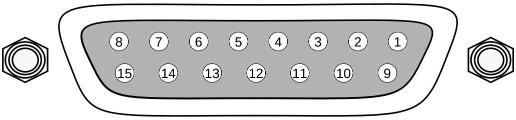


Figure 9: 15 Pin X.21 Port

The pin assignements 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	

7

FRAME RELAY

What's covered

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 encloses 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 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 ommitance 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.

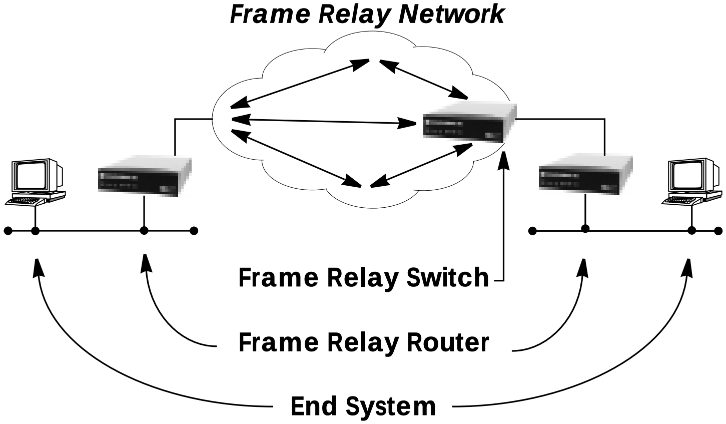
An Overview of Frame Relay Technology

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 acknowledgement routines.
- Users are guaranteed a minimum amount of bandwidth which is always available (the Committed Information Rate, 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 (CBR) and Excess Burst Rate (EBR).
- Congestion notification allows frame relay device to notify neighbouring devices (in either direction) of bandwidth bottlenecks to help maintain quality of services.

There are different types of equipment found in a typical Frame Relay Networks based on the various tasks they perform.



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. Its 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 product 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.

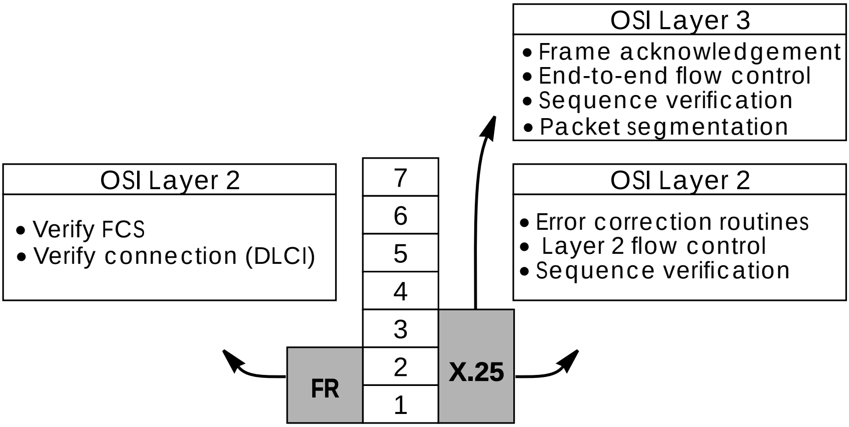
Frame Relay Switches

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.

Protocol Structure

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.



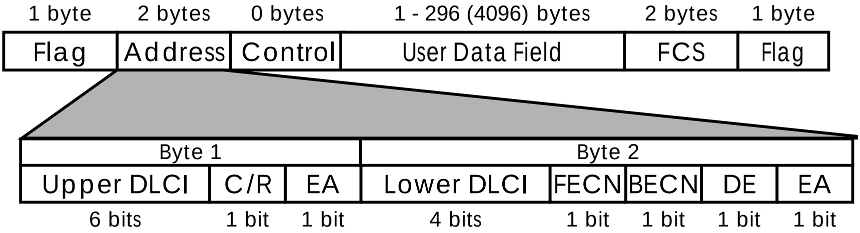
OSI Reference Model

This greatly simplifies the tasks a frame relay switch must perform.

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.

Frame Relay Frame



Flag	HDLC Flag (bit sequence: 01111110)
FCS	Frame Checksum Sequence
DLCI	Data Link Connection Identifier
C/R	Command / Response Indicator
EA	Extended Address bit
FECN	Forward Explicit Congestion Notification
BECN	Backward Explicit Congestion Notification
DE	Discard Eligibility Indicator

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., it’s value = 1) determines the address.

Congestion Notification

The FECN and BECN bits (see above) are used to notify neighbouring frame relay devices of possible congestion.

Virtual Circuits

In Frame Relay multiple connections are mapped to a single physical network connection.

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, 2^{10}, possible simultaneous virtual channels that can be multiplexed on to a PVC.

A DLCI may specify a value between 0 and 1023; however not all values are valid. As shown below some values are reserved for network management or other features such as LAPD in the D-channel.

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

NOTE: 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.

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 below.
3. The number of PVCs you are receiving.

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.

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.

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.

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.

frGlobals frDlcmiTablefrCircuitTable
frErrTable frMprTable

Overview: Frame Relay System Tables

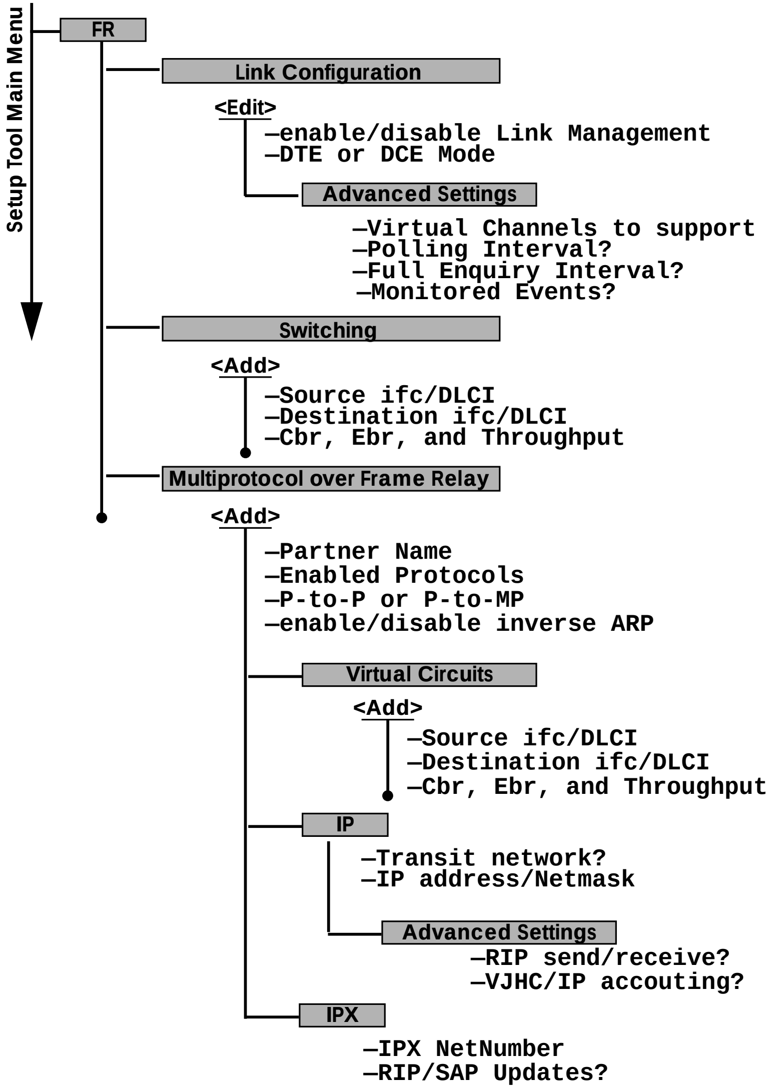
- ***frGlobals***
  Global settings for Frame Relay on the BRICK. Currently only contains the frTrapState object which is used to enabled / disable ***frDLCIStatusChange*** traps on the BRICK. (This trap indicates that the state of a particular Virtual Circuit has changed.)
- ***frDlcmiTable***
  Contains parameters for each DLCM (Data Link Connection Management) interface for each instance of frame relay service on the BRICK.
- ***frCircuitTable***
  Contains information for each Data Link Connection Identifiers and corresponding virtual circuits.
- ***frErrTable***
  Used to store important status messages reported for interfaces configured with Local Management Interface.
- ***frMprTable***
  Contains Multiprotocol Routing over Frame Relay interfaces (MP-FR) on the BRICK. 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.

Frame Relay System Messages

biboAdmSyslogMessage	***~Level***
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

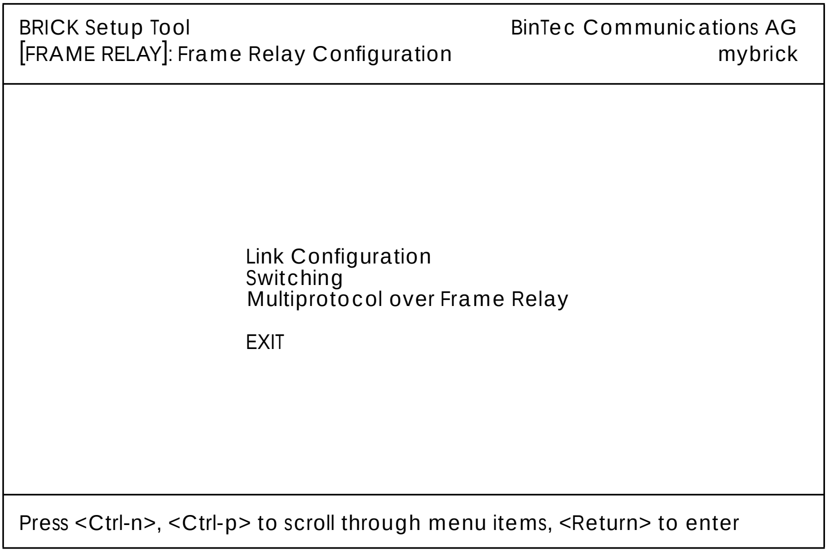
Frame Relay Setup Tool Menus

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.



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
                    Switching
                    Multiprotocol over Frame Relay

                    EXIT




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

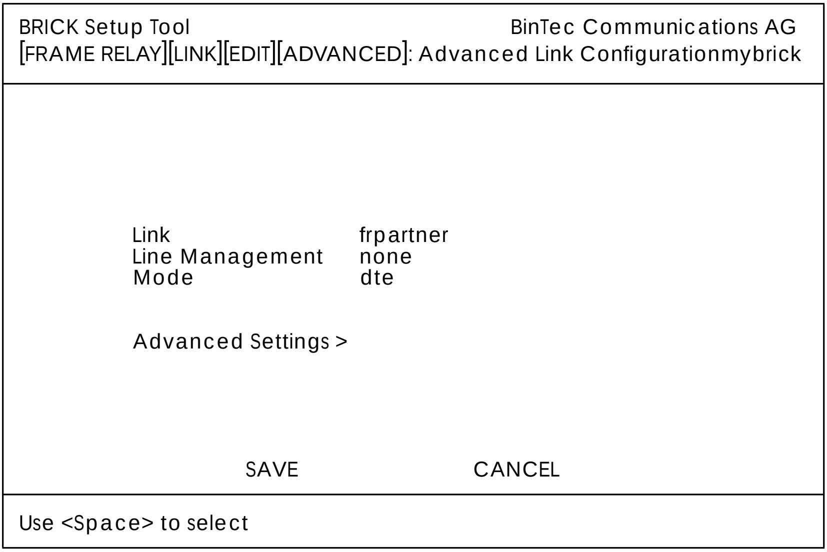
LINK CONFIGURATION contains the settings relative to the layer 2 of Frame Relay interface.

SWITCHING lists settings for each Frame Relay Virtual Circuit.

MULTIPROTOCOL OVER FRAME RELAY lists all existing MPFR interfaces configured on the BRICK.

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 hit enter) to edit link's settings.



```
BRICK Setup Tool                                    BinTec Communications AG
[FRAME RELAY][LINK][EDIT][ADVANCED]: Advanced Link Configurationmybrick




          Link                    frpartner
          Line Management         none
          Mode                    dte


          Advanced Settings >




                   SAVE                      CANCEL

Use <Space> to select
```

Link = Shows the link that is currently being edited.

Line Management = Determines whether or not link management is being performed on this link. Currently, the method described in Q.933 is supported.

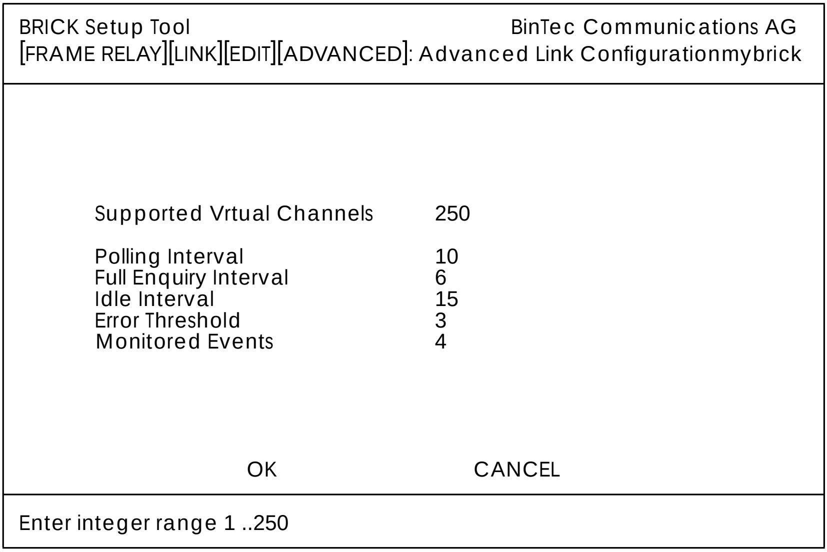
Mode = Defines the mode (DTE or DCE) the BRICK operates at for this connection. Note that one side of the link must operate as DTE and one as DCE.

Select **SAVE** to accept the settings and return to the previous menu.

Select **CANCEL** to discard all changes made since the last SAVE and return to the previous menu.

FR → LINK CONFIGURATION → 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 Configurationmybrick


        Supported Vrtual 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
```

Supported Virtual Channels = This field can be used to control how many Virtual Channels this Link supports; a maximum of 250 (default) VCs are possible.

Polling Interval = 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 BRICK. (default 10 seconds)

Full Enquiry Interval = 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).

Idle Interval = 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).

Error Threshold = When line management is enabled this field defines the maximum number of unanswered Status Enquiries the BRICK accepts before declaring the interface down (default 3 messages).

Monitored Events = 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 'MonitoredEvents' number of events the station receives 'ErrorThreshold' number of errors, the interface is marked as down (default 4 intervals).

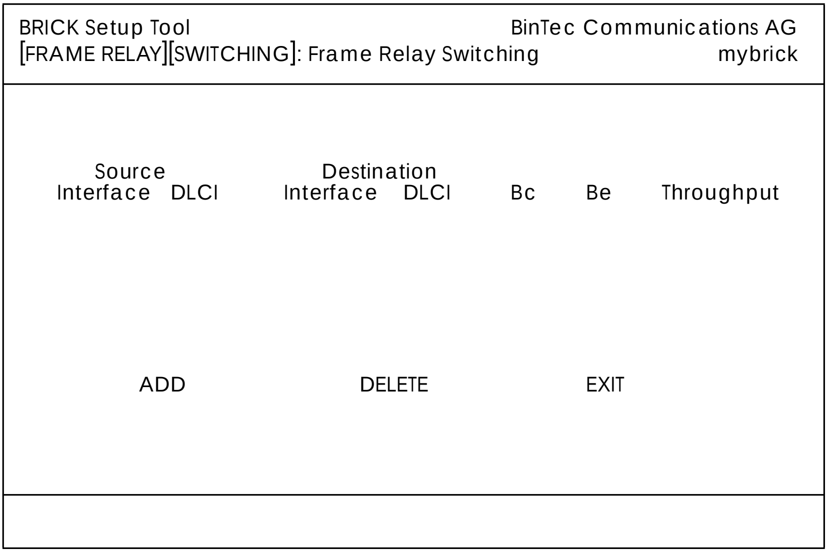
Select **OK** to accept the settings and return to the previous menu.

Select **CANCEL** to discard all changes made since the last SAVE and return to the previous menu.

FR → 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.



```
BRICK Setup Tool                                    BinTec Communications AG
[FRAME RELAY][SWITCHING]: Frame Relay Switching                      mybrick


      Source                  Destination
  Interface   DLCI        Interface    DLCI      Bc      Be     Throughput



           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.

Source Interface = Use the spacebar and scroll through the list of Frame Relay interfaces to select the source interface for this route.

Source DLCI = Defines the DLCI of the source interface for this route.

Destination Interface = Use the spacebar to scroll through the list of Frame Relay interfaces and select the destination interface.

Destination DLCI = Defines the DLCI on the destination interface to use.

Committed Burst Rate = (Abbreviated Bc) This field defines the maximum amount of data (in bits) to transfer under normal conditions.

Excess Burst Rate = (Abbreviated Be) This field defines the maximum amount of uncommitted data (in bits) to attempt deliver.

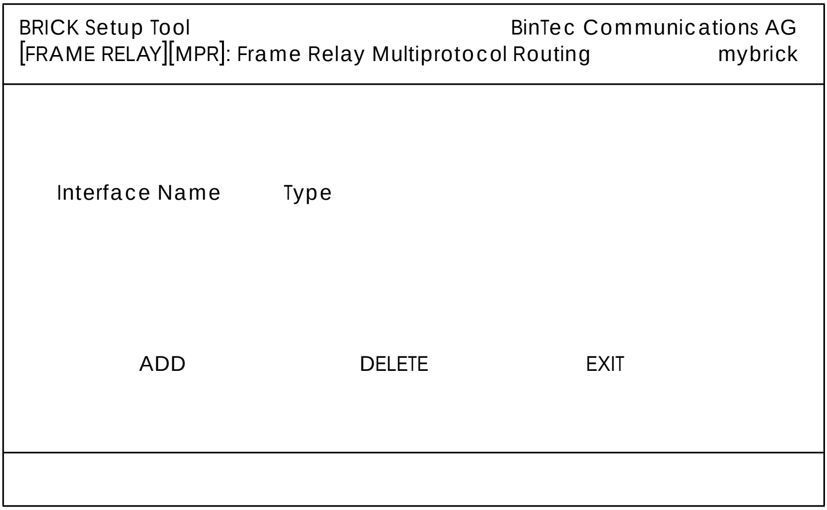
Throughput = This field defines the physical throughput for this interface (and defaults to ifSpeed).

Select **OK** to accept the settings and return to the previous menu.

Select **CANCEL** to discard all changes made since the last SAVE and return to the previous menu.

FR → 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
```

Interface Name = Identifies the interface name (taken from the ***ifDescr*** object from the ***ifTable***).

Type = Specifies whether the interface is a point-to-point, or point-to-multipoint interface.

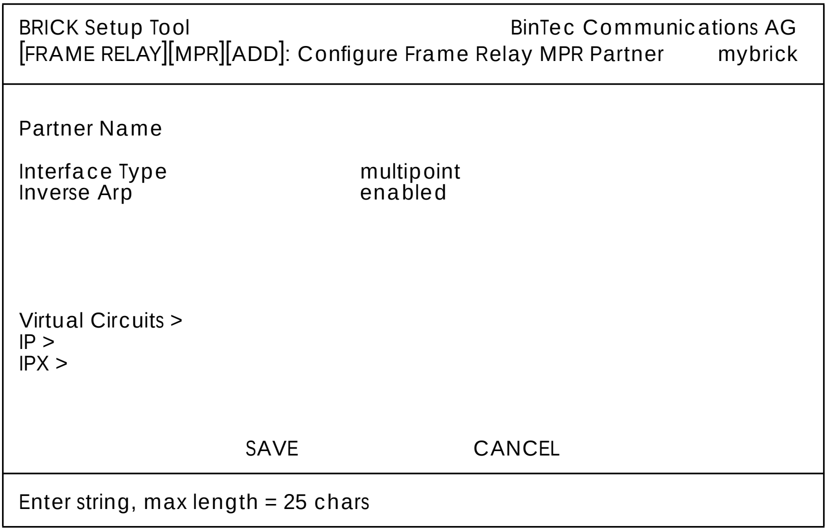
Select ADD to create a new MPFR interface. (See the EDIT/ ADD menu on the following page.)

Select DELETE to remove a MPFR interface that has been tagged (using the spacebar) for deletion.

Select EXIT to accept the interface list and return to the previous menu.

FR → **MULTIPROTOCOL OVER FRAME RELAY** → **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
```

Partner Name = Define a unique name to identify this MPFR partner.

Interface Type = Determines the interface type as being either “multipoint” or “point to point”.

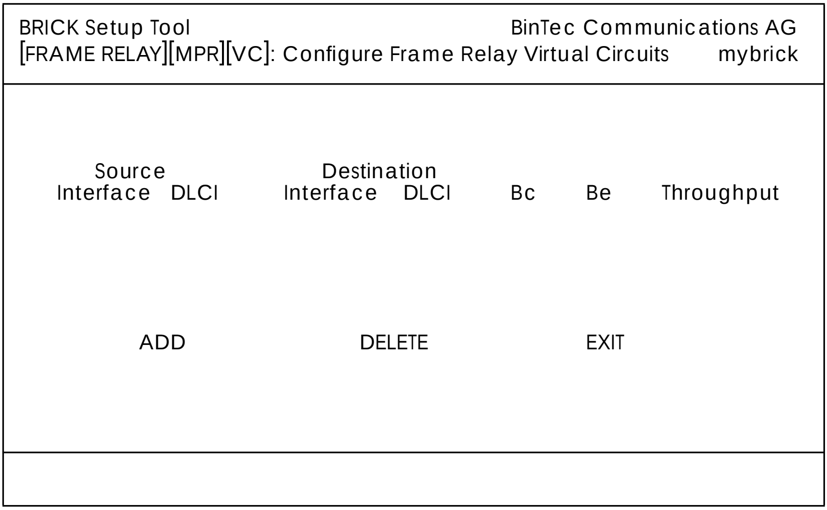
Inverse Arp = Enables / disables inverse ARP over this interface.

Select **SAVE** to accept the settings and return to the previous menu.

Select **CANCEL** to discard all changes made since the last SAVE and return to the previous menu.

FR → MULTIPROTOCOL OVER FRAME RELAY → 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.



```
BRICK Setup Tool                                    BinTec Communications AG
[FRAME RELAY][MPR][VC]: Configure Frame Relay Virtual Circuits      mybrick


       Source                 Destination
   Interface   DLCI       Interface    DLCI       Bc       Be      Throughput



             ADD                  DELETE               EXIT
```

Source Interface = Using the spacebar scroll through the list of Frame Relay interfaces.

Source DLCI = Defines the DLCI used on this interface.

Committed Burst Rate = The maximum amount of data that is guarenteed to be transferred by the service provider.

Excess Burst Rate = The amount of additional data that is uncommitted by the service provider.

Throughput = The physical throuput of this interface.

Select **ADD** to create a new Virtual Circuit for this FR interface.

Select **DELETE** to remove an existing Virtual Circuit that has been tagged (using the spacebar) for deletion.

Select **EXIT** to accept the list and return to the previous menu.

FR → MULTIPROTOCOL OVER FR → ADD → IP →

This is where you configure the IP settings for this remote MPFR partner .

Note: 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.

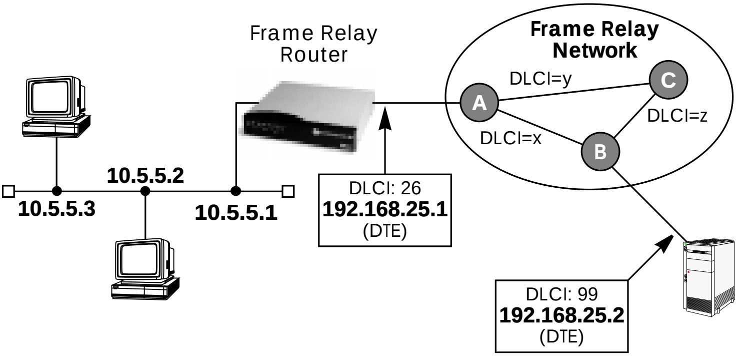
FR → MULTIPROTOCOL OVER FR → ADD → IPX →

This is where you configure the IPX settings for the remote MPFR partner.

Note: 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.

Example Configuration using Setup Tool

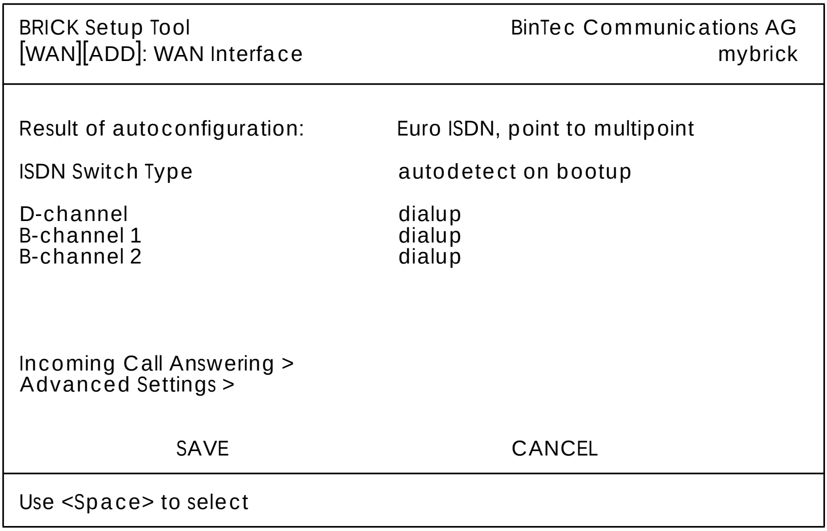
Frame Relay over ISDN Lines



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**).

1.Define the physical interface

In Setup Tool's main menu select the ISDN interface where the Frame Relay service is being received.

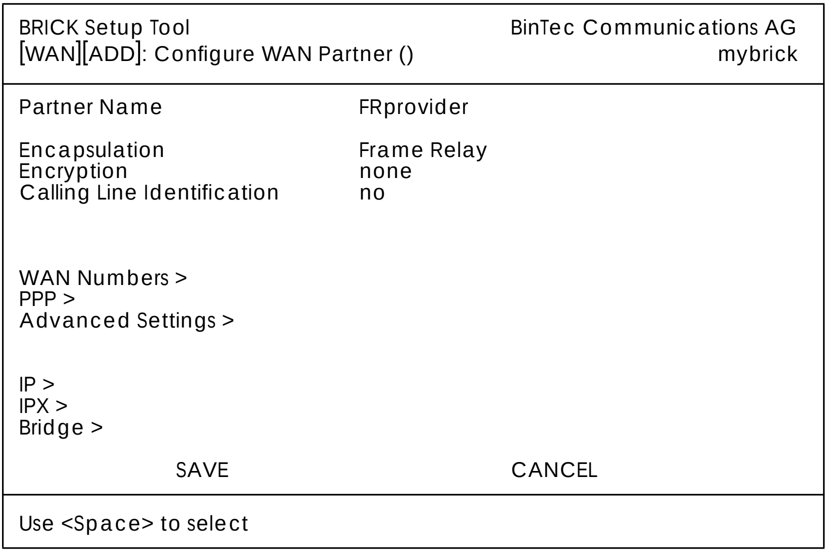


```
BRICK Setup Tool                                  BinTec Communications AG
[WAN][ADD]: WAN Interface                                          mybrick

Result of autoconfiguration:       Euro ISDN, point to multipoint

ISDN Switch Type                   autodetect on bootup

D-channel                          dialup
B-channel 1                        dialup
B-channel 2                        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.

2.**Configure a new WAN Partner**

This step defines the (physical) link to the next switch in the Frame Relay network (host A shown above). Create a new interface in the WAN PARTNER → ADD → menu.

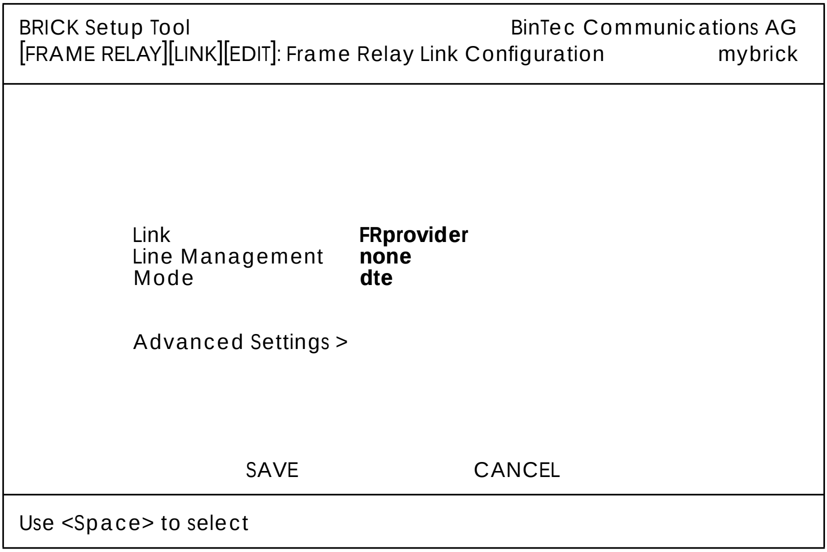


```
BRICK Setup Tool                                BinTec Communications AG
[WAN][ADD]: Configure WAN Partner ()                             mybrick

Partner Name                    FRprovider

Encapsulation                   Frame Relay
Encryption                      none
Calling Line Identification     no



WAN Numbers >
PPP >
Advanced Settings >


IP >
IPX >
Bridge >

            SAVE                              CANCEL

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.

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 hit 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.



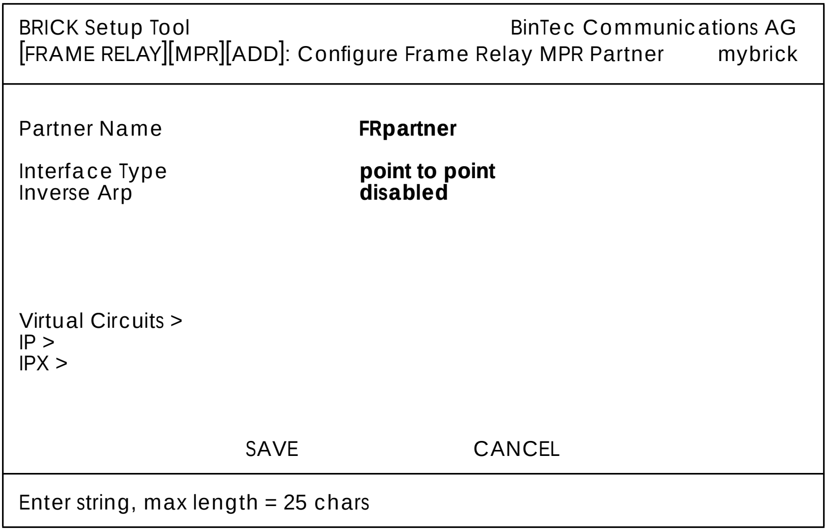
```
BRICK Setup Tool                                  BinTec Communications AG
[FRAME RELAY][LINK][EDIT]: Frame Relay Link Configuration          mybrick




            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.

4. **Configure the Multi-Protocol Routing Interface**

Go to the FR → 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.

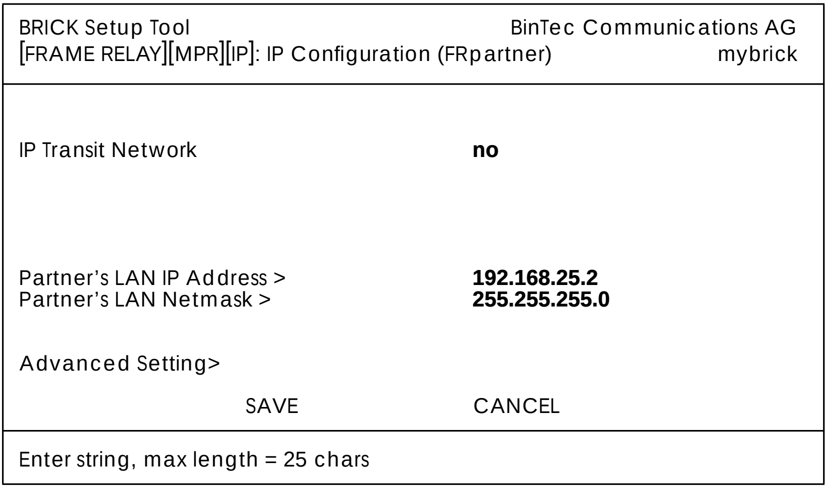
Note: When enabling protocols to route over Frame Relay please note that at current, only IP over Frame Relay has been tested on the BRICK.



```
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
```

5.**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 Netmask >                  255.255.255.0

Advanced Setting>
                    SAVE                    CANCEL

Enter string, max length = 25 chars
```