IP

February 2002

IP

Table of Contents

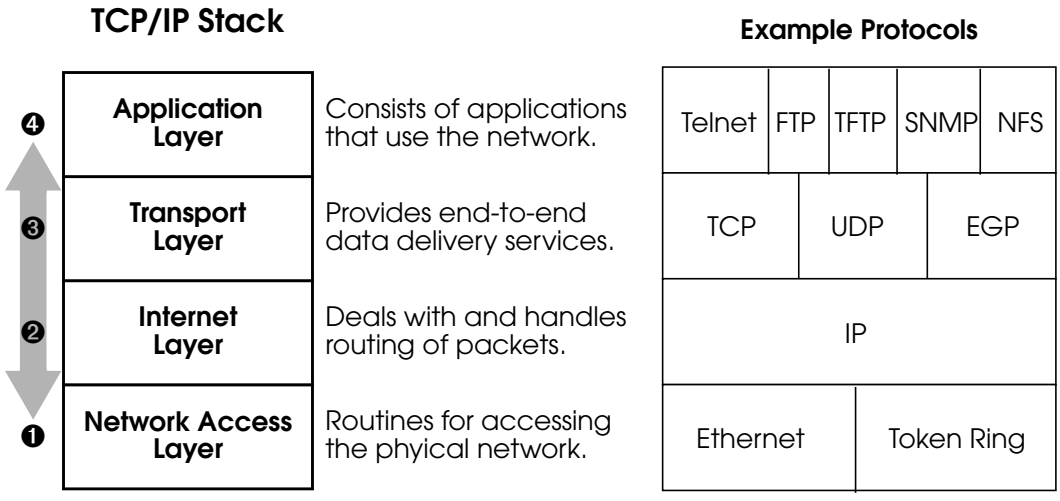
REFERENCE

1 Configuring the BinTec router as an IP Router

1.1 TCP/IP Primer

TCP (Transmission Control Protocol)/IP (Internet Protocol)[1] is often used to refer to the more general set of protocols known as the internet protocol suite. The protocols were designed to allow different types of computers and networks to communicate effectively. The internet protocol suite can be broken down into several layers, each of which provides/requires the services of an adjacent layer. These layers are often referred to as the TCP stack. The ordering and a brief description of each of these layers is shown below.

1. This and following sections provides a greatly condensed discussion of TCP/IP. For detailed information the reader is referred to a comprehensive discussion of TCP/IP such as the *Internetworking wth TCP/IP* Series by Douglas E. Comer or *TCP/IP Illustrated* by W. Richard Stevens.



Note that each layer sends and receives information from adjacent layers. When a computer receives information from the network data passes upwards through the stack until it reaches the user's application. In the opposite direction; a user application sends data over the network, data moves downward through the stack until it reaches the physical network cabling.

Depending on where in the stack the data is and the direction it is moving, each layer performs applies additional information to or removes information from the packet. This mechanism is referred to as Encapsulation and is discussed in the next section.

1.1.1 Encapsulation

The diagram on the following page shows the header information used at different layers when passing information between layers.

When information moves down the stack the sending layer applies control information to the data; this is referred

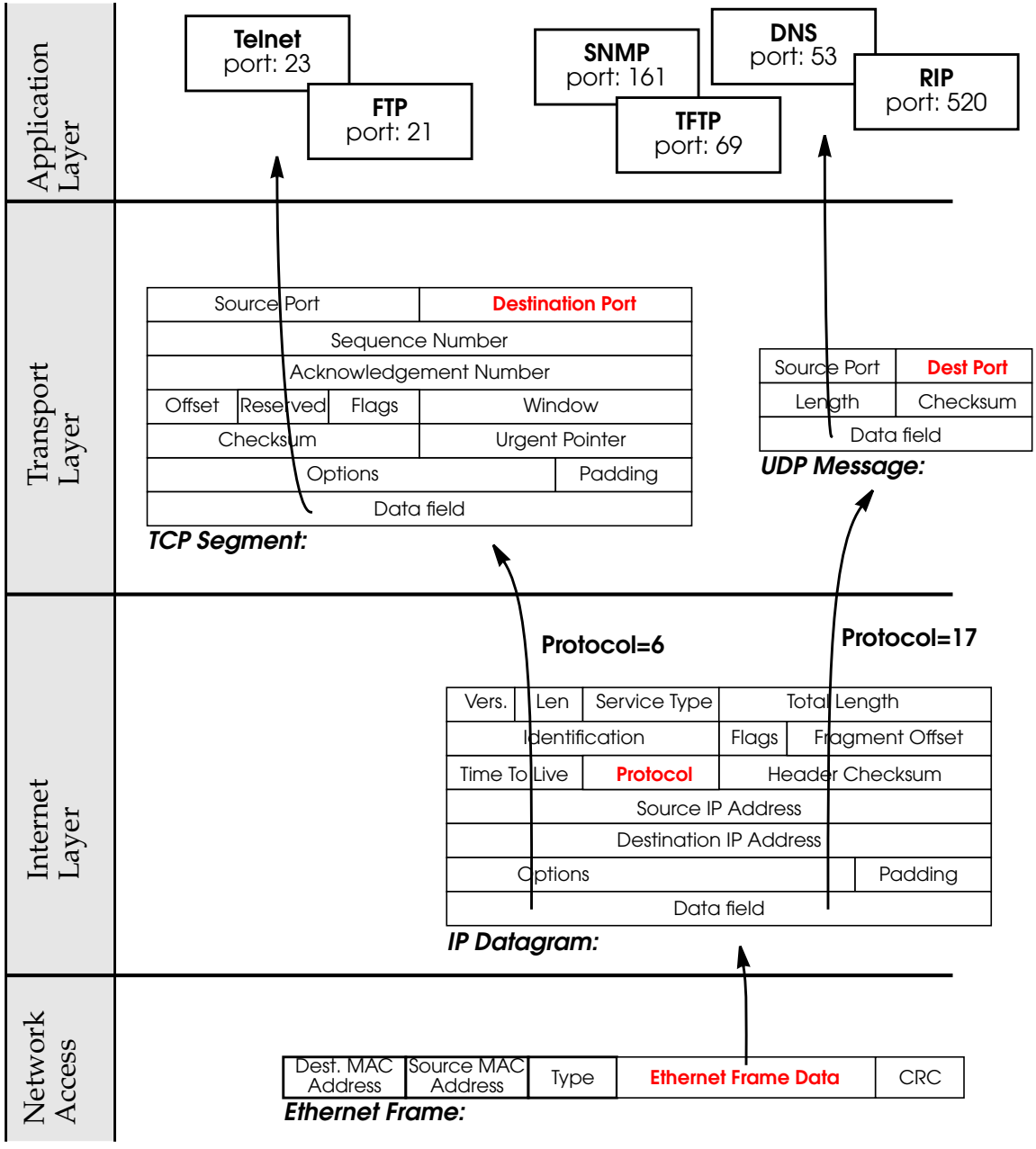
to as header information. Each layer treats all information it receives from the layer above as data.

When information moves up the stack the receiving layer reads the header information (included by the sending computer), strips the header information away, and gives the leftover data to the next layer above. As information moves up the stack each layer treats the information as header information and data combined.

- **Network Access Layer → Internet Layer**
  At this step, the contents of the ethernet frame's data field are simply passed to the Internet Layer. Note that the frame format used at this level may be slightly different (see the chapter on Ethernet Framing Types) but the concept is the same.
- **Internet Layer → Transport Layer**
  Here, the Internet Layer removes the IP header from the bytestream and decides which protocol in the Transport Layer to pass the data to using the value of the Protocol field.
- **Transport Layer → Application Layer**
  The transport layer provides two types of very different services. The transport layer is responsible for passing the information to the proper port at the receiving host. This is determined by the contents of the destination port field.

The TCP protocol is connection-oriented and provides error-detection and error-correction. Applications in the higher level layers requiring such services establish network connections using the TCP protocol.

The UDP protocol is connection-less and provides a datagram delivery service. UDP based applications are message oriented and don't require the extensive services provided by TCP.


Application Layer
Transport Layer
Internet Layer
Network Access
Telnet port: 23
FTP port: 21
SNMP port: 161
TFTP port: 69
DNS port: 53
RIP port: 520
Source Port
Destination Port
Sequence Number
Acknowledgement Number
Offset
Reserved
Flags
Window
Checksum
Urgent Pointer
Options
Padding
Data field
TCP Segment:
Source Port
Dest Port
Length
Checksum
Data field
UDP Message:
Protocol=6
Protocol=17
Vers.
Len
Service Type
Total Length
Identification
Flags
Fragment Offset
Time To Live
Protocol
Header Checksum
Source IP Address
Destination IP Address
Options
Padding
Data field
IP Datagram:
Dest. MAC Address
Source MAC Address
Type
Ethernet Frame Data
CRC
Ethernet Frame:

1.1.2 IP Addressing

The Internet Protocol delivers packets to hosts using the Source and Destination host's Address fields found in the IP header. IP addresses consist of 4 octets, 8 bits/each totalling 32 bits. Addresses are commonly written in decimal form with each octet separated by dots (hence the term dot notation).

A typical IP address is 192.168.16.8, or
1100 0000.1010 1010.0001 0000.0000 1000 in binary.

An IP address consists of a network portion that identifies the network number and a host portion that identifies the host's number on that network. The location of the dividing line that separates the network portion from the host portion is different based on the network's "Class". There are 3 network classes which can be identified as follows:

Class	Octect 1 begins with	Octet 1	Octet 2	Octet 3	Octet 4
Class A	0...	< 128	1 - 254	1 - 254	1 - 254
		Networks	Hosts		
Class B	10...	129 - 191	1 - 254	1 - 254	1 - 254
		Networks		Hosts	
CLass C	110...	192 - 223	1 - 254	1 - 254	1 - 254
		Networks			Hosts

There is a 4th network class (Class D, octect 1 > 223) that is used for multicast addresses. Multicast addresses are used to address groups of computers that share a common protocol (as opposed to a common network) at one time.

1.1.3 Subnetting

Subnetting involves dividing an IP network into separate networks. It's often used to overcome topological constraints (cable lengths) or for organizational reasons (delegation of network management tasks).

Recall that a 32 bit IP address consists of a network portion and a host portion. Local sites can extend the meaning of the network portion to include some bits from the host's portion. Essentially this moves the dividing line between the network bits and the host bits creating additional networks but reducing the number of hosts on them.

To create a subnet each network host must use a 32 bit (4 octets) network mask, or "netmask". The bit values in the mask determine where the dividing line between the net and host portions are.

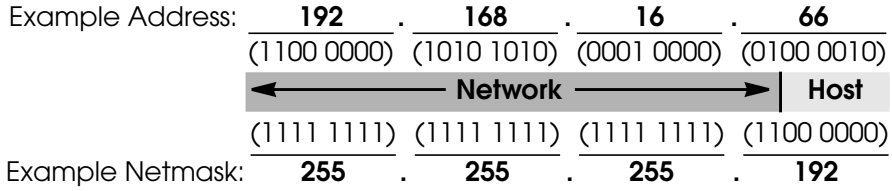
1. If the bit in the mask is **ON** (=1), the respective bit in the IP address belongs to the **NETWORK** portion.
2. If the bit in the mask is **OFF** (=0), the respective bit in the IP address belongs to the **HOST** portion.

This is where the standard network masks come from.

Class C Address:	**192** .	**168** .	**16** .	**66**
	(1100 0000)	(1010 1010)	(0001 0000)	(0100 0010)
	← **Network** →			← **Host** →
	(1111 1111)	(1111 1111)	(1111 1111)	(0000 0000)
Class C Netmask:	**255** .	**255** .	**255** .	**0**

A subnet mask commonly used on Class C networks is 255.255.255.192. This mask could be used to divide the 19.168.16.0 network into 4 subnets because the first two high order bits of the last octet are set. These 2 bits limit us

to 4 possible subnets. This would include networks: 0, (0000 0000), 64 (0100 0000), 128 (1000 0000), and 192 (1100 0000).



Once this netmask is applied, six bits of octet 4 are left over to identify the host. Six bits limit us to 64 (or 2^6) hosts per subnet. The example above identifies host number 2 ($00000010_2 = 2_{10}$).

The netmask above extends the network part to include the first two bits of octect 4 to identify the subnetwork. As stated above, 2 bits limits us to 4 subnets. The example above identifies subnetwork 64 ($01000000_2 = 64_{10}$).

So, the example address 192.168.16.66 when used with netmask 255.255.255.192, becomes equivalent to host 2 on subnet 192.168.16.64.

1.1.4 Protocols, Ports and Sockets

Together port numbers and protocol numbers identify a specific application (often referred to as a network service) on a host conputer.

The **protocol number** (the *protocol* field of an IP datagram) is an 8 bit number that identifies the transport protocol (UDP or TCP) in the Transport Layer. The Internet Layer uses this field when passing data up the stack. Some of the most commonly used protocol numbers include:

Number	Protocol and Name	
0	IP	Internet Protocol
1	ICMP	Internet Control Message Protocol
3	GGP	Gateway Gateway Protocol
6	TCP	Transmission Control Protocol
8	EGP	Exterior Gateway Protocol
12	PUP	PARC Universal Packet Protocol
17	UDP	User Datagram Protocol
20	HMP	Host Monitoring Protocol
22	XNS-IDP	Xerox NS IDP
27	RDP	Reliable Datagram Protocol
29	OSPF	Open Shortest Path Routing First

The current list of Protocol Numbers are contained in RFC 1700. This information is also available via the WWW from IANA (Internet Assigned Numbers Authority) via: ftp://ftp.isi.edu/in-notes/iana/assignments/protocol-numbers

A **port number** is a 16 bit number that identifies an application in the Application Layer. The Transport Layer uses this number (the *destination port* field of the UDP message or TCP segment) when passing data up the stack.

Both a Source and a Destination Port field is present in the IP Datagram.

For IP packets moving up the TCP stack:
Src Port = port number of the sending application on remote host.
Dest Port = port number of the receiving application on the local host.

For IP packet moving down the stack:
Src Port = port number of the sending application on the local host.
Dest Port = port number of the receiving application on the remote host

The 16 bit port number defines a limit of 65,536 (2^{16}) possible port numbers. These 65,536 ports are divided as follows.

0 → 1023	1024 → 4999	5000→ 32767	32768 → 65535
priviledged	unprivileged		
server	clients	server	client

The *privileged* ports consist of standard port numbers, often referred to as "well known ports" that identify standard network services available on a computer;. The *unprivileged* ports are non-standard ports that may be defined by local hosts. Logically server port numbers are used by server applications and client ports by client applications. The assignment of port numbers will be made clear in the example network connection diagram that follows.

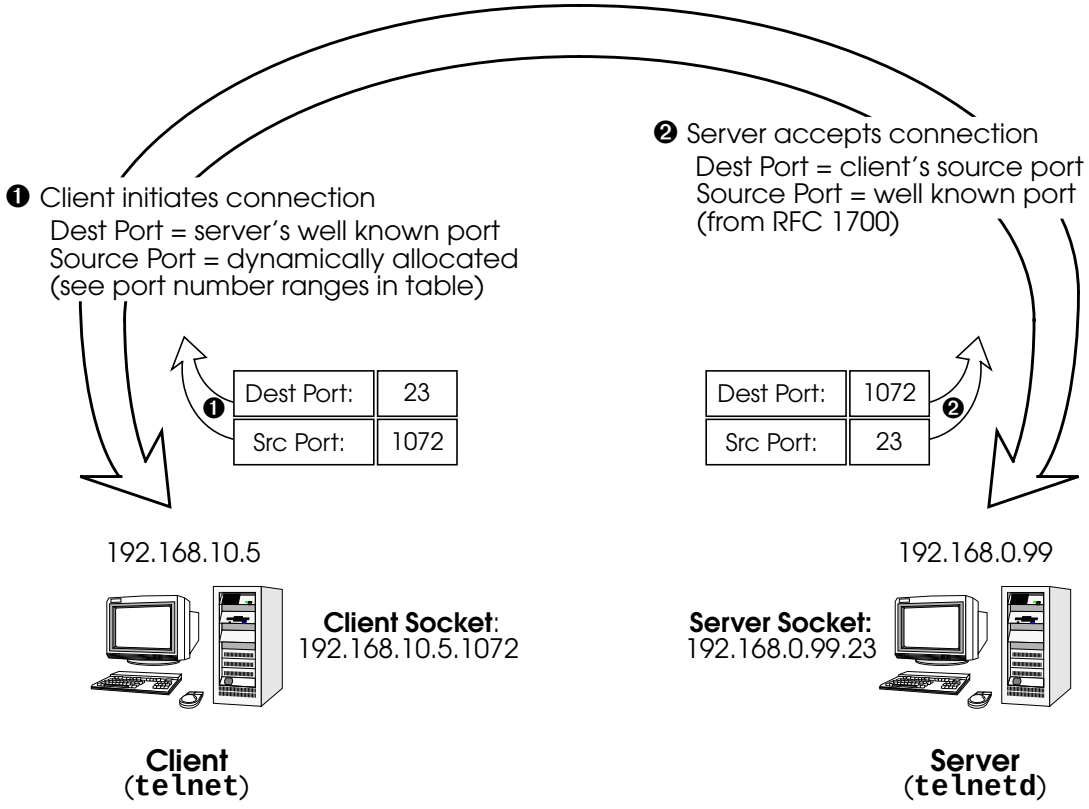
A few of the commonly used server port numbers are shown below.

Number	Port am Name	
21	FTP	File Transfer Protocol
23	telnet	The TELNET protocol/service
25	SMTP	Simple Mail Transfer Protocol
53	domain	Domain Name Service or DNS
80	HTTP	Hypertext Transmission Protocol
119	NNTP	Network News Transfer Protocol

The current list of Well Known Port Numbers are contained in RFC 1700. This information is also available via the WWW from IANA (Internet Assigned Numbers Authority) via: ftp;//ftp.isi.edu/in-notes/iana/assignments/port-numbers

A ***Socket*** identifies a specific network service on a computer (or other device). A socket consists of **IP Address**.**Port Number**. A computer at IP address 192.168.10.5 might provide a TELNET service at TCP port 23; the TCP socket is said to be: 192.168.10.5.23. Since many network services are multi-user applications a socket pair is required to identify a specific network connection. This socket pair consists of **Client Socket**:**Server Socket**. The diagram

shown below shows how ports and sockets are used in a typical network connection.



1.2 IP Routing Protocols

In general, routing can be described as a method to determine the best interface to use when forwarding an incoming packet. The term "**best interface**" means selecting the interface from the router's routing table(s) that has the lowest cost. Cost is often measured by the number of interme-

diate stations the packet would pass through before reaching its destination.

The contents of the routing table may be configured statically. A router may optionally update its routing tables dynamically by exchanging information between other routers. This exchange of routing information is defined by a routing protocol.

Although all systems route data (PCs, workstations, routers) not all systems run a routing protocol. Some networks don't necessitate routing protocols —sites where routing information doesn't change or where only one route (or a set number of routes) exists.

Routing protocols allow a router to dynamically adapt to changing network conditions and to quickly make the best routing decision in complex networks. The two most commonly used (interior)[1] routing protocols; **RIP** and **OSPF** are covered briefly below.

1.2.1 RIP

With RIP (Routing Information Protocol) a router transmits and receives routing information among other routers. Approximately every 30 seconds a router broadcasts messages to adjacent networks using information from it's current routing table. This information consists of pairs of ***IP Address:Distance*** relationships. RIP determines a route's cost by the number of "hops" (distance) it takes for a packet to reach it's final destination. For this reason RIP is sometimes referred to as a distance vector algorithm.

By listening for information sent by other routers new routes and shorter paths for existing routes, are saved to the routing table when discovered via RIP. Because intermediate routes between networks may become unreachable, RIP

1. The distinction between *Interior* and *Exterior* protocols is beyond the scope of this overview.

also removes routes older than 5 minutes (i.e. routes that haven't been verified in the last 300 seconds).

1.2.2 OSPF

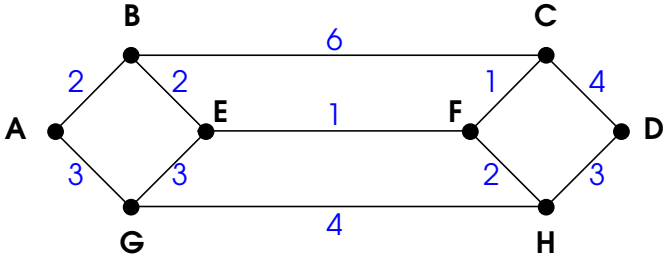
OSPF (Open Shortest Path First), is an interior routing protocol that is often used by larger network installations as an alternative to RIP. It was originally designed to address some of the limitations of RIP (when used in larger networks). Some of the problems (with RIP) that OSPF addresses include:

- **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.



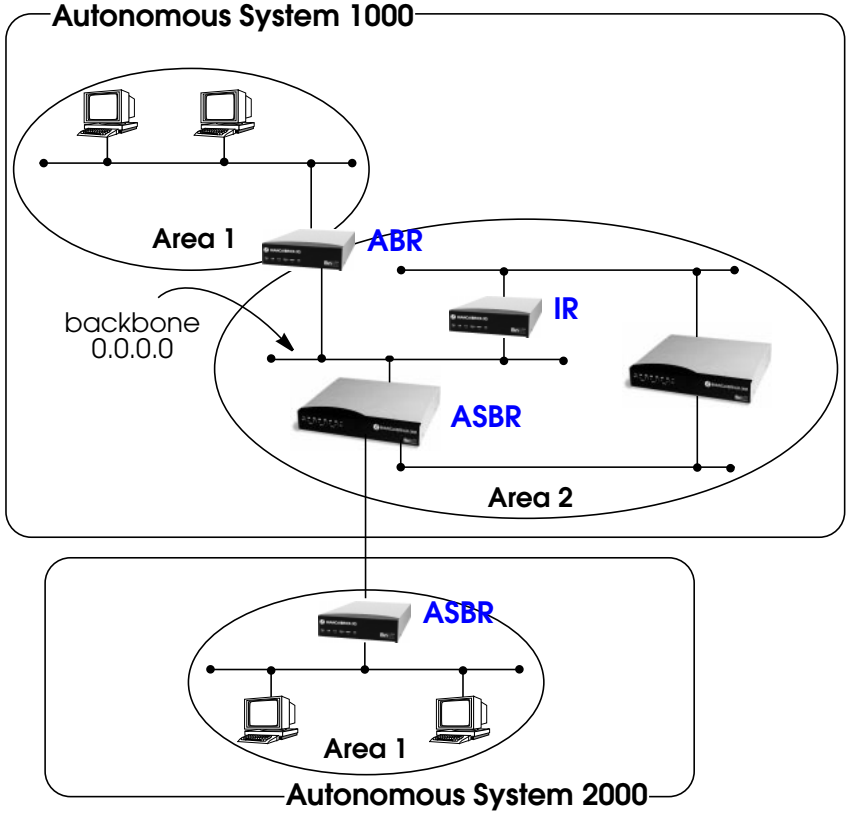
In OSPF each link has a cost associated with it (typically some fixed number divided by the bandwidth of the link). Routes are calculated and selected according to the least cost of the overall path a packet will travel. Thus in shortest-path routing the best path is also the fastest path (theoretically),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

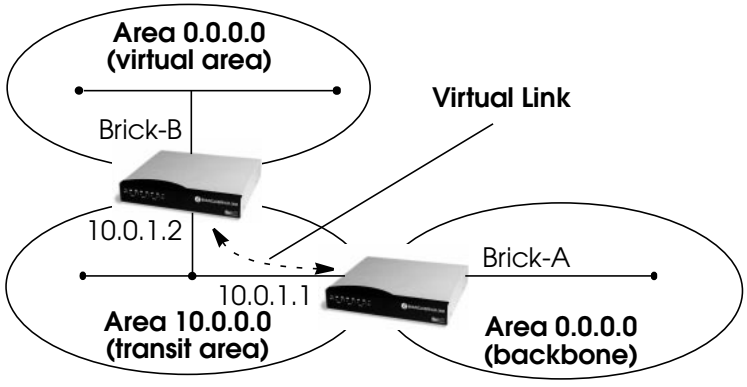


Any of the routers shown above could additionally be the Designated Router or Backup Designated Router for its respective network.

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.

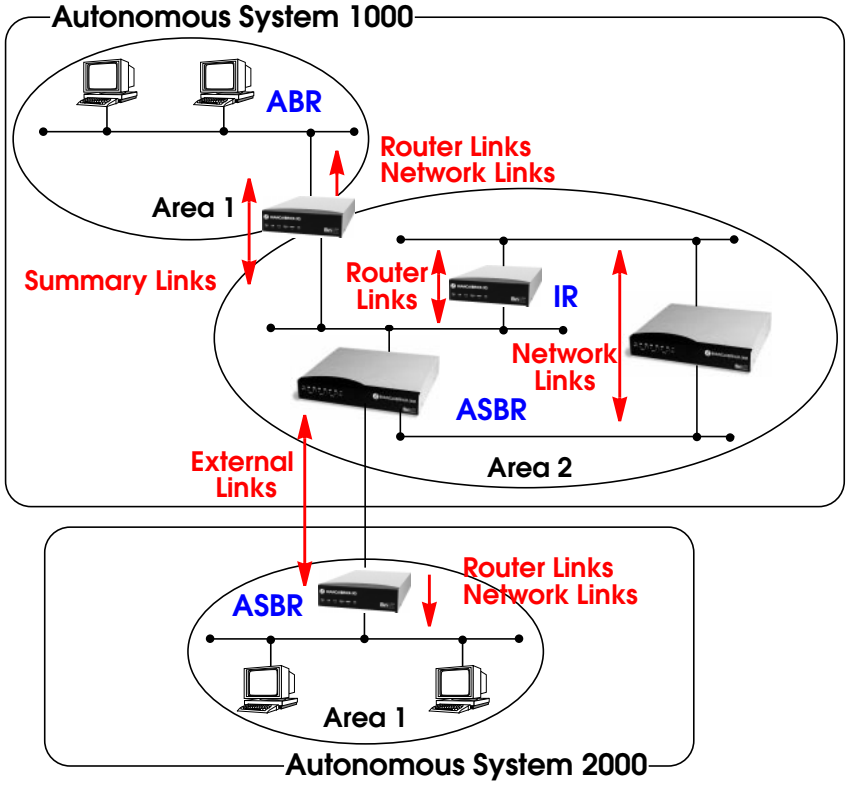
Link State Advertisement Types

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.

LSA Type	Purpose:
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.
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:

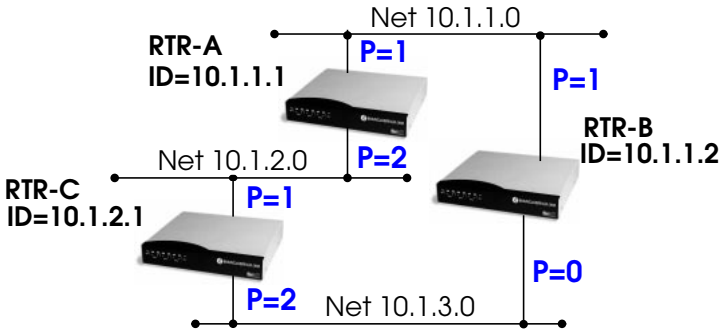
1. Share a common network.
2. Are using the same Area Number for that segment.
3. Are using the same Authentication for the segment.
4. Are using the same parameters (HELLO interval, etc.).

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 high-

est 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 BinTec router 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 BinTec router 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.

1.2.3 The Point-to-Point Protocol

The PPP (Point-to-Point Protocol) was designed as a standard method of communicating over point-to-point links. PPP actually consists of several underlying protocols, each of which perform a portion of the services offered by PPP.

In addition to HDLC (High level data link control) framing, PPP uses LCP (Link Control Protocol). LCP is used to negotiate options pertaining to the data link. Some of the options which can be negotiated using LCP are:

1. **Maximum-Receive-Unit**: The MRU specifies the maximum size of data packets to be processed over this link. The default value is 1500 bytes.
2. **Authentication-Protocol**: This option is used to specify which authentication procedure (CHAP or PAP), if any should be used for this link.
3. **Quality-Control**: This option specifies whether or not the quality of the link should be monitored.
4. **Protocol-Field-Compression**: This option specifies which, if any, protocol fields should be compressed over the link. Using this option could allow a higher throughput rate to be achieved.

Establishing a PPP connection

Establishing a PPP connection is accomplished step by step, in three simple phases.

1. Before any user data can be sent, the communicating partners must agree on which communications parameters the connection will use. This is accomplished using LCP mentioned earlier. Step by step, each side of the connection negotiates with the other to establish the best possible communications parameters.
2. The second phase is where the optional process are actually performed. This is where the authentica-

tion procedure (CHAP or PAP) would be performed if specified in phase 1. Additional parameters agreed upon in phase 1 are also performed here as well; i.e. if the Quality-Control option was agreed upon, a mechanism would then be started between the communicating partners, which helps to ensure a stable and secure connection.

3. The last phase of connection establishment involves making the connection available to the various network protocols. The actual closing of links is performed by LCP. However, as each network connection (multiple network connections are possible) closes, LCP may keep the physical connection open. PPP does not specify a default time limit to wait before automatically closing connections. Connections can be closed manually, or by setting a default wait time.

Debugging and Status Info of PPP Connections

The **pppSessionTable** simplifies the debugging of PPP connections and provides a means of getting reliable status information about active PPP connections. The table is read only. For a brief description of the table´s variables and values, see the MIB Reference.

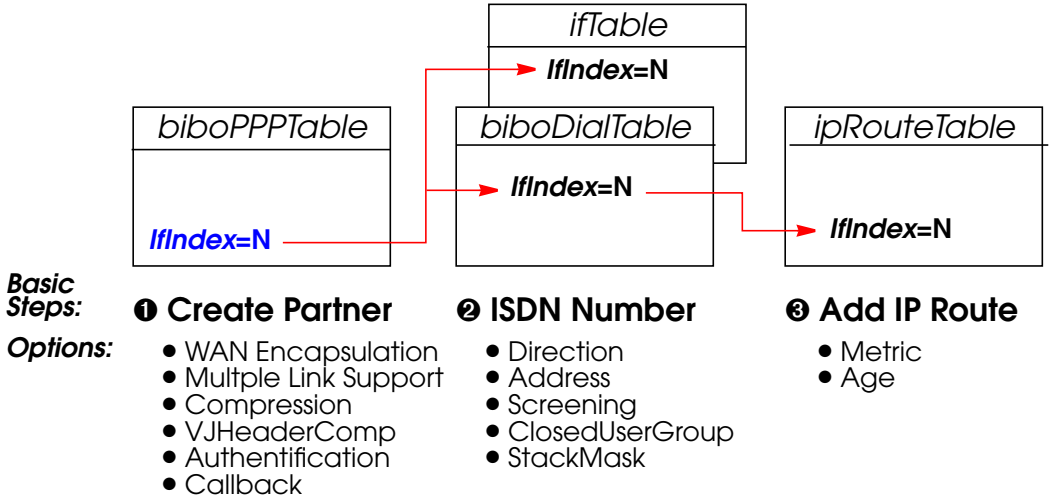
1.3 DialUp IP Interfaces

Creating Dial-Up PPP interfaces on the BinTec router basically involves three steps which correspond to creating the system tables shown below. Many different options are available when creating the respective table entries. An overview of the types of options available is shown in the diagram below.

- Create the PPP Partner Interface—***biboPPPTable***

- Identify the Partner's ISDN Number—***biboDialTable***
- Create an IP Route for the Partner—***ipRouteTable***

Note that after creating the partner interface in the ***biboPPPTable*** the BinTec router generates a new ***ifIndex*** value and automatically creates an entry in the ***ifTable***. This ***ifIndex*** is very important since it identifies a specific software interface; it must be used when creating other system table entries to associate settings with the respective software inferface.



An example SNMP shell session describing how to create a standard ISDN DialUp PPP interface is shown below. Most of the available optional settings mentioned above simply involve setting the respective variable to an appropriate non-default setting.

For a more detailed description of these optional settings and how to properly configure them, please refer to the section DialUp Options.

1.3.1 Creating a DialUp IP Interface

Step 1

The first step is to create the ***biboPPPTable*** entry. The only index variable in this table is the ***Type*** field; it defines this partner as either an **`isdn_dialup`** or **`leased`** line partner (Although leased line interfaces do appear here, leased-line partner interfaces can not be created using the ***biboPPPTable***.).

To create the table entry we can set this field and adjust the other entries as needed (objects not explicitly set revert to their default values).

```
mybrick: admin> biboPPPType=isdn_dialup
05: biboPPPType.1.5( rw):      isdn_dialup

mybrick : biboPPPTable > biboPPPTable
inx IfIndex(ro)              Type(*rw)          Encapsulation(-rw)
    Keepalive(rw)            Timeout(rw)        Compression(rw)
    Authentication(rw)       AuthIdent(rw)      AuthSecret(rw)
    IpAddress(rw)            RetryTime(rw)      BlockTime(rw)
    MaxRetries(rw)           ShortHold(rw)      InitConn(rw)
    MaxConn(rw)              MinConn(rw)        Callback(rw)
    Layer1Protocol(rw)       LoginString(rw)    VJHeaderComp(rw)
    Layer2Mode(rw)           DynShortHold       LocalIdent

05  10006                    isdn_dialup        ppp
    off                      3000               none
    none
    static                   4                  300
    5                        20                 1
    1                        1                  disabled
    data_64k                                    disabled
    auto                     0
```

The new dialup interface created above displays standard (default) setting consisting of the following characteristics:

Encapsulation	**`ppp`**	(See: WAN Encapsulation)
IP Address	**`static`** IP Address	(See: IP Address Settings)

Compression	**none**	(See: Compression)
Authentication	**none**	(See: Authentication)
MultiLinkSupport	**1** B-channel	(See: Mulitple Link Support)
ShortHold	**20** seconds	(See: Mulitple Link Support)
Callback	**disabled**	(See: ISDN Callback)
Layer1Protocol	**data_64k**	(See: Layer 1 Protocol)
LoginString	"empty"	(See: Auto-Login)
VJHeaderComp	**disabled**	(See: Header Compression)
Layer2Mode	**auto**	(See: Layer2Mode)
LocalIdent	"empty"	(See: PPP Identification)

Step 2

Next we need to define the partner's ISDN telephone number in the ***biboDialTable*** by associating it with the ***IfIndex*** created in step 1. As shown in step 1 display the contents of the ***biboPPPTable*** and locate the new interface index. The **inx** number for the new table entry is displayed to the screen when the entry is created. In most cases this will be the ***IfIndex*** field of the last table entry.

You may optionally verify this value is also present in the ***ifTable*** (in the ***Index*** field of the last table entry). In the ex-

ample below the ISDN number 555 is associated with our new software interface 10006

```
mybrick: biboDialTable > biboDialIfIndex=10006 biboDialNumber=555

06: biboDialIfIndex.10006.6( rw):        10006
06: biboDialNumber.10006.6( rw):        "555"

mybrick: biboDialTable > biboDialTable

inx IfIndex(*rw)          Type(-rw)          Direction(rw)
    Number(rw)            Subaddress(rw)     ClosedUserGroup(rw)
    StkMask(rw)           Screening(rw)

06  10006                 isdn               both
    "555"
    0xffffffff            dont_care

mybrick : biboDialTable >
```

Several options are also available in the Dial Table. Unless otherwise set, the following default values are used.

Type **isdn**
A Type of isdn_spv is used for a semi-permanent link and is used in connection with the German 1TR6 protocol.

Direction **both**
Direction may be limited to **incoming** or **outgoing**.

ClosedUserGoups
Not used by default but may be set for sites receiving ISDN Closed User Groups services.

StkMask **0xffffffff** means all available ISDN stacks.

Screening **dont_care** (See: ISDN Screening)

Step 3

Now we need to create the appropriate routing table entry for this partner. One, possibly two, routing entries must be created in this step depending on whether a transfer net-

work is being used . The example below assumes no transfer network is being used.

Using our partner's IP address (192.168.5.5) we add an indirect route to the partner's network. As before we need to associate this entry with the ***IfIndex*** for our partner interface from step 1 (10006).

```
mybrick: biboPPPTable > ipRouteIfIndex=10006 ipRouteDest=192.168.5.0
                        ipRouteType=indirect

03: ipRouteIfIndex.192.168.5.0.3( rw):  10006
03: ipRouteDest.192.168.5.0.3( rw):     192.168.5.0
03: ipRouteType.192.168.5.0.3(-rw):     indirect

mybrick: ipRouteTable> ipRouteTable

inx Dest(*rw)          IfIndex(rw)      Metric1(rw)      Metric2(rw)
    Metric3(rw)        Metric4(rw)      NextHop(rw)      Type(-rw)
    Proto(ro)          Age(rw)          Mask(rw)         Metric5(rw)
    Info(ro)

 03 192.168.5.0        10006            0                -1
    -1                 -1               0.0.0.0          indirect
    netmgmt            355              255.255.255.0    -1
    .0.0

mybrick : ipRouteTable >
```

This route can also be created using the ifconfig command. (See External Commands for the command syntax of ifconfig in the chapter on the SNMP Shell).

1.3.2 DialUp Options

This section describes the various options found in the ***biboPPPTable***.

WAN Encapsulation

The ***biboPPPEncapsulation*** object defines the method used to encapsulation data packets transmitted over the ISDN link. The ISDN partner must also support the specified method for connections to be established. The type of encapsulation selected here also limits the types of protocols that can be routed over the interface. Possible encapsulation types and the protocols they support are shown below.

By default **ppp** encapsulation is used. Special information regarding some of the encapsulations (checkmarked in red) is contained below.

Note: In this table 5 means that the encapsulation may be configured but is not useful in most cases.

biboPPPEncapsulation	Supported Protocols			
	IP	IPX	Bridge	X.25
ppp	✓	✓	✓	
x25				✓
x25_ppp	✓	✓	✓	✓
ip_lapb	✓			
ip_hdlc	✓			
mpr_lapb	✓	✓	✓	
mpr_hdlc	✓	✓	✓	
frame_relay	✓	✓	✓	✓
x31_bchan				✓
x75_ppp	✓	×	×	
x75btx_ppp	✓	×	×	
x25_nosig				✓

biboPPPEncapsulation	Supported Protocols			
	IP	IPX	Bridge	X.25
x25_ppp_opt	✓	✓	✓	✓

Encapsulation: x75_ppp

x75_ppp encapsulation is used for asynchronous PPP over X.75 and is mainly used for accessing commercial service providers such as CompuServe Online Services. The *biboPPPLoginString* object is intended to be used with this encapsulation to automate the logon process with such service providers.

A typical logon string that might be used for logging onto Compuserve directly is shown below:

```
"-d1 \n e: CIS\n ID: 12345,6789/go:pppconnect\n
  word -d1 secret\n PPP"
```

Encapsulation: x75btx_ppp

x75btx_ppp encapsulation can be used to access CompuServe Online Services indirectly via the German Telekom's T-Online gateway. The *biboPPPLoginString* can be set to include the appropriate login information to automate the login process to the service provider.

A typical logon string that might be used for logging onto Compuserve via the T-Online gateway is shown below:

```
".n\ :000000 000327278259\n gabeseite 11 # # Name: CIS\n
  ID:12345,6789/go:pppconnect\n wor -d1 secret\n PPP"
```

Encapsulation: x25_nosig

x25_nosig (no signalling) encapsulation uses the same encapsulation method as x25. The only difference between the two is that with x25_nosig outgoing ISDN calls are not signalled as X.25 calls but as a data transfer

call (DSS :Bearer Service unrestricted digital info without LLC).

Encapsulation: `x25_ppp_opt`

x25_ppp_opt encapsulation provides a special case of the *x25_ppp* encapsulation. It allows the BinTec router to determine whether an incoming call is an X.25 call or a PPP call even if no outband authentication (by CLID) is possible. This is done by scanning the first incoming data packet.

Dial-in partners that can't be authenticated outband (CLID) are then given an X.25 connection via ISDN, or optionally a PPP connection, if they can be authenticated inband by using CHAP or PAP.

Once the dial-up connection is established only one protocool,, X.25 or IP, may be routed over the interface.

Note:

You will need one WAN partner definition for X.25, where the x25_ppp_opt encapsulation is selected, and one or more for PPP con nections (authentication via PAP, CHAP or RADIUS)

IP Address Settings

The ***biboPPPIpAddress*** object defines the BinTec router's relationship to this host regarding its IP address. By default, **`static`** is used here. This assumes the PPP partner already has a fixed IP address configured and the appropriate IP routes (using this address) are already configured in the ***ipRouteTable***.

This object can also be set to dynamic_server or dynamic_client which is explained below.

dynamic_server This means the BinTec router will attempt to assign this partner a new IP address at connection time. The next available IP address is retrieved from the ***biboPPPIpAssignT-***

able.
If the dialup partner is configured to request a primary and/or secondary nameserver address, the BinTec router responds by sending the current values of the ***biboAdmNameServer*** and ***biboAdmNameServ2*** objects.

dynamic_client This means the BinTec router will accept its own IP address (for this dialup interface) from this partner at connection time. If not already set the BinTec router requests the primary and/or secondary nameservers address. If the dialup partner provides this information the BinTec router sets the ***biboAdmNameServer*** and/or ***biboAdmNameServ2*** objects.

Compression

The ***biboPPPCompression*** object defines the type of data compression (performed in software) to use with this partner. The BRICK-XS, BRICK-XM, and V!CAS support both STAC and V42bis data compression. On the BRICK-XL V42bis data compression is supported in software; STAC compression will be performed in hardware via an additional feature module available in a future release.

Data compression can only be used in connection with the ***Encapsulation*** settings shown below. Although its possible to configure any Compression–Encapsulation combination in the ***biboPPPTable,*** compression over the link will only be achieved when configured as follows.

biboPPPCompression	*biboPPPEncapsulation*
stac	ppp
stac	x25_ppp
v42bis	mpr_lapb

biboPPPCompression	*biboPPPEncapsulation*
v42bis	ip_lapb

STAC compression is supported according to RFC 1974 and 1962 (PPP *Stac LZS Compression* and *PPP Compression Conrol Protocol* respectively) standards, which, depending on the data can increase performance variably. Typically, performance is increased by a factor of 2 to 3; with the best case scenario at a factor of 30. The Stacker LZS algorithm is developed by Hi/fn Inc.

STAC compresssion on the BinTec router is also compatible with Cisco's proprietary STAC implementation which is automatically detected at connection time.

Note:

Due to heavy system requirements made by this algorithm only
4 instances of can be used simultaneously, for example,
4 partner connections @ 1 B-Channel each, OR
2 partner connections @ 2 B-Channels each, etc.
This limit does not affect the BRICK-XS or V!CAS products.

Authentication

The ***biboPPPAuthentication*** determines the type of authentication to use when establishing dialup connections with this partner. The types of authentication methods available here are: **chap**, **pap**, **ms_chap**, **ms_chapv2** and **radius**. The value **both** can be set and means that both PAP and CHAP should be used. The value **all** means **chap**, **pap**, **ms_chap** will be used.

Mulitple Link Support

Multiple Link Support allows data connections to and from dialup ISDN partners to be run over multiple channels concurrently. By dynamically allocating bandwidth (automatically openning and closing additional channels) greater

throughtput rates can be acheived when needed. For dialup ISDN connections this of course can lead to increased costs.

Every 5 seconds the BinTec router calculates the current throughput for each dialup interface that is open. When throughput rises above a preset upper bound additional ISDN channels are opened. If throughput drops below a specified level unneeded channels are closed.

Using the following fields of the ***biboPPPTable*** the BinTec router determines how multiple link support should be handled for the specified partner.

`InitConn` InitConn defines the number of ISDN channels to initially open when a connection is established with this partner. By default 1 B-Channel is opened.

`MaxConn` MaxConn defines the maximum number of channels to have open at any given time for connections to this partner. By default the max number of chanels is 1.

`MinConn` MinConn defines the minimum number of channels to keep open with this partner. If throughput drops, the number of open channels will never become less than this value. The only exception is when ***ShortHold*** (or ***DynShortHold***; see Short Hold) timer runs out. By default 1 channel is always kept open.

Short Hold

Short Hold means that an existing ISDN connection can be automatically taken down by waiting a specified (configurable) amount of time once the line becomes silent. Silent

here means that for the adjusted time no more data packets have been going out. Data, which is generated by the BinTec router itself cyclicly, like for example RIP broadcasts and KeepAlives are not considered.The BinTec router supports two types of Short Hold, Static and Dynamic. Note that Dynamic Short Hold can only be used if the ISDN AOCD[1] (advice of charge during the call) feature is activated.

Static Short Hold

Static Short Hold and involves setting the ***biboPPPShortHold*** variable to the amount of time (in seconds) to wait before disconnecting the line. Though less flexible than Dynamic method static short hod can always be used.

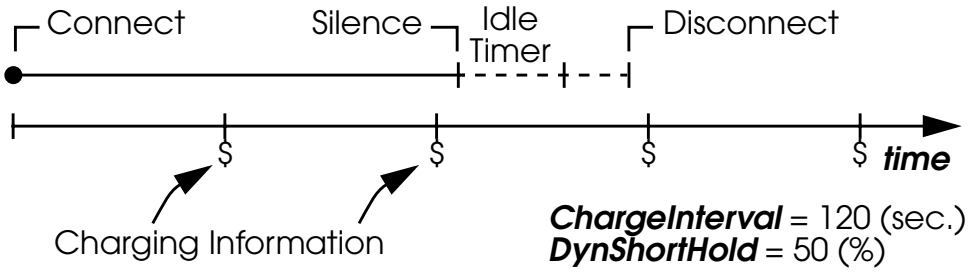
Dynamic Short Hold

Dynamic short hold provides greater flexibility in determining when the line is taken down. Here the ***biboPPPDynShortHold*** variable is used. This object defines the percentage of the current Charging Interval (sent by the ISDN and saved on the BinTec router in the ***biboPPPChargeInterval*** object) to wait before closing the link.

For example, if ***biboPPPDynShortHold*** is set to 50 (%), and the last measured ***biboPPPChargeInterval*** was 120 seconds, the idle timer is set to 60 seconds. If the ChargeInterval length changes (weekday/weekend,

1. Called *»Übermittlung der Tarifeinheiten* während *der Verbindung«* in Germany

time of day, etc.) the idle timer setting adjusts accordingly.



Recommended Dynamic Short Hold Settings:

- For ***interactive connections*** (e.g. telnet) you should specify a rather high Dynamic Short Hold percentage (e.g. 80-90) to avoid frequent disconnects due to short periods of inactivity.
- For ***internet connections*** (WWW, http, etc.) you should specify a medium to high Dynamic Short Hold percentage (e.g. 50-80) to avoid frequent disconnects due to waiting periods.
- For ***data connections*** (e.g. ftp) you should specify a low Dynamic Short Hold percentage (e.g. 10-40) to avoid unnecessarily waiting—and incurring charges—once a transfer is complete.

Note:

If configured, the Static Short Hold timer will *always* take precedence over Dynamic Short Hold to avoid permanent connections.
Make sure to set the Static Short Hold to a value greater than the length of a charging unit if you want Dynamic Short Hold to have any effect.
For example, in Germany there are different maximum charging unit lengths for different tarif zones (City = 4 minutes, long distance calls = 2 minutes), so you can set the *Static* Short Hold to 245 (>4 minutes) for City connections, and to 125 (>2 minutes) for long distance calls, to avoid nullifying your Dynamic Short Hold settings.

Note:

If you are using Dynamic Short Hold in connection with channel bundling, please note that the channels are released one by one, keeping open each channel until shortly before the next advice of charge is expected for this channel, thus maximizing the connection time without further cost. The call will of course be disconnected immediately if either side actively closes it.

Permanent Connection

The **PPPShortHold** variable in the **biboPPPTable** is the time in seconds which must elapse after no further data exchange occurs before the link is terminated.
By setting ***PPPShortHold*** to *-1*, however, it is possible to set this variable in a way in which after termination of the link, a dial-up connection is automatically initiated and the link is reestablished. The current operational status of the interface, **ifOperStatus**, only takes the values *up* or *down*, no longer *dormant* or *blocked*. Configuration of this feature can be made over the MIB table and Setup Tool.

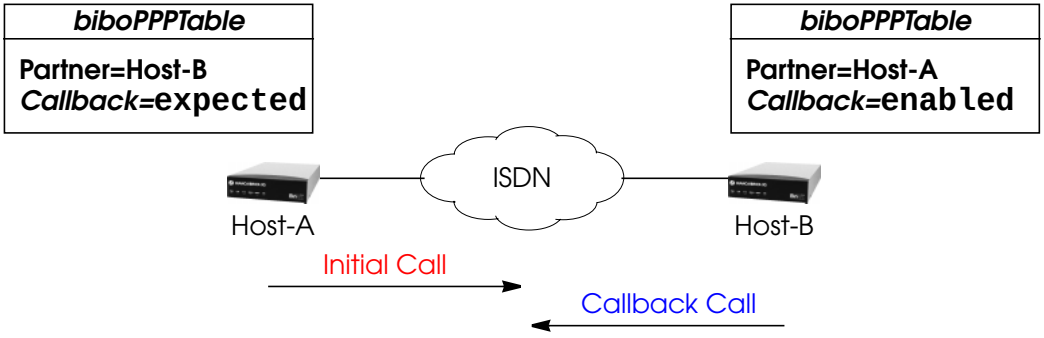
Caution:

This immediate reestablishment of the link should be expressly wished as setting **PPPShorthold** to *-1* can obviously have considerable financial implications.
If you wish to prevent constant reestablishment of a link, make sure to set **PPPShorthold** to a value other than *-1*.

ISDN Callback

ISDN callback operation is supported in both directions on the BinTec router. Using the ***biboPPPCallback*** object ISDN

callback can be configured separately for each PPP partner in **`several`** modes.



`Callback: enabled`

Enabled mode operates as follows:

1. The BinTec router receives an ISDN call from this partner.
2. After authenticating the caller (via Calling Line ID or CHAP/PAP) the BinTec router closes the initial connection and places a new call to the partner.

For the Initial Call to be acknowledged on the receiving host an incoming number entry (***Direction*** = **`incoming`** or **`both`**) must be present for the calling partner in the ***biboDialTable*** and may not contain wildcard characters.

To place the Callback Call an outgoing number (not containing wildcards) must be present for this partner in the ***biboDialTable*** (***Direction*** is either **`outgoing`** or **`both`**). If the Callback Call is not successful the BinTec router waits a preset amount of time and re-attempts callback up to ***biboPPPMaxRetries*** times (by default 5 attempts are made).

Callback: disabled

No Callback is made.

Callback: `expected`

Expected mode operates as follows:

1. The BinTec router places an initial call to the ISDN partner. The partner may be another BinTec router (configured for enabled mode) or another system that supports ISDN callback.
2. The remote partner closes the initial connection and returns the ISDN call; hence callback is "expected" from this partner.

For the Initial Call at least one ISDN number (not containing wildcards) must be present in the ***biboDialTable*** (***Direction*** is either **`outgoing`** or **`both`**). The first number found for this partner (Host-B above) is used to place the call.

For the Callback Call to be accepted by the receiving host an incoming number entry must be present for calling partner in the ***biboDialTable***. This entry may not contain wildcard characters.

Callback: ppp_offered

Setting this value to ppp_offered allows a called peer to call back the calling site if offered by ppp negotiation.

Callback: delayed

The Callback can be delayed for ***biboPPPRetryTime*** seconds by setting Callback to Delayed (5)

Callback: ppp_callback_optional (MS-Callback Termination Option)

When callback is configured on a BinTec router for a Windows 95/98/NT client, normally the procedure is that you dial in to the central-site BinTec router, a win-

dow opens in which you enter the dial number of the terminal from which you are presently calling, the original call is disconnected and callback is initiated, i.e. your headquarters calls you back and bears the charges. It is also possible for the administrator at your head office to configure the telephone number from which you regularly require the callback function, in which case you need only confirm the callback mode without entering your telephone number.

If, however, you want to retain the existing connection to your head office without initiating callback, if you do not know the number of the phone you are calling from or you cannot be called back (if the necessary authentication, dialing code or extension are not available, for example), a feature to terminate the callback function is available.

Configuration

It is possible to give the Windows client the option to demand a callback or to access the head office by the initial connection.
Configure the central-side BinTec router by setting the **biboPPPCallback** variable to *callback_optional* in the **biboPPPTable**. This has the effect that the following options are now available to the Windows client.

Application

Windows 95/98 Clients:

1. For callback numbers to be specified by the caller (user-defined number):
   A window appears in which the caller can select either **OK**, to enter a number to be called back or **CANCEL**, to retain the existing connection.

2. For callback numbers predefined by the central-side administrator (administrator-defined number): A window appears requesting the user to either confirm the callback mode with **OK**, or to retain the existing connection by clicking **CANCEL.**

Windows NT clients:

1. For callback numbers to be specified by the caller:A window appears in which the caller can select either **OK**, to enter a number to be called back or **CANCEL**, to retain the existing connection.
2. For callback numbers predefined by the central-side administrator: In this case, no window appears and the callback termination option can not be availed of.

Layer 1 Protocol

The ***biboPPPLayer1Protocol*** object defines the layer 1 protocol to use for connections to/from this dialup partner. By default data_64k is selected. The list of possible layer 1 settings is shown below .

Layer1Protocol	Comment
data_64k	Default setting.
data_56k	For connections over ISDN lines limited to 56k lbandwidth (e.g., calls to/from North America).
modem	The actual layer 1 connection parameters are negotiated by the calling/receiving modems[a].
modem_profile_1 – modem_profile_8	The incoming/outgoing call to the specified partner uses the modem settings defined in the respectve profile. Refer to the ***mdmProfileTable***.[a]
dovb	Special setting for "Data over Voice Bearer"[b]

Layer1Protocol	Comment
v110_1200 – v110_38400	V.110 bit rate adaptation. Identifies the settings to use (1200 baud, 8, N, 1 through 38400 baud, 8, N, 1) for calls to this partner.

a.Only for products with internal modems (BRICK-XL, V!CAS and XS-Office).
b.Used mainly in N.America to allow data transfers over voice circuits (i.e., the digital call is initially setup using voice signalling).

Auto-Login

The ***biboPPPLoginString*** defines a text string that is used to automatically log into the called system. This variable is only useful in connection with the x75_btx and x75_ppp encapsulations (see Encapsulation: x75_ppp) for automating dialup connections with CompuServe Online Services.

The Login String consists of special characters and alternating expect – send sequences separated by spaces. Blank spaces in strings which are part of the variable ***biboPPPLoginString*** must be preceded by a backslash. The first string detected as not being a special character is assumed to be an expect string. Currently the following special characters are recognized.

Special TAG	Meaning
-d<*number*>	Indicates a pause of <*number*> seconds.
\n	Indicates transmit a carriage return.
\<space>	A backslash should precede a space.

A typical logon string that might be used for logging onto Compuserve directly is shown below:

```
“-d1 \n e: CIS\n ID: 12345,6789/go:pppconnect\n
    word -d1 secret\n PPP”
```

Once the initial connection is established this string would be used to:

Wait 1 second
transmit a carriage return
expect the string: “e:”
transmit “CIS” followed by a carriage return
expect the “ID:” string
transmit “12345,6789/go:pppconnect” and a carriage return
expect the string: “word”
wait 1 second
then transmit “secret” followed by a carriage return
expect the string “PPP”

The Compuserve UserID and Password, shown above (12345,6789 and secret), would have to be changed, of course.

Header Compression

The ***biboPPPVJHeaderComp*** object defines whether Van Jacobson TCP/IP header compression (VJHC) should be used with this partner. For IP capable interfaces VJHeaderComp may be set to either **`enabled`** or **`disabled`**.

If the dialup partner supports header compression this option can be used to help reduce the size of TCP/IP packets and provide improved performance (line efficiency). VJHC is supported acording to RFC 1144.

Compression settings are negotiated at connection time during PP setup. If the called party is does not support VJHC (or if disabled for this partner but the calling party requests it) the link is still established, but without header compression enabled. When negotiated successfully a system message (level info) is generated in the syslogTable, see the subject PPP in the chapter on Syslog Messages).

Layer2Mode

The ***biboPPPLayer2Mode*** object defines the mode to use at layer 2 for connections to this dialup partner. For leased line partners the layer 2 mode set in the ***Type*** field of the ***isdnChTable*** is used.

Layer2Mode is only relevant, if ***biboPPPEncapsulation*** involves a LAPB based protocol which is the case for the following settings:

`x25` **`x25_ppp`** **`x25_ppp_opt`** **`ip_lapb`**
`mpr_lapb` **`x31_bchan`** **`x75_ppp`** **`x75btx_ppp`**
`x25_nosig`

By default ***Layer2Mode*** is set to **`auto`**. This means that the BinTec router will adjust the its layer 2 mode appropriately depending on the direction of the call for this partner. For an incoming call from this partner the BinTec router operates a DCE, when placing a call to this partner the BinTec router operates as DTE.

Setting this object to either **`dte`** or **`dce`** means the BinTec router will always operate as DTE or DCE respectively, regardless of the direction of the call. Also if dte or dce is set an approriate entry in the ***biboDialTable*** must also be present.

if *biboPPPLayer2Mode* =	if *biboDialDirection* =
`dte`	**`both or outgoing`**
dce	both or incoming

PPP Identification

When PAP and/or CHAP authentication is used with the dial-up partner the BinTec router must identify itself with a special string known as the PPP ID.

When authentication is performed with this partner the BinTec router sends the value of the ***biboPPPLocalIdent*** varialbe. If this object is not set, the BinTec router uses the contents of the ***biboAdmLocalPPPIdent*** variable in the ***admin*** tabe.

1.4 ADSL Connection via PPPoE

1.4.1 Introduction

BinTec Communications AG offers the PPP-over-Ethernet protocol to enable networked terminals access to the Internet over the T-DSL connection of the Deutsche Telekom AG.

Why use a BinTec router for T-DSL access?

The use of a Bintec router on a T-DSL connection is of particular benefit when you have one or more of the following requirements:

1. LAN / WAN:

   You want to connect an entire LAN via T-DSL to the Internet and not just one workstation.

   In addition to T-DSL Internet access, you also need other WAN connections (e.g. Modem dial-in, ISDN-Intranet connection etc.).

2. Security:

This point relates only to BinTec routers that have two Ethernet interfaces. The security advantages listed below that can be enjoyed by users of products with two Ethernet interfaces can not be shared by users of products with just one Ethernet interface. Indeed, the use of PPPoE over one Ethernet interface presents several disadvantages you should be aware of, see "Two Ethernet interfaces or one?", page 60.

The customer´s network should be protected from unauthorised access from the Internet.

The Internet should be made strictly inaccessable to unauthorised individuals from the customer´s network.

3. Accounting:

Online time: the number of connections and transmission volumes for IP traffic should be recorded in detail.

Superfluous load connections (such as broadcasts) should be prevented.

This last point relates only to BinTec routers that have two Ethernet interfaces. The use of PPPoE over one Ethernet interface presents several disadvantages you should be aware of, see 1.4.3: "Using T-DSL with BinTec routers with one Ethernet Interface".

4. Platforms:

Workstations running operating systems for which the PPPoE protocol is not available should be connected (e.g. OS/2, Linux, Windows 3.x etc.)

5. Backup:

A high degree of availability should be guaranteed; should the T-DSL access fail, an alternative path should be activated.

6. Services:

   In addition to T-DSL Internet access, other communications services are required network-wide (e.g. Fax, Eurofiletransfer etc.).

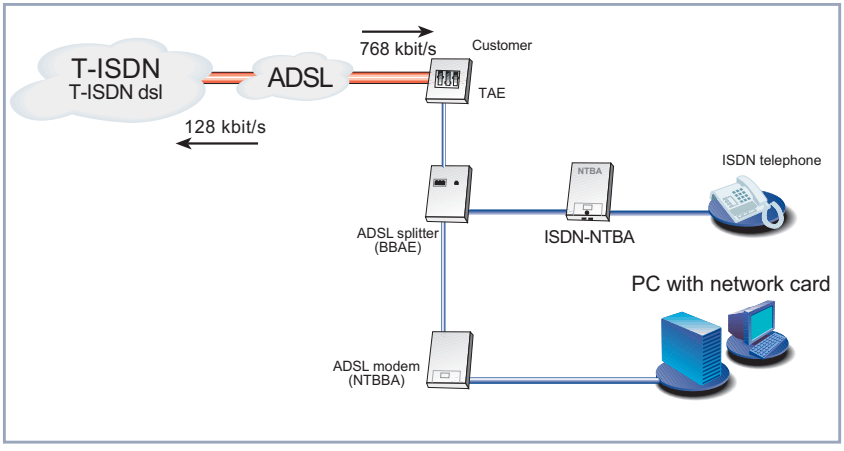
7. Configuration:

   Access should be configured and administrated from a central site or by an external service provider.

Furthermore, you would like to continue to benefit from the full range of functions available with your BinTec multiprotocol router.

A brief introduction to T-DSL

With T-DSL Deutsche Telekom AG is offering high-speed Internet access. The underlying technology is ADSL. Large amounts of data can be asymetrically transmitted over conventional, copper telephone lines by ADSL (Asymmetric Digital Subscriber Line). The T-DSL packet consists of an ISDN connection and a data line with a bandwidth of up to 768 kbps from the Internet Service Provider to the customer (downstream) and 128 kbps in the opposite direction (upstream). This bandwidth capacity provides downstream Internet services availability at speeds of up to twelve times faster than with ISDN.
The T-DSL connection (without a BinTec router) looks like this:



Two Ethernet interfaces or one?

To be able to use ADSL (Asymmetric Digital Subscriber Line) with a BinTec router, you must configure a PPP-over-Ethernet interface over the LAN interface. This is done by connecting your BinTec router to T-DSL, which is the ADSL connection of Deutsche Telekom AG. It is possible to avail of the services of ADSL by connecting a BinTec router to two Ethernet interfaces or to just one, depending on how your router is equipped.

> The use of PPPoE over one Ethernet interface presents several disadvantages you should be aware of, see 1.4.3: "Using T-DSL with BinTec routers with one Ethernet Interface".

At the time of writing, BinTec Communications AG has three routers fitted or with the potential to be fitted with two Ethernet interfaces: XM-PPPoE, XM2 and XL2 (this list will be quickly outdated as new modular devices supplement the product range). All other BinTec routers addressed in this release (BinGO!, XS2, XS-Office, XMP) are

only capable of connecting to the ADSL modem and to the LAN using just the one Ethernet interface.

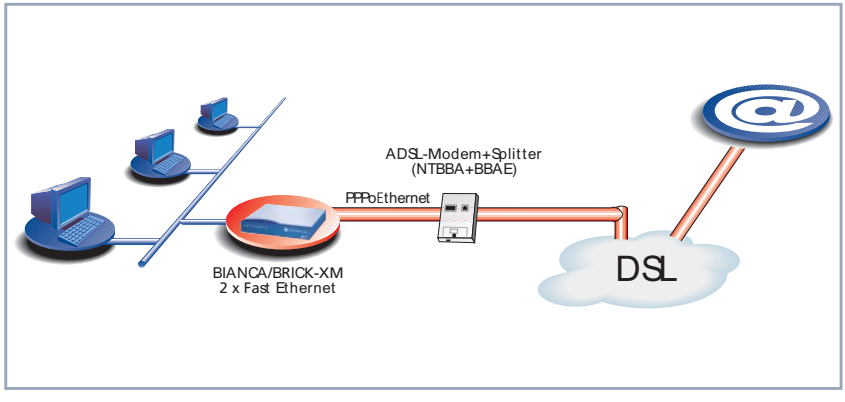
1.4.2 Using T-DSL with BinTec routers with two Ethernet Interfaces

BinTec Communications AG recommends using a BinTec router with 2 Ethernet interfaces for your ADSL connection: one back to the LAN, the other to the ADSL connection. When using 2 Ethernet interfaces, all the advantages mentioned can be enjoyed without exception or restriction.

Scenario: Internet access for several PCs

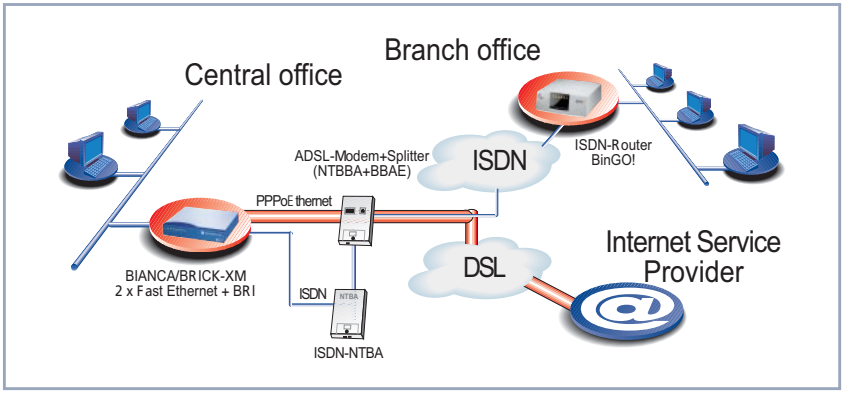
In order to give your Local Area Network cheap and fast access to the Internet, your BinTec router is connected to the Ethernet between the PCs and the ADSL modem:

If you receive a special cable from Deutsche Telekom AG for connecting the ADSL modem, please use only this cable.



Scenario: Connecting to a second location

In order to give your Local Area Network cheap and fast access to the Internet, as well as to a second remote location, your BinTec router is connected to the Ethernet between the PCs and the NTBBA (ADSL network termination):



Scenario: Connecting with fax servers

In order to give your Local Area Network cheap and fast access to the Internet, as well as simultaneous use of the ISDN connection for professional fax services, the BinTec router is fitted with two Fast Ethernet modules and a 2XBRI-module:

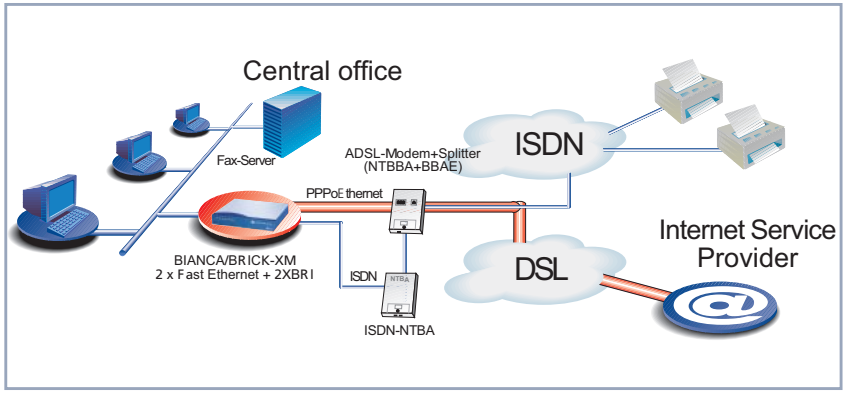
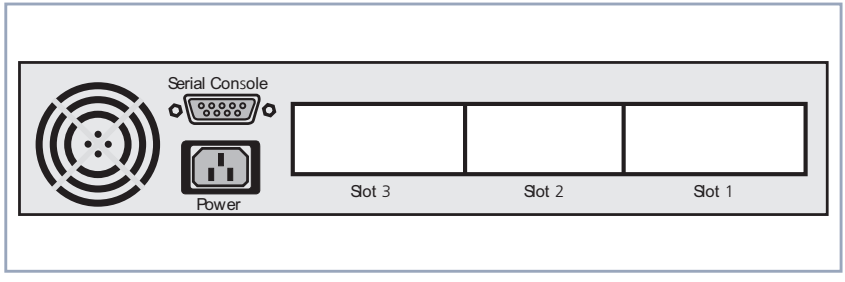


Figure A-1: Connecting with fax servers

Hardware connections on the XM-PPPoE to T-DSL

When connecting a LAN, WAN or ADSL connection to the XM-PPPoE for example, the following slot assignments should be observed:



Slot	Module/Function
Slot 1	Ethernet (to the LAN)
Slot 2	Ethernet (to the ADSL)
Slot 3	S_0 or another module (optional)

Configuration

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

```
BRICK Setup Tool                                BinTec Communications AG
                                                                 MyBrick

Licences                  System

Slot1:                    CM-BNC/TP, Ethernet
Slot2:                    CM-BNC/TP, Ethernet

Slot3:                    CM-1BRI, ISDN S0

WAN Partner
IP  PPP IPX     X.25      VPN

Configuration Management
Monitoring and Debugging
Exit

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

Configuring IP Addresses

- Go to ***SLOT 1*** (Ethernet)

.

```
BRICK Setup Tool                              BinTec Communications AG
[SLOT 1 ETHERNET]: Configure Ethernet Interface                MyBrick


            IP-Configuration
                local IP-Number      192.168.1.254
                local Netmask        255.255.255.0
                Encapsulation        Ethernet II

            IPX-Configuration
                local IPX-NetNumber  0
                Encapsulation        none

            Bridging                 disabled

            Advanced Settings>

                SAVE                 CANCEL

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

Field	Meaning
local IP-Number	Enter the LAN IP address of your BinTec router here. This address should be the default gateway for the hosts in your LAN.
local Netmask	Enter the netmask for your LAN here.

General PPP Settings

Here you must configure an interface over which PPP-over-Ethernet should run. All other settings can be left as they are.

- From the main menu, go to ***PPP***.

```
BRICK Setup Tool                                BinTec Communications AG
[PPP]: PPP Profile Configuration                                 MyBrick


Authentication Protocol                 CHAP + PAP + MS-CHAP
Radius Server Authentication            inband

PPP Link Quality Monitoring             no

PPPoE Ethernet Interface                en2


                SAVE                            CANCEL

Use <Space> to select
```

The following field is relevant:

Field	Meaning
PPPoE Ethernet Interface	This field defines the interface over which PPP-over-Ethernet runs.

WAN Partner Settings

The configuration of a PPP-over-Ethernet partner is exactly the same as the configuration of any other WAN partner.

- Go to ***WAN PARTNER*** ➧ ***ADD***.

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

Partner Name                          t-online

Encapsulation                         PPP
Compression                           none
Encryption                            none
Calling Line Identification           no

PPP>
Advanced Settings>
WAN Numbers>

IP>
IPX>
Bridge>

                 SAVE                      CANCEL

Enter string, max length = 25 chars
```

These are the relevant fields:

Field	Meaning
Partner Name	The name assigned to this PPP-over-Ethernet partner.
Encapsulation	Defines how data packets are encapsulated for transmission to the WAN partner. For the purpose of PPP-over-Ethernet, only *PPP* should be selected.

PPP Submenu Settings

- Go to ***PPP***.

```
BRICK Setup Tool                              BinTec Communications AG
[WAN][ADD][PPP]: PPP Settings (t-online)                      MyRouter


Authentication              PAP
Partner PPP ID
Local PPP ID                000460004256091169386#0001@t-online.de
PPP Password                1234567

Keepalives                  on
Link Quality Monitoring     off


                    OK                        CANCEL


Use <Space> to select
```

These are the relevant fields:

Field	Meaning
Authentication	PAP. The default value CHAP + PAP must be changed.
Partner PPP ID	WAN-Partners ID. Leave blank.
Local PPP ID	Your T-Online User-ID. This is how the ID is constructed: <Code><T-Online-Nr.>#<Other user-Nr.>@t-online.de. Code = a 12-digit connection code (e.g.: ***000460004256***) T-Online-Nr. = Telephone number (e.g.: ***091169386***) Other user-Nr. = a four-digit other user number (e.g.: ***0001***)
PPP Password	Your T-Online password.
Keepalives	Activates keepalive packets.

- Set **Keepalives** to *on*.

When the keepalive function is active, the status of the interface is checked. If the connection to the Provider

fails, this feature quickly recognises and signals the altered status of the interface.

Advanced Settings

- Return to ***ADVANCED SETTINGS***.

You can define the Layer 1 Protocol of the ISDN B channel the BinTec router should use for connections to the WAN partner. The protocol for ISDN data connections, standard value for the B-channel, is preconfigured. For PPP-over-Ethernet, this setting must be changed.

```
BRICK Setup Tool                                BinTec Communications AG
[WAN][ADD][ADVANCED]: Advanced Settings (t-online)            MyRouter


Callback                                        no
Static Short Hold (sec)                         20
Idle for Dynamic Short Hold (%)                 0
Delay after Connection Failure (sec)            300

Extended Interface Settings (optional)<

Channel-Bundling                                no

Layer 1 Protocol                                PPP over Ethernet (PPPoE)



                      OK                        CANCEL

Use <Space> to select
```

- In the field **Layer 1 Protocol**, select *PPP over Ethernet (PPPoE)*.

IP Settings

- Return to ***IP***.

```
BRICK Setup Tool                              BinTec Communications AG
[WAN][ADD][IP]: IP Confuguration (t-online)                    MyRouter

IP Transit Network              dynamic client

local IP Address


Advanced Settings>
                SAVE                CANCEL
```

The following field is relevant here:

Field	Meaning
IP Transit Network	Defines whether the BinTec router is to establish a Transit Network to the WAN partner. IP address is dynamically assigned if *dynamic client* is selected.

- Set **IP Transit Network** to *dynamic client*.
- The **local IP Address** field remains blank.

General IP Settings

Configuring the default route

- Go to ***IP*** ➧ ***ROUTING*** ➧ ***ADD***.

```
BRICK Setup Tool                                BinTec Communications AG
[IP][ROUTING][ADD]: IP Routing                                  MyRouter


Route Type                    Default route
Network                       WAN without transit network

Partner / Interface           t-online

Metric                        1

                SAVE                    CANCEL

Use <Space> to select
```

The following field is relevant here:

Field	Meaning
Partner / Interface	Ihr PPPoE Partner.

- In the **Route Type** field, select *Default route*.
- In the field **Partner / Interface**, select *PPPoEPartner*, e. g. ***t-online***.

Activate Network Address Translation (NAT)

This results in the following:

1. your network can not be accessed from the Internet (as long as no session profiles are configured),
2. The source address of connections to the Internet only appears as the one dynamically assigned IP address.

- Go to ***IP*** ➧ ***NETWORK ADDRESS TRANSLATION***.
- Select the WAN interface you want to activate NAT for, e. g. ***t-online***.

```
BRICK Setup Tool                              BinTec Communications AG
[IP][NAT][CONFIG]: NAT Configuration (t-online)               MyRouter

Network Address Translation              on

Configuration for sessions requested from outside
Service      Destination      Source Dep.  Dest. Dep.   Port Remap


    ADD          DELETE       SAVE         CANCEL
```

The following field is relevant here:

Field	Meaning
Network Address Translation	Here you can activate Network Address Translation (NAT) for your WAN partner. Thereby, you conceal your entire LAN behind the one official IP address.

- Set **Network Address Translation** to *on*.

1.4.3 Using T-DSL with BinTec routers with one Ethernet Interface

To be able to use ADSL (Asymmetric Digital Subscriber Line) with a BinTec router, you must configure a PPP-over-Ethernet interface over the LAN interface. This is done by connecting your BinTec router to T-DSL, which is the ADSL connection of Deutsche Telekom AG. It is possible to avail of the services of ADSL by connecting a BinTec router to two Ethernet interfaces or to just one, depending on how your router is equipped.

Security risks and restrictions

BinTec Communications AG recommends using a BinTec router with 2 Ethernet interfaces for your ADSL connection. Due to customer demand, however, PPP over Ethernet is also being made available with this release for BinTec routers with just one Ethernet interface.
When using PPP over Ethernet with one Ethernet interface, you should be aware of the following security risks and other disadvantages.

The following restrictions and security risks exist when the BinTec router connection to T-DSL is only over one Ethernet interface:

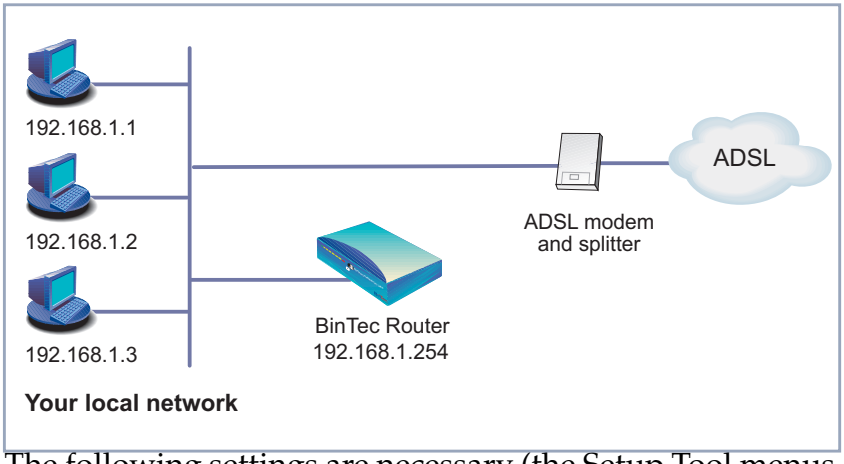
- If PPP-over-Ethernet is operated with only one Ethernet interface, there is a risk of unauthorized accesses from the Internet to the local BinTec router LAN. Such unauthorized accesses can originate from the first node of the Internet.
- Users of the local network can configure a PPP-over-Ethernet client on their PC and use the Internet unnoticed by the BinTec router.
- Broadcasts in the local LAN are always forwarded by the ADSL modem (NTBBA) to the PTT exchange and are not rejected until the exchange. This means that the maximum bandwidth of 128 kbps upstream to the PTT may not be fully available.

Scenario: ADSL with BinTec routers with one Ethernet Interface

The following scenario is used to describe the necessary configuration steps: The LAN interface of the BinTec router

is connected to your hub, and to the ADSL modem (NTBA) of Deutsche Telekom AG.

> If you receive a special cable from Deutsche Telekom AG for connecting the ADSL modem, please use only this cable.



The following settings are necessary (the Setup Tool menus concerned are described elsewhere):

- Go to ***PPP***.
- Select **PPPoE Ethernet Interface**: *en1*.
- Press **SAVE**.
- Go to ***WAN PARTNER*** ➧ **ADD**.
- Enter your **Partner Name**: e.g. *t-online*.
- Select **Encapsulation**: *PPP*.
- Go to ***WAN PARTNER*** ➧ **ADD** ➧ ***PPP***.
- Enter **Local PPP ID** (= your user name): e.g. *00046000425609116938 6#0001@t-online.de*.

The T-Online user name comprises the following elements:
<user account><T-Online number>#<co-user number>@t-online.de
The user account is a 12-digit number, in this case: *000460004256.*
The T-Online number is the extension number, in this case: *091169386.*
The co-user number is a 4-digit number, in this case: *0001.*
The T-Online number and the co-user number must be separated by # if the T-Online number has less than 12 digits.

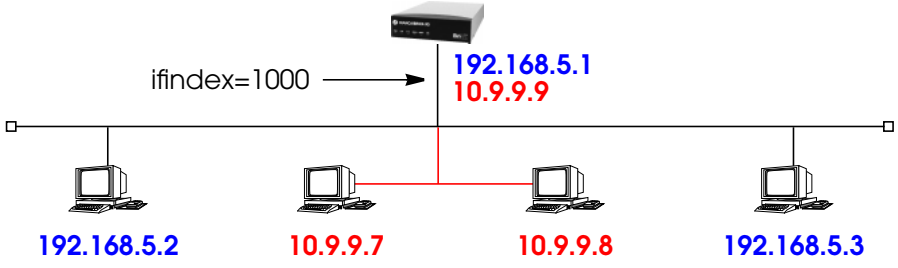
- Enter **PPP Password** (= your T-Online password).
- Select **Keepalives**: *on.*
- Confirm with **OK**.
- Go to ***WAN Partner*** ➧ **ADD** ➧ ***Advanced Settings***.
- Select **Layer 1 Protocol**: *PPP over Ethernet (PPPoE).*
- Confirm with **OK**.
- Go to ***WAN Partner*** ➧ **ADD** ➧ ***IP***.
- Select **IP Transit Network**: *dynamic client.*
- Press **SAVE**.
- Go to ***IP*** ➧ ***Routing*** ➧ **ADD**
- Select **Route Type**: *Default route.*
- Select **Network**: *WAN without transit network.*
- Select **Partner / Interface**: e.g. *t-online.*
- Enter **Metric**: e.g. *1*.
- Press **SAVE**.
- Go to ***IP*** ➧ ***Network Address Translation***
- Select the PPPoE interface, e.g. **t-online**, and confirm with **Return**.
- Select **Network Address Translation**: *on.*

Press **SAVE**.

1.5 Dual IP Address Interfaces

Normally each (physical) BinTec router ethernet interface is assigned a single IP address. This address can be seen in the BinTec router's ***ipAddrTable*** which lists the current IP address for all BinTec router interfaces.

A second IP address may be assigned to an ethernet interface by creating a direct route in the ***ipRouteTable*** that points to the interface.



The ethernet interface for the BinTec router in the diagram (assumed to already be assigned 192.168.5.1) could be assigned a second address by adding the following IP route.

```
mybrick: system > ipRouteIfIndex=1000 ipRouteDest=10.9.9.0 ipRouteNextHop=10.9.9.9

01: ipRouteDest.10.9.9.0.2( rw): :        10.9.9.0
01: ipRouteNextHop.10.9.9.0.2( rw): :     10.9.9.9
01: ipRouteIfIndex.10.9.9.0.2( rw): :     1000

mybrick: ipRouteTable> ipRouteTable

inx Dest(*rw)          IfIndex(rw)       Metric1(rw)       Metric2(rw)
    Metric3(rw)        Metric4(rw)       NextHop(rw)       Type(-rw)
    Proto(ro)          Age(rw)           Mask(rw)          Metric5(rw)
    Info(ro)

01  10.0.0.0           1000              0                 -1
    -1                 -1                10.9.9.9          direct
    netmgmt            5                 255.0.0.0         -1
    .0.0

mybrick : ipRouteTable >
```

Both IP addresses will appear in the ***ipAddrTable***.

```
mybrick: ipRouteTable > ipAddrTable

inx Addr(*ro)          IfIndex(ro)       NetMask(ro)       BcastAddr(ro)
    ReasmMaxSize(ro)

00  192.168.5.1        1000              255.255.255.0     1
    65535
01  10.9.9.9           1000              255.0.0.0         1
    65535

mybrick : ipAddrTable >
```

The BinTec router can now route between the two networks.

1.6 IP Routing on the BinTec router

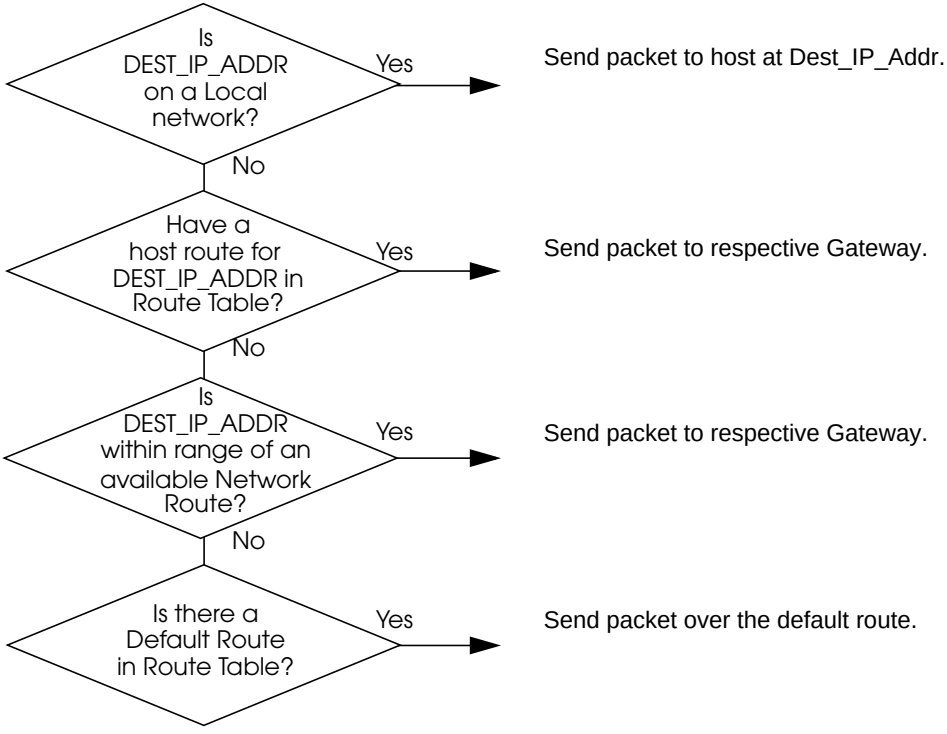
The BinTec router's IP routing table is contained in the ***ipRouteTable***. It contains, among other information, a list of destination addresses (host or network) and gateway addresses to be used when routing IP packets to those destinations. When the routing table is kept current, the BinTec router is prepared to make intelligent decisions as to where to route incoming packets.

Before any IP packets can be routed the BinTec router first determine:

1. The Destination IP Address (DEST_IP_ADDR) from the IP packet.
2. The Destination Network (DEST_ADDR) from DEST_IP_ADDR.
3. Whether a default route exists.

The BinTec router can then decide which interface to route the packet over. To make this decision the BinTec router uses a rather complicated internal routing algorithm. The general routing algorithm might proceed as shown below.

Routing involves doing this for each packet that arriveing packet.



1.7 Extended IP Routing

Most routing decisions are based solely on a packet's destination address, Extended IP routing allows you to route IP traffic based on additional information. Extended routes are configured in the ***ipExtRtTable***. Each extended route table entry defines a separate route which can be separately or jointly based on:

- Contents of the IP packet header.
- The source interface the packet arrived on.

- The current state of a BinTec router interface (normally a dialup interface).

Table Field		Global	Meaning
Contents of IP Header.	***Protocol***	dont_verify	Protocol field of IP header
	SrcIfIndex	-0	The interface packet is routed from.
	SrcAddr	0.0.0.0	Source Address field of IP header.
	SrcMask	0.0.0.0	Used with SrcAddr.
	SrcPort	-1	Source Port field of IP header
	SrcPortRange	-1	If not = -1 last number of range of ports, starting from SrcPort.
	DstAddr	0.0.0.0	Destination Addr field of IP header
	DstAddrMask	0.0.0.0	Used together with DstAddr.
	DstPort	-1	Destination Port field of IP header.
	DstPortRange	-1	Used together with DstPort field.
	Tos	0	Type of Service field of IP header.
	TosMask	0	Type of Service field of IP header.
BinTec router Interface	***DstIfMode***	-	The state of the DstIfIndex.
	DstIfIndex	-0	The interface to route packet to.

1.7.1 Route Priority

When routing IP packets, the BinTec router always checks for extended routes first. If the ***ipExtRtTable*** is empty, or a matching entry is not found, the ***ipRouteTable*** is consulted.

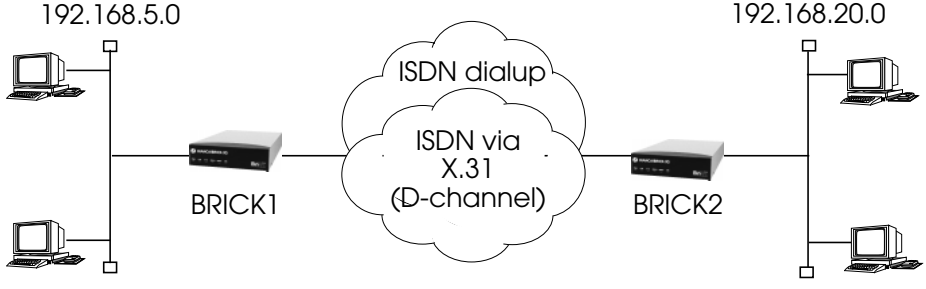
1. First check ***ipExtRtTable***; if a route is found, route packet, otherwise
2. Check ***ipRouteTable***.

NOTE:

If more that one route is found in a routing table, the route with the lowest metric value specified in the ***Metric1*** field is used. If multiple routes are found with the same metric, it is not possible to determine which route will be used.
The ***Metric2***, ***Metric3***, ***Metric4***, and ***Metric5*** fields are not used.

1.7.2 Configuring Extended Routes

For example the two LANs shown below could be connected via an ISDN basic rate interface. For telnet sessions we might want to take advantage of volume-based charging of X.31 (X.25 in the D-channel) and avoid the much higher costs for ISDN dialup connections. All other IP traffic could continue to use the dialup ISDN link.



Presetting

1. Each BinTec router needs to be configured to allow for normal routing via our ISDN dialup link (dialup1, ifindex=10001).
2. Next, we'll need to create an MPX25 (mpx1, ifindex=20001) interface to allow IP traffic to be routed over X.25.

Configuration

Since IP is already routed between the LANs using our dialup interface, we only need to filter out the telnet traffic. This can be done using the *Protocol*, *SrcPort*, and *DstPort* variables in the ***ipExtRtTable***.

Step 1

For BRICK1 the extended IP routes would be added as follows. The first route is for IP packets destined for hosts on network 192.168.20.0 with a source IP port of 23. This is for outgoing telnet sessions.

The second route is for IP packets destined for hosts on the remote network, with a destination IP port of 23. This is for the incoming telnet sessions.

```
brick1: system > ipExtRtTable

inx Protocol(*rw)     SrcIfIndex(rw)       SrcAddr(rw)      SrcMask(rw)
    SrcPort(rw)       SrcPortRange(rw)     DstAddr(rw)      DstMask(rw)
    DstPort(rw)       DstPortRange(rw)     Tos(rw)          TosMask(rw)
    DstIfMode(rw)     DstIfIndex(rw)       NextHop(rw)      Type(-rw)
    Metric1(rw)       Metric2(rw)          Metric3(rw)      Metric4(rw)
    Metric5(rw)       Proto(rw)            Age(rw)

brick1: ipExtRtTable> Protocol=tcp SrcPort=23 DstAddr=192.168.20.0 DstIfIndex=20001

brick1: ipExtRtTable> Protocol=tcp DstPort=23 DstAddr=192.168.20.0 DstIfIndex=20001

inxProtocol(*rw)  SrcIfIndex(rw)       SrcAddr(rw)      SrcMask(rw)
    SrcPort(rw)       SrcPortRange(rw)     DstAddr(rw)      DstMask(rw)
    DstPort(rw)       DstPortRange(rw)     Tos(rw)          TosMask(rw)
    DstIfMode(rw)     DstIfIndex(rw)       NextHop(rw)      Type(-rw)
    Metric1(rw)       Metric2(rw)          Metric3(rw)      Metric4(rw)
    Metric5(rw)       Proto(rw)            Age(rw)

00  tcp               0                    0.0.0.0          0.0.0.0
    23                -1                   192.168.20.0     255.255.255.0
    -1                -1                   0                0
    dialup_wait       20001                0.0.0.0          indirect
    0                 0                    0                0
    0                 netmgmt              0 00:20:25.00

 01 tcp               0                    0.0.0.0          0.0.0.0
    -1                -1                   192.168.20.0     255.255.255.0
    23                -1                   0                0
    dialup_wait       20001                0.0.0.0          indirect
    0                 0                    0                0
    0                 netmgmt              0 00:20:26.00

brick1 : ipExtRtTable >
```

Step 2 The same extended routes would also be added to BRICK2. The ifIndex for our MPX25 partner (BRICK1) is assumed to

be 20003 on BRICK2. Again the first route is for the outgoing telnet sessions, the second is for the incoming sessions.

```
brick2 : ipExtRtTable > Protocol=tcp SrcPort=23 DstAddr=192.168.5.0 DstIfIndex=20003

brick2 : ipExtRtTable > Protocol=tcp DstPort=23 DstAddr=192.168.5.0 DstIfIndex=20003

brick2 : ipExtRtTable > ipExtRtTable

inxProtocol(*rw)  SrcIfIndex(rw)       SrcAddr(rw)       SrcMask(rw)
    SrcPort(rw)       SrcPortRange(rw)  DstAddr(rw)       DstMask(rw)
    DstPort(rw)       DstPortRange(rw)  Tos(rw)           TosMask(rw)
    DstIfMode(rw)     DstIfIndex(rw)    NextHop(rw)       Type(-rw)
    Metric1(rw)       Metric2(rw)       Metric3(rw)       Metric4(rw)
    Metric5(rw)       Proto(rw)         Age(rw)

00 tcp                0                 0.0.0.0           0.0.0.0
    23                -1                192.168.5.0       255.255.255.0
    -1                -1                0                 0
    dialup_wait       20003             0.0.0.0           indirect
    0                 0                 0                 0
    0                 netmgmt           0 00:20:25.00

 01 tcp               0                 0.0.0.0           0.0.0.0
    -1                -1                192.168.5.0       255.255.255.0
    23                -1                0                 0
    dialup_wait       20003             0.0.0.0           indirect
    0                 0                 0                 0
    0                 netmgmt           0 00:20:26.00

brick2 : ipExtRtTable > ipExtRtTable
```

Additional Options

A range of ports can also be specified using the ***SrcPort*** and ***SrcPortRange*** variables together. ***SrcPort*** specifies the first port number in the range, ***SrcPortRange*** defines the last port in the range. Both ports are included in the range. For example, the extended routes in the above examples could have looked like this:

Protocol=tcp SrcPort=21 SrcPortRange=23
DstAddr=192.168.5.0 DstIfIndex=20003

Protocol=tcp DstPort=21 DstPortRange=23 DstAddr=192.168.5.0 DstIfIndex=20003

allowing ftp and telnet sessions (IP ports 21 and 23) to be routed over a specific interface.

1.8 BOOTP and DHCP

BootP and DHCP (Dynamic Host Configuration Protocol) are two methods that are commonly used by a host to retrieve important configuration information from a remote host on the LAN. For diskless workstations this might be it's IP address and netmask but could also include other information such as a nameserver's address to use. The BinTec router supports both protocols.

BootP The BinTec router can operate as a **BootP Relay Agent**. This means BootP requests received over the BinTec router's interfaces are forwarded to the BootP Server (***biboAdmBootpRelayServer***) if one exists. To configure a Relay Agent see BootP Relay Agent Settings.

NOTE:

The BinTec router can retrieve it's own configuration information at boot time via a BootP server. See the section BOOT Options on the BinTec router, in System Administration on the BinTec router.

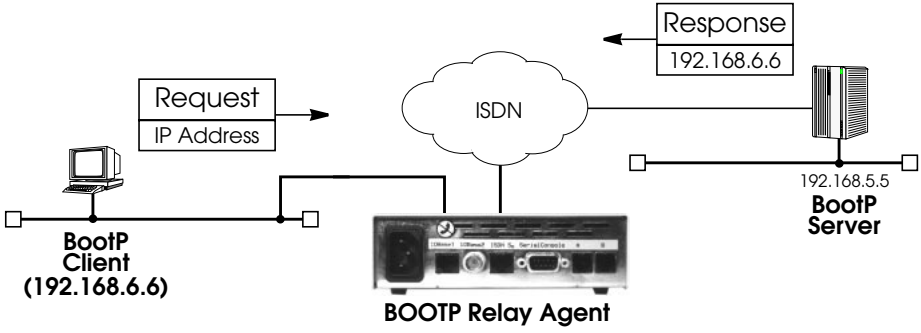
DHCP The BinTec router can also operate as a **DHCP Server.** DHCP is intended to make maintenance of remote and/or diskless workstations easier. It is also commonly used on Windows based systems. The major benefit of DHCP is that it gives the network manager the ability to manage a limited number of IP addresses among several hosts. The DHCP server on the BinTec router

provides the extra benefits of accessibility. To configure a DHCP Server see DHCP Server Setting.

NOTE: BootP and DHCP use the same message format. When a DHCP message is received on the BinTec router but the DHCP server is NOT configured the DHCP request is automatically forwarded to the BootP server if one is configured.
Although BootP and DHCP use the same message format, they are not completely compatible. Clients that support BootP can't be configured via the BinTec router's DHCP server.

1.8.1 BootP Relay Agent Settings

The BinTec router forwards all BootP requests received over it's LAN interfaces to the host defined in the ***biboAdmBootpRelayServer*** field of the ***admin*** table.



BootP Relaying for this example would be configured as follows:

Step 1 First, configure the IP address for the BootP Server in the admin table.

```
mybrick: > admin
...
biboAdmBootpRelayServer(rw):       0.0.0.0
...
mybrick: admin> biboAdmBootpRelayServer=192.168.5.5
biboAdmBootpRelayServer(rw):       192.168.5.5

mybrick: admin>
```

Step 2 Before the BinTec router can begin forwarding requests to this host a partner interface must be created in the ***biboPPPTable***. Below we create a standard dial-up interface for the server.

```
mybrick: admin> biboPPPType=isdn_dialup

05: biboPPPType.1.5(rw):      isdn_dialup

mybrick : biboPPPTable > biboPPPTable

inx IfIndex(ro)              Type(*rw)          Encapsulation(-rw)
    Keepalive(rw)            Timeout(rw)        Compression(rw)
    Authentication(rw)       AuthIdent(rw)      AuthSecret(rw)
    IpAddress(rw)            RetryTime(rw)      BlockTime(rw)
    MaxRetries(rw)           ShortHold(rw)      InitConn(rw)
    MaxConn(rw)              MinConn(rw)        Callback(rw)
    Layer1Protocol(rw)       LoginString(rw)

05  10006                    isdn_dialup        ppp
    off                      3000               none
    none
    static                   4                  300
    5                        20                 1
    1                        1                  disabled
    data_64k

mybrick: biboPPPTable >
```

Step 3 Don't forget to add the telephone number to the BinTec router's Dial table.

```
mybrick: biboPPPTable > biboDialIfIndex=10006 biboDialNumber=555

06: biboDialIfIndex.10006.6(rw):      10006
06: biboDialNumber.10006.6(rw):       "555"

mybrick : biboDialTable >
```

For information on other options when creating partner interfaces, refer to section Creating a DialUp IP Interface.

1.8.2 *DHCP* Server Setting

DHCP on the BinTec router consists of the ***ipDhcpTable*** and ***ipDhcpInUseTable***.

ipDhcpTable	Used to define a pool of addresses the BinTec router will use when assigning IP addresses to DHCP clients. Each entry in the table defines a range of addresses to use for requests received on the respective interface.
ipDhcpInUseTable	Displays which addresses are in use (by host's MAC address) as well the address' expire time.

In Setup Tool the Configuration of DHCP can be entered in the **DHCP** submenu of the **IP** menu.

A range of 13 addresses for the BinTec router's first LAN interface could be configured as follows:

```
mybrick: ipDhcpTable> IfIndex=1000 First=192.168.5.80 Range=9
...
mybrick: ipDhcpTable> IfIndex=1000 First=192.168.5.90 Range=4
...
mybrick: ipDhcpTable> ipDhcpTable

inx IfIndex(*rw)          State(-rw)          First(rw)          Range (rw)
    Lease(rw)             Phys(ro)            NodeType(rw)       Gateway(rw)

00  1000                  on                  192.168.5.80       9
    15                    0:0:0:0:0:0         bnode

01  1000                  on                  192.168.5.90       4
    15                    0:0:0:0:0:0

mybrick : ipDhcpTable >
```

When an address is assigned, the BinTec router keeps track of the address' availability in the ***ipDhcpInUseTable.***

```
mybrick: ipDhcpTable> ipDhcpInUseTable

inx Address(*ro)          Phys(ro)            Expires(ro)

00  192.168.5.80          8:0:20:19:ef:eb     30/06/97 17:33:21
01  192.168.5.81          8:0:20:a3:b3f:9     30/06/97 17:36:59
02  192.168.5.83          8:0:20:12:4f:9a     30/06/97 18:06:19

mybrick : ipDhcpInUseTable >
```

Note that 192.168.5.82 wasn't assigned above. The BinTec router verifies an address is available via ping: ICMP requests/reply before assigning an address from the DHCP tables. Since a reboot results in the loss of the audit trail of assigned addresses this is done to avoid duplicate assignments.

According to DHCP, the server may provide clients with additional information such as netmasks, nameserver and

other addresses, etc. The table shown below, lists the types of information the BinTec router currently supports.

DHCP Request Tag	Retrieved from:
IP_ADDRESS	***ipDhcpTable*** (amd ***ipDhcpInUseTable***) (the next available address, see above)
SUBNET_MASK	***ipRouteTable***
GATEWAY	***ipRouteTable*** IP address of the interface the request was received on.
BROADCAST_ADDR	***ipRouteTable*** (using default route: ***Dest*** | ~ ***Mask*** = broadcast address)
TIME_SERVER	***biboAdmTimeServer***
NAME_SERVER	***biboAdmNameServer, biboAdmNameServ2,*** ***biboAdmWINS1***or ***biboAdmWIN2***, see below.
DOMAIN_NAME	***biboAdmDomainName***
LOG_SERVER	***biboAdmLogHostTable***
HOST_NAME	***ipDhcpTable*** The client's IP address (IP_ADDRESS) is sent as a text string.

The variables ***biboAdmNameServer, biboAdmNameServ2, biboAdmWNS1*** and ***biboAdmWNS2*** can also be configured in Setup Tool in the **Static** submenu of the **IP** menu. Another way to get the values for these variables is to receive them via PPP like described below.

DNS and WINS (NBNS) Relay

Client messages that include requests for the domain name server or the NetBios server's address are handled as follows.

1. If ***biboAdmNameServer/biboAdmWNS1*** is set (≠ 0.0.0.0) send IP address, otherwise

2. If ***biboAdmNameServ2/biboAdmWNS2*** is set (≠ 0.0.0.0) send IP address, otherwise
3. Send BinTec router's IP address as NAME_SERVER and
   attempt to resolve the name server's address dynamically (see below) upon reception of first name resolution request.

NetBIOS Node Type by DHCP

The definition of the NetBIOS node type can be set directly in the registry under Windows 95/98/NT or, however, configuration (via DHCP) on the BinTec router is possible. The **ipDhcpNodeType** variable in the **ipDhcpTable** enables a NetBIOS node type to be set for one IP address pool, making it applicable for all of the DHCP clients.

Methods of name resolution

Basically, the node type defines the way in which Windows has names resolved into their corresponding IP addresses. Each node type contains various methods of name resolution. The following are the methods of name resolution contained in the different node types.

- Broadcast resolution
  This is a means within a LAN by which the owner of a NetBIOS name is requested directly for his IP address.
- WINS (Windows Internet Name Service)
  A server is configured as a NetBIOS databank containing lists of NetBIOS names of clients registered with the server.
- LMHOSTS
  This is a file on the Windows client which contains lists of NetBIOS names and their corresponding IP addresses.

NetBIOS node types

The following node types employ some or all of the above means of name resolution, though arranging and employing them in different sequences. The most appropriate node types for name resolution by a WAN partner are either the P or M-Nodes; default is none or **not specified** in SetupTool.

- B-Node (**Broadcast Node**)

1. Broadcast resolution
2. LMHOSTS or Domain Name Service (DNS) if configured

- P-Node (**Point-to-Point Node**)

1. WINS
2. LMHOSTS or Domain Name Service (DNS) if configured

- M-Node (**Mixed Node**)

1. Broadcast resolution
2. WINS
3. LMHOSTS or Domain Name Service (DNS) if configured

- H-Node (**Hybrid Node**)

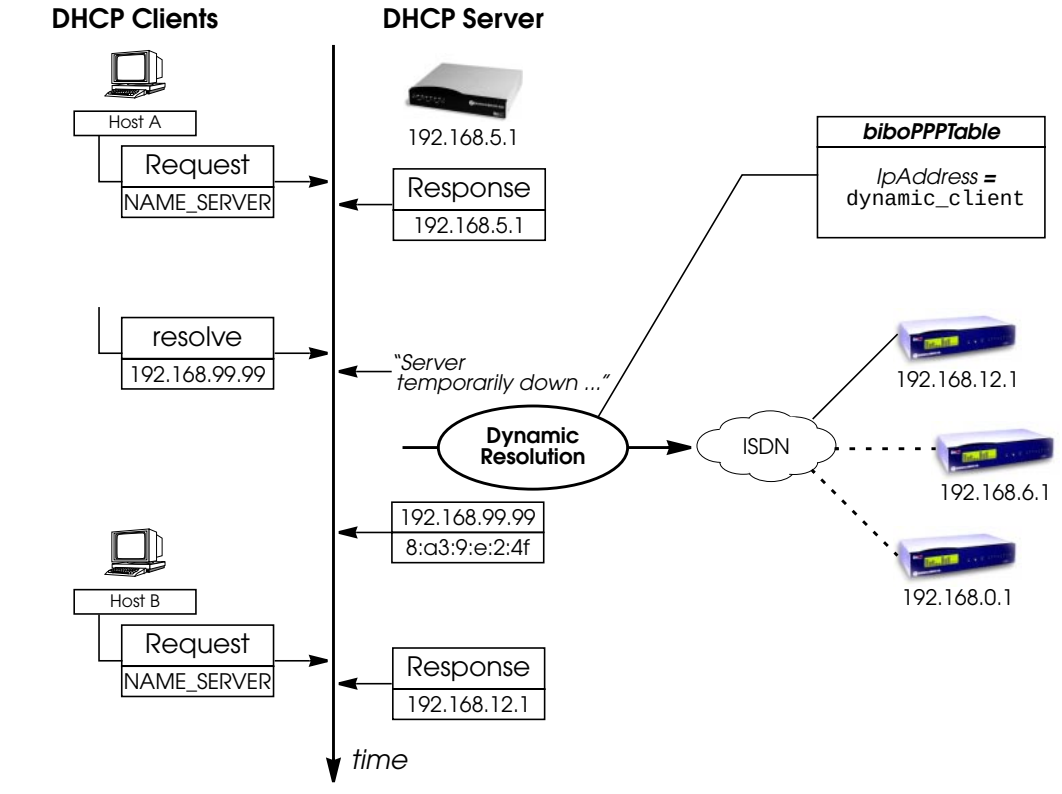
1. WINS
2. Broadcast resolution
3. LMHOSTS or Domain Name Service (DNS) if configured

Setup Tool

This setting can also be conveniently configured over Setup Tool in the menu IP –> DHCP –> ADD. The menu looks like this:

BinTec routerSetup Tool (IP) (DHCP) (ADD): Add Range of IP Addresses	BinTec Communications AG MyRouter
Interface	en1
IP Address	172.16.100.50
Number of consecutive addresses	50
Lease Time (Minutes)	300
MAC Address	0060B283D02F
NetBT Node Type	Point-to-Point Node
SAVE	CANCEL
Use <Space> to select	

Dynamic Name Server Address Resolution



- The BinTec router parses the ***biboPPPTable*** for partners that support DNS/WINS negotiation; i.e., ***DNSNegotiation*** is **enabled** and the ***IpAddress*** field is set to **`dynamic_client`** or ***DNSNegotiation*** is set to **dynamic_client**.
- While attempting to configure it's DNS server, DNS requests are answered with a "Server temporarily down" resp. "Server Failure"message.
- WAN partners are only called once.

- After successful DNS Address negotiation, the BinTec router can inform subsequent DHCP requests for a name server with its newly configured address. See the section on DNS and WINS (NBNS) Negotiation over PPP, for information on dynamic DNS address negotiation.
- Clients that were given the BinTec router's address as name server can't be informed of a new address. For these hosts, the BinTec router simply continues relaying resolution requests to the actual DNS/WINS server.

1.9 DNS and WINS Addresses over PPP

DNS and WINS (NBNS = NetBios Name Server) negotiation can be configured on a per partner basis and allow to better control how (and from which partners) the BinTec router will negotiate DNS and WINS settings.

Each partner can be separately configured so that the BinTec router either

- accepts DNS/WINS settings from the partner,
- offers DNS/WINS settings to this partner
- or does not negotiate DNS/WINS settings with the partner.

Together the MIB variables biboPPPIpAddress and biboPPPDNSNegotiation control how DNS/WINS Negotiation is handeled with the respective PPP partner.

biboPPPDNSNegotiation	The type of negotiation to perform with this client. Default value: ***enabled*** Possible values include: ***disabled*** (1), ***enabled*** (2), ***dynamic_client***(3),

biboPPPIpAddress	***dynamic_server*** (4) The type of IP address for this dial-up partner. Possible values include: ***static*** (1), ***dynamic_server*** (2), ***dynamic_client*** (3)

Th table below illustrates the effect of using this two variables to control DNS and WINS negotiation:

Variable Settings	Negotiation Handling
biboPPPDNSNegotiation: disabled	No negotiation is performed.
biboPPPDNSNegotiation: enabled *and* biboPPPIpAddress: dynamic_client	DNS resp. WINS addresses are requested from the remote side. Correspondingly the variables ***biboAdmNameServer, biboAdmNameServ2, biboAdmWINS1*** or ***biboAdmWIN2*** are overwritten.
biboPPPDNSNegotiation: enabled *and* biboPPPIpAddress: dynamic_server *or* biboPPPIpAddress: static	DNS resp. WINS addresses are sent to the remote side, when requested, as far as they are configured. The addresses are taken from the variables ***biboAdmNameServer, biboAdmNameServ2, biboAdmWINS1*** or ***biboAdmWIN2***.
biboPPPDNSNegotiation: dynamic_client	DNS resp. WINS addresses are requested from the remote side. Correspondingly the variables ***biboAdmNameServer, biboAdmNameServ2, biboAdmWINS1*** or ***biboAdmWIN2*** are overwritten.

Variable Settings	Negotiation Handling
biboPPPDNSNegotiation: dynamic_server	DNS resp. WINS addresses are sent to the remote side, when requested, as far as they are configured. The addresses are taken from the variables ***biboAdmNameServer, biboAdmNameServ2, biboAdmWINS1***or ***biboAdmWIN2***.

In Setup Tool the values of the variables ***biboPPPDNSNegotiation*** and ***biboPPPIpAddress*** can also be configured:

When you configure a WAN Partner, you will find the item **Dynamic name Server Negotiation** in the **Advanced** submenu of the WAN partner's **IP** menu. This item corresponds to the variable ***biboPPPDNSNegotiation.***

The value of the ***biboPPPIpAddress*** variable is configured via the item **IP Transit Network** in the WAN partner's **IP** menu. Here the settings **yes** and **no** correspond to the variable's value ***static*** and **dynamic client** to ***dynamic_client***, the same **dynamic server** to ***dynamic_server***.

1.10 Name Resolution with DNS Proxy

1.10.1 Why Name Resolution?

IP address = ?

Name resolution is necessary for converting host names in a LAN or on the Internet into IP addresses. For example, if you would like to reach the host "Goofy" in your LAN or enter the URL "http://www.bintec.de" in your Internet browser, you need the associated IP address before you can set up the required connection. The following options are available:

- DNS (Domain Name Server):

  A DNS stores the relevant IP addresses for host names in the form of DNS records and resolves the names if a relevant request is received, i.e. the name server sends a DNS record with the IP address associated with the name to the source of the request. Name servers form a hierarchical tree structure. If a name server cannot resolve a name, it therefore asks a higher-order name server, etc.

- HOSTS files:

  HOSTS files are located on the PCs in the LAN. You can use these files to create a table of host names with associated addresses. This means connections to DNS are no longer needed to resolve these names. As the HOSTS files must be updated on each PC, this method of name resolution is not very practicable.

In practice, the DNS of the Internet Service Provider is often used for name resolution.

1.10.2 Advantages of Name Resolution

With your BinTec router, the following functions and facilities for name resolution (port 53) are available:

- DNS Proxy, for passing DNS requests to the right DNS.
- DNS cache, for saving the results of DNS requests.
- Static name entries, for defining assignments of names to IP addresses.
- Filter function, to prevent the resolution of certain names.
- Monitoring via Setup Tool, to provide an overview of DNS requests.

This is how it works:

DNS Proxy

DNS Proxy makes the tedious updating of HOSTS files on PCs in the LAN unnecessary, as you can enter your BinTec router as DNS on the relevant PCs. DNS requests are passed by the PC to the BinTec router for processing. The configuration of the PCs in the LAN is then easy and can also be left when changing providers. This also works if the PCs in the LAN do not have any static DNS entries, but are assigned these dynamically by your BinTec router as DHCP server.
Forwarding entries enable the BinTec router to decide which DNS is to be used for the resolution of certain names. If you have configured two WAN partners on your router, your head office and your Internet Service Provider, it is advisable to have Internet names resolved by the DNS of your ISP, but names of the corporate network by the DNS of the head office. A DNS request for resolution of an internal company address usually cannot be answered by the DNS of the ISP and is thus superfluous, causes unnecessary costs and resolution takes longer than necessary. A forwarding entry, which passes DNS requests for names such as "*.intranet.de" to the WAN partner "head office", is therefore advisable.

DNS cache

If a DNS request is passed by a BinTec router to a DNS and this DNS answers with a DNS record, the resolved name is saved with the associated IP address as a positive dynamic entry in the DNS cache of the BinTec router. This means that once a name has been resolved and is required again, the BinTec router can answer the request from the cache and a new request to an external name server is not necessary. These requests can therefore be answered more quickly,

bandwidth is reduced on the WAN connections and the costs of unnecessary connections are saved.
If a DNS request cannot be answered by any of the DNS asked, this is saved in the cache as a negative dynamic entry. As failed DNS requests (requests that cannot be answered) are not usually saved by applications or IP stacks, these negative dynamic entries in the cache prevent frequent unsuccessful connection setups to external DNS.
The validity of the positive dynamic entries in the cache is given by the TTL (Time To Live), which is contained in the DNS record. Negative entries are assigned the value **Maximum TTL for Neg Cache Entries**. A dynamic entry is deleted from the cache when the TTL expires.

Static name entries

You use positive static entries to enter names with the associated IP addresses on the BinTec router. If you save frequently needed IP addresses in this way, the BinTec router can answer relevant DNS requests itself and the connection to an external name server is not necessary. This speeds up access to these addresses. For a small network, such a name server can be configured on the BinTec router. The installation of a separate DNS and the tedious updating of HOSTS files on the PCs in the LAN is not necessary.
With negative static entries, a name is not assigned an IP address, a corresponding DNS request is answered negatively and not passed to any other name server either.

You can easily change a dynamic entry to a static entry "at the press of a button" in ***IP*** ➧ ***DNS*** ➧ ***DYNAMIC CACHE*** .

Filter function

By using negative static entries, you can limit name resolution on the BinTec router using a filter function. This makes access to certain domains much more difficult for users in the LAN, as it prevents the corresponding names being resolved. You can use wildcards (*) when entering the name. When you enter a static entry, you define how long this assignment of name and IP address is valid by setting the TTL. This TTL is entered in each DNS record with which the BinTec router answers a relevant DNS request.

Make sure your static entries are always up to date. Names or IP addresses can change at any time!

Monitor function

Which IP addresses are requested by hosts in the LAN and how often?
The Setup Tool permits rapid access to this and other statistical information. You can also use the `nslookup` command in the command line (SNMP shell) to check how a name or an IP address is resolved by your BinTec router or another name server. To obtain help information for the command, enter `nslookup -?`.

1.10.3 Other Options

Global name server

In ***IP*** ➧ ***STATIC SETTINGS***, you can also enter the IP address of preferred global name servers that are to be asked if the BinTec router cannot answer requests itself or with forwarding entries.

For local applications, the IP address of BinTec router or the loopback address (127.0.0.1) can be entered as global name server.
If necessary, the BinTec router can send or receive the addresses of name servers to and from WAN partners:

Default interface

In **Default Interface**, you can also select a WAN partner to whom a connection is set up as standard for name server negotiation if name resolution was not successful using the methods already stated.

1.10.4 Exchanging DNS Addresses with LAN Partners

DHCP

If the BinTec router is configured as DHCP server, DHCP clients in the LAN can be sent IP addresses from name servers. In this case, the addresses of the global name servers entered on the BinTec router can be sent or the address of BinTec router itself. In the latter case, DNS requests from the DHCP clients are sent to the BinTec router, which either answers these itself or passes them on if necessary (proxy function).

1.10.5 Exchanging DNS Addresses with WAN Partners

IPCP

The same applies if the dynamic negotiation of name servers is activated for the IP configuration of a WAN partner and the BinTec router is operating in Server Mode (**Dynamic Name Server Negotiation** = *server (send)*). In

this case, the addresses of the global name servers or the address of the BinTec router itself can also be sent for name server negotiations via IPCP to the WAN partner, who is the IP address client.
If the BinTec router is operating in Client Mode (**Dynamic Name Server Negotiation** = *client (receive)*), name server addresses can if necessary be negotiated with the WAN partner, who is the IP address server, and sent to the BinTec router. These can be entered as global name servers on your BinTec router and are thus available for future name resolutions.

1.10.6 Strategy for Name Resolution

A DNS request is handled by the the BinTec router as follows:

1. Can the request be answered directly from the static or dynamic cache (IP address or negative answer)?

   If yes, the information is forwarded.

   If no, see 2.

2. Is a matching forwarding entry available?

   In this case, the relevant DNS are asked. If the connection to the WAN partner is not active, an attempt is made to set it up.

   If a DNS can resolve the name, the information is forwarded and a dynamic entry created in the cache.

   If none of the DNS asked can resolve the name or no matching forwarding entry is available, see 3.

3. Are global name servers entered?

In this case, the relevant DNS are asked. If the IP address of the BinTec router or the loopback address is entered for local applications, these are ignored here.

If a DNS can resolve the name, the information is forwarded and a dynamic entry created in the cache.

If none of the DNS asked can resolve the name or no static name servers are entered, see 4.

4. Is a WAN partner selected as default interface?

   In this case, the associated DNS are asked. If the connection to the WAN partner is not active, an attempt is made to set it up.

   If a DNS can resolve the name, the information is forwarded and a dynamic entry created in the cache.

   If none of the DNS asked can resolve the name or no default interface has been selected, see 5.

5. Is overwriting the global name server addresses admissible (**Overwrite Global Nameserver** = *yes*)?

   In this case, a connection is set up to the first WAN partner, which is configured so that addresses of DNS can be sent – provided this has not previously been attempted. If name server negotiation is successful, these are entered as global name servers and are therefore available for further requests.

6. Request is answered with server error.

If one of the DNS answers with "non-existent domain", this answer is forwarded to the source of the request immediately and included in the cache as negative entry.

1.10.7 Overview of Configuration with the Setup Tool

The configuration and monitoring of name resolution on the BinTec router is set in:

- ***IP*** ➧ ***STATIC SETTINGS***:
- ***IP*** ➧ ***DNS***
- ***IP*** ➧ ***DNS*** ➧ ***STATIC HOSTS***
- ***IP*** ➧ ***DNS*** ➧ ***FORWARDED DOMAINS***
- ***IP*** ➧ ***DNS*** ➧ ***DYNAMIC CACHE***
- ***IP*** ➧ ***DNS*** ➧ ***ADVANCED SETTINGS...***
- ***IP*** ➧ ***DNS*** ➧ ***GLOBAL STATISTICS...***
- ***WAN PARTNER*** ➧ ***EDIT*** ➧ ***IP*** ➧ ***ADVANCED SETTINGS***

IP ➧ ***STATIC SETTINGS*** contains the following fields:

Field	Meaning
Domain Name	Defines BinTec router's Domain Name.
Primary Domain Name Server	IP address of BinTec router's first global Domain Name Server (DNS).
Secondary Domain Name Server	IP address of another global Domain Name Server.
Primary WINS	IP address of BinTec router's first global WINS (Windows Internet Name Server) or NBNS (NetBIOS Name Server).
Secondary WINS	IP address of another global WINS or NBNS.

IP ➧ ***DNS*** contains the following fields:

Field	Meaning
Positive Cache	Enables positive dynamic entries in the cache. Possible values: • *enabled* (default value): Successfully resolved names and IP addresses are saved in the cache. • *flush*: All positive dynamic entries in the cache are deleted. • *disabled*: Successfully resolved names and IP addresses are not saved in the cache and existing dynamic positive entries are deleted (static entries are not deleted).
Negative Cache	Enables negative dynamic entries in the cache. Possible values: • *enabled* (default value): Names that could not be resolved are saved in the cache as negative entries. • *flush*: All negative dynamic entries in the cache are deleted. • *disabled*: Names that could not be resolved are not saved in the cache and existing dynamic negative entries are deleted (static entries are not deleted).

Field	Meaning
Overwrite Global Nameservers	Defines whether the addresses of global name servers on the BinTec router (in ***IP*** ➧ ***STATIC SETTINGS***) may be overwritten with name server addresses sent by WAN partners. Possible values: • *yes* (default value) • *no*
Default Interface	Defines the WAN partner to which a connection is normally set up for name server negotiation if other name resolution attempts were not successful.
DHCP Assignment	Defines which name server addresses are sent to the DHCP client if the BinTec router is configured as DHCP server. Possible values: • *none*: No name server address is sent. • *self* (default value): The address of BinTec router is sent as name server address. • *global*: The addresses of the global name servers entered on the BinTec router are sent.

Field	Meaning
IPCP Assignment	Defines which name server addresses are sent by the BinTec router to a WAN partner for dynamic name server negotiation. Possible values: • *none*: No name server address is sent. • *self*: The address of the BinTec router is sent as name server address. • *global* (default value): The addresses of the global name servers entered on the BinTec router are sent.
Static Hosts	The number of static entries is displayed in brackets.
Forwarded Domains	The number of forwarding entries is displayed in brackets.
Dynamic Cache	The number of positive and negative dynamic entries in the DNS cache is displayed in brackets.

IP ➧ ***DNS*** ➧ ***STATIC HOSTS*** ➧ ***ADD*** contains the following fields:

Field	Meaning
Default Domain:	The Domain Name of the BinTec router entered in ***IP*** ➧ ***STATIC SETTINGS*** is displayed.
Name	Host name, which is assigned the **Address** with this static entry. May also contain wildcards (*) (only at the start of **Name**, e.g. *.bintec.de). If an incomplete name is entered without a dot, this is completed with ".**Default Domain**" after confirming with **SAVE**.

Field	Meaning
Response	Defines the type of static entry. Possible values: • *positive* (default value): A DNS request for **Name** is answered with a DNS record, which contains the associated **Address**. • *ignore*: A DNS request is ignored; no answer is given (not even a negative answer). • *negative*: A DNS request for **Name** is answered with a negative answer.
Address	(Only for **Response** = *positive*) IP address, which is assigned to **Name**.
TTL	Period of validity in s for the assignment of **Name** to **Address** (only relevant for **Response** = *positive*). This value is displayed in the TTL field (Time To Live) if the BinTec router sends a corresponding DNS record. Default value: *86400* (= 24 h)

IP ➧ ***DNS*** ➧ ***FORWARDED DOMAINS*** ➧ ***ADD*** contains the following fields:

Field	Meaning
Global Nameservers:	The global name servers entered in ***IP*** ➧ ***STATIC SETTINGS*** are displayed.
Default Domain:	The Domain Name of the BinTec router entered in ***IP*** ➧ ***STATIC SETTINGS*** is displayed.

Field	Meaning
Name	Host name that is to be resolved with this forwarding entry. May also contain wildcards (only at the start of **Name**, e.g. *.bintec.de). If an incomplete name is entered without a dot, this is completed with ".**Default Domain**" after confirming with **SAVE**.
Interface	Defines the WAN partner to which a connection is set up for the resolution of **Name**.
TTL	Period of validity in s for the assignment of **Name** to **Address**. Default value: *86400* (= 24 h) If the request of the BinTec router for **Name** is answered with a DNS record, this contains a TTL field (= Time To Live in s), whose value is not normally changed by the BinTec router on forwarding the DNS record. If the TTL field received has the value 0 or exceeds **Maximum TTL for Pos Cache Entries**, then **TTL** is also sent with the DNS record forwarded.

IP ➧ ***DNS*** ➧ ***DYNAMIC CACHE*** contains the following fields:

Field	Meaning
Name	Host name, which is assigned the **Address** with this dynamic entry in the cache.
Address	IP address, which is assigned to **Name**.

Field	Meaning
Resp	Defines the type of dynamic entry. Possible values: • *positive*: A DNS request for **Name** is answered with the associated IP address from the cache. • *negative*: A DNS request for **Name** is answered with a negative answer from the cache.
TTL	Indicates how many seconds the dynamic entry remains in the cache. The entry is deleted on expiry of **TTL**. When a positive dynamic entry is saved in the cache, the value of the TTL field (= Time To Live in s) contained in the DNS record is used. If the TTL field in the DNS record is set to 0 or exceeds **Maximum TTL for Pos Cache Entries**, the value **Maximum TTL for Pos Cache Entries** is used when saving the entry. When a negative dynamic entry is saved in the cache, **Maximum TTL for Neg Cache Entries** is always assigned as this value.
Ref	Indicates how often the entry has been referenced, i.e. how often a DNS request has been answered with the entry from the cache.
STATIC	A dynamic entry can be converted to a static entry by tagging the entry with the **Space** bar and confirming with STATIC. The relevant entry then disappears from ***IP*** ➧ ***DNS*** ➧ ***DYNAMIC CACHE*** and is listed in ***IP*** ➧ ***DNS*** ➧ ***STATIC HOSTS***. **TTL** is transferred in this operation.

IP ➧ *DNS* ➧ *ADVANCED SETTINGS...* contains the following fields:

Field	Meaning
Maximum Number of DNS Records	Defines the maximum number of static and dynamic entries. Once this value is reached, an older dynamic entry is deleted from the cache when a new entry is added. The entry deleted is always the dynamic entry that has not been requested for the longest period of time. If **Maximum Number of DNS Records** is reduced by the user, dynamic entries are also deleted, if necessary. Static entries are not deleted; **Maximum Number of DNS Records** cannot be set lower than the current number of existing static entries. If **Maximum Number of DNS Records** corresponds to the number of static entries, no further dynamic entries are possible!
Maximum TTL for Pos Cache Entries	Is assigned to a positive dynamic entry in the cache as **TTL** if the field of the DNS record has the value 0 or exceeds **Maximum TTL for Pos Cache Entries**.
Maximum TTL for Neg Cache Entries	Is assigned as **TTL** to a negative dynamic entry in the cache.

IP ➧ *DNS* ➧ *GLOBAL STATISTICS...* contains the following fields (the menu is updated every second):

Field	Meaning
Received DNS Packets	Displays the number of received DNS packets, including the answer packets for forwarded requests.
Invalid DNS Packets	Displays the number of invalid DNS packets received.

Field	Meaning
DNS Requests	Displays the number of correct DNS packets received.
Cache Hits	Displays the number of requests that could be answered with static or dynamic entries from the cache.
Forwarded Requests	Displays the number of requests forwarded to other name servers.
Cache Hitrate (%)	Displays the number of **Cache Hits** per **DNS Request** in %.
Successfully Answered Queries	Displays the number of successful requests (positive and negative) answered.
Server Failures	Displays the number of requests that could not be answered by any name server (either positively or negatively).

The following part of *WAN PARTNER* ➧ *EDIT* ➧ *IP* ➧ *ADVANCED SETTINGS* is of interest for this configuration step:

Field	Meaning
Dynamic Name Server Negotiation	In the event of dynamic name server negotiation, defines whether the Bin-Tec router receives IP addresses for **Primary Domain Name Server**, **Secondary Domain Name Server**, **Primary WINS** and **Secondary WINS** from the WAN partner or sends them to the WAN partner.

The **Dynamic Name Server Negotiation** field contains the following selection options:

Possible Values	Meaning
off	The BinTec router does not send or answer requests for name server addresses.
yes	The response is linked to the mode for issuing/receiving an IP address (setting in ***WAN PARTNER*** ➧ ***EDIT*** ➧ ***IP*** under **IP Transit Network**): • The BinTec router sends requests for name server addresses to the WAN partner if *dynamic client* is selected. • The BinTec router answers requests for name server addresses from the WAN partner if *dynamic server* is selected. • The BinTec router answers but does not send requests for name server addresses if *yes* or *no* is selected.
client (receive)	The BinTec router sends requests for name server addresses to the WAN partner.
server (send)	The BinTec router answers requests from the WAN partner for name server addresses.

1.10.8 Procedure for Configuration with the Setup Tool

To do

Proceed as follows to configure name resolution with DNS Proxy on a BinTec router:

Name resolution on a BinTec router

If applicable, first enter the global name servers on the BinTec router:

- Go to ***IP*** ➧ ***STATIC SETTINGS***.
- Enter **Domain Name**, e.g. ***mycompany.com***.
- Enter **Primary** or **Secondary Domain Name Server**, if applicable.
- Enter **Primary** or **Secondary WINS**, if applicable.

> If you do not have a Secondary DNS or Secondary WINS server, you can enter the IP address of the Primary DNS or WINS server in the **Secondary Domain Name Server** or **Secondary WINS** field again.
> This may be necessary for connection to some data communications

- Press **SAVE**.

Activate or deactivate the cache function and define general settings for DNS Proxy:

- Go to ***IP*** ➧ ***DNS***.
- Select **Positive Cache** and **Negative Cache**, e.g. *enabled*.
- Select **Overwrite Global Nameservers**, e.g. *yes*, if you do not wish to enter any static global name servers under ***IP*** ➧ ***STATIC SETTINGS***.
- Select **DHCP Assignment**, e.g. *self*.
- Select **IPCP Assignment**, e.g. *global*.

Defines the values for the static and dynamic entries:

- Go to ***IP*** ➧ ***DNS*** ➧ ***ADVANCED SETTINGS...***.

- Enter **Maximum Number of DNS Records**.
- Enter **Maximum TTL for Pos Cache Entries**.
- Enter **Maximum TTL for Neg Cache Entries**.
- Press **SAVE**.

How to create static entries:

- Go to ***IP*** ➧ ***DNS*** ➧ ***STATIC HOSTS***.
- All the existing static entries are listed here.
- You can create a new entry with **ADD**.
- Enter **Name**.
- Select **Response**.
- Enter **Address**, if applicable.
- Enter **TTL**.
- Press **SAVE**.

How to create forwarding entries:

- Go to ***IP*** ➧ ***DNS*** ➧ ***FORWARDED DOMAINS***.
- All the existing forwarding entries are listed here.
- You can create a new entry with **ADD**.
- Enter **Name**.
- Select **Interface**.
- Enter **TTL**.
- Press **SAVE**.
- Select **EXIT**.
- Press **SAVE**.

BinTec router <–––> WAN partner

Proceed as follows if you would like to configure a WAN partner so that the address of a name server is sent from the BinTec router to the WAN partner or from the WAN partner to the BinTec router, as applicable:

- Go to ***WAN PARTNER*** ➧ ***EDIT*** ➧ ***IP*** ➧ ***ADVANCED SETTINGS***.
- Select **Dynamic Name Server Negotiation**.
- Confirm with **OK**.
- Press **SAVE**.

Monitoring and statistics

- How to obtain a list of dynamic entries in the cache:
- Go to ***IP*** ➧ ***DNS*** ➧ ***DYNAMIC CACHE***.
- This menu contains a list of all the dynamic entries in the cache.
- To convert a dynamic entry into a static entry, tag the entry with the **Space** bar and confirm with **STATIC**.
- The entry disappears from the list of dynamic entries and is listed as a static entry under ***IP*** ➧ ***DNS*** ➧ ***STATIC HOSTS***.
- How to obtain a list of some static parameters:
- Go to ***IP*** ➧ ***DNS*** ➧ ***GLOBAL STATISTICS...***.
- Here you will find some statistics for DNS Proxy.

1.11 Dynamic IP Address Assignment

As the name suggests, Dynamic IP Address Assignment is a method used to configure a host's IP address dynamically. It's generally based on a client-server system; clients ask for an address and the server assigns one.

Although it's used for a variety of reasons by different sites, it's primary benefit is that it allows for efficient (and centralized) management of a limited number of IP addresses. Internet service providers commonly use it to assign IP addresses to dial-in hosts at connections time.

NOTE: DHCP can also be used to provide hosts with an IP address (and other information, see: BOOTP and DHCP) but is mainly used for assigning IP addresses to hosts on the BinTec router's LAN.

Dynamic IP Address Assignment on the BinTec router is used for hosts that connect to the BinTec router via ISDN; the BinTec router can operate as a Server or as a Client for

such hosts. Configuring Server Mode or Client Mode is described below.

1.11.1 Server Mode

It is possible to define separate IP Address Pools for dynamic IP address assigments. For Internet Service Providers (ISP) and other sites with many dial-in accounts, using IP address pools is convenient for defining separate user groups. One might assign “official” addresses from one pool 1 for special accounts, and assign “non-official” addresses from pool 2 for private accounts.

In server mode, the BinTec router assigns an IP address to a host (the client) at connection time from the Pool (Pool ID) defined for the respective WAN Partner. When dynamically assigning an IP address to a dial-in client the static IP address respectively the Pool from which the address is retrieved are determined in the following order.

1. **Assigning a Static IP Address**
   When there exists an entry in the ***ipRouteTable*** for the dial-in client, where ***ipRouteMask*** is set to a host route (= ***255.255.255.255***) and ***ipRouteType*** has the value ***direct***, in this case the IP address stored in the variable ***ipRouteDest*** of this routing entry is taken to be assigned for this WAN partner.
   If caller can’t be authenticated locally via the MIB, RADIUS server(s) are contacted. If a server authenticates the caller, and there is a User-Record entry
   BinTec-ipRouteTable=”ipRouteMask=255.255.255.255
   ipRouteType=direct
   ipRouteDest= x”
   the IP address stored in the variable ***ipRouteDest*** of this entry is taken to be assigned for this WAN partner.

2. **Assigning an IP Address from an Address Pool**
   When the procedure described under 1.) was not successful, the IP address is assigned from the Pools.
   Once the caller is identified (either inband or outband), the WAN partner's ***biboPPPTable*** entry is compared. If the ***IPAddress*** field = "dynamic_server" AND an address is available from the pool identified by the ***PoolId*** field, then a free address is assigned.
   If caller can't be authenticated locally via the MIB, RADIUS server(s) are contacted. If a server authenticates the callerand there is a User-Record entry BinTec-biboPPPTable="biboPPPIpAddress=dynamic_server", the pool ID is determined from the User-Record entry BinTec-biboPPPTable="biboPPPIpPoolId=x".

For detailed description of individual system table fields please refer to the MIB Reference on the accompanying Companion CD or at *BinTec's WWW* site.

Example Configuration of an IP Address Pool via Setup Tool

A. Dial-In Partner without RADIUS

IP → DYNAMIC IP ADDRESS → ADD

First, create/modify a Pool ID to contain IP addresses that will be available for assignment at connect time.

Pool ID	1
Number	10.5.5.5
Number of Consecutive Addresses	5

WAN PARTNER → ADD

Here you'll need to set:

Partner Name	test
Encapsulation	*PPP*
Compression	none
Encryption	none
Calling Line ID	no

Then, in the IP submenu configure the BinTec router as a Dynamic IP Address server for this partner.

IP Transit Network	dynamic_server

In the ADVANCED SETTINGS submenu define the Pool ID

IP Address Pool	1

B. Dial-In Partner with RADIUS server

IP → DYNAMIC IP ADDRESS → ADD

Next, modify a Pool ID to contain IP addresses that will be available for assignment at connect time.

Pool ID	2
Number	192.168.80.20
Number of Consecutive Addresses	20

Then you must define the following entry in the User-Record of the RADIUS server:

BinTec-biboPPPTable=”biboPPPIpPoolId=2”

Example Configuration of IP Address Pools via SNMP Shell

A. Dial-In Partner without RADIUS

1. Create an IP address pool in the ***biboPPPIpAssignTable***.

```
brick:> biboPPPIpAssignAddress=10.5.5.5 biboPPPIpAssignPoolId=1 biboPPPIpAssignRange=5
```

2. Set the WAN partner in ***biboPPPTable*** to use Pool ID.
   Assuming entry 4 is the existing WAN partner we want to configure for Dynamic IP address assignment.

```
brick:> biboPPPIpPoolId:4=1 biboPPPIpAddress:4=dynamic_server
```

B. Dial-In Partner with RADIUS server

1. Create an IP Address pool in the ***biboPPPIpAssignTable***.

```
brick:> biboPPPIpAssignAddress=192.168.80.20 biboPPPIpAssignPoolId=2
        biboPPPIpAssignRange=20
```

2. Define the following entry in the User-record of the RADIUS server:

   BinTec-biboPPPTable=”biboPPPIpPoolId=2”

3. Once the caller is authenticated via a RADIUS server a temporary ***biboPPPTable*** entry is generated. The ***PoolId*** field for this entry is determined by the contents of the User-Record discussed above.

Overlapping Address Pools

Although it’s legally possible to define IP address pools that overlap (as shown below) the BinTec router will not assign an address twice.

The ***biboIpInUseTable*** is consulted for this purpose. The ***biboIpInUseTable*** shows all IP addresses, which are dynamically assigned to WAN partners or reserved for WAN partners and is continously updated.

Example for overlapping Address Pools:

```
brick: biboPPPIpAssignTable> biboPPPIpAssignTable

inx     Address(*rw)  State(-rw) PoolId(rw)    Range(rw)

0       10.5.5.1      unused     0             2
1       10.5.5.2      unused     1             2
2       10.5.5.3      unused     2             2

brick: biboPPPIpAssignTable>
```

With the ***biboPPPIpAssignTable*** shown above, only four IP addresses could actually be used at any given time.

An address pool may be removed from the table at any time by assigning **`delete`** to the respective ***State*** object.

Reserved IP Addresses

In the ***biboIpInUseTable*** all IP addresses currently assigned or reserved to a partner are shown.

After a disconnect the ***State*** of the entry is set to reserved and the variable ***PPPIPInUseAge*** is reset to 0. From then on it is tried to reserve the IP address for the partner for a maximum of 3600 s.

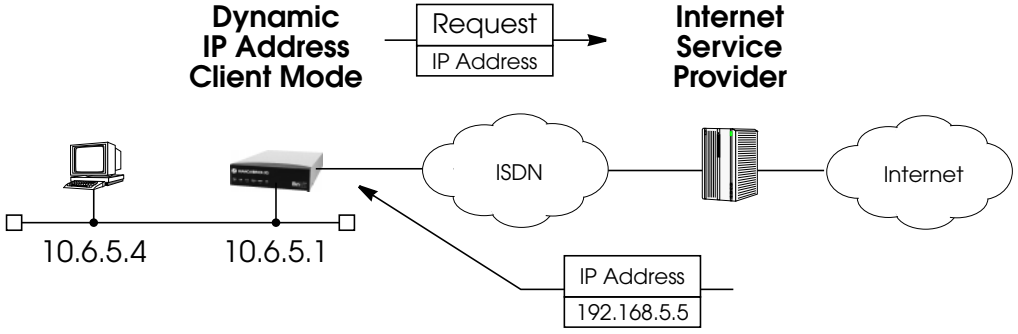
When within this time the same partner calls again, the BinTec router tries to assign the respective address again. The assignement is made via the variable ***PPPIpInUseIdent***.

When a partner dials in and no reserved IP address is available, the next step is to assign a free IP address from the specified pool. When no more free IP address is available from the pool, the oldest of the IP addresses, reserved for other partners, is used.

1.11.2 Client Mode

In client mode, the BinTec router can be configured to accept its own IP address at connection time from a dial-up PPP partner at connection time. The BinTec router will use

this IP address as the local side of the dialup connection as long as the connection is established.



Upon receiving an IP address from the dialup (server) host, the BinTec router will automatically create an IP route to allow hosts on the LAN to access networks via the dialup connection. The route is also automatically removed when the connection is closed.

1.12 Bandwidth on Demand

1.12.1 Bandwidth on Demand for leased lines

In order to support high levels of traffic over leased lines, bandwidth on demand (BOD) over different media, i.e.dynamic channel bundling of leased lines and dialup lines is possible.
You can bundle channels over an optional table, the **pppExtIfTable**, which can also be configured over Setup Tool. You can, for example, specify if you want one or more dialup lines to be bundled with your leased line when the load on the leased line reaches a certain level for a certain period of time. You can also specify the kind of algorithmus used by which the load of your leased line or bundle is cal-

culated or the number of seconds such a sample calculation should take, giving you more control over the switching of dialup lines.
BOD is activated by setting the **pppExtIfBodMode** variable to *BOD_active* or *BOD_passive*. Switching on and off of additional B-channels only occurs in the active mode, i.e. one partner must be configured as an active part the other as a passive part, otherwise call collisions are unavoidable.
Dialup lines are dynamically switched on and off when the line utilization of the leased line/bundle reaches a certain level for a certain period of time, see Switchover thresholds for BOD.

1.12.2 Backup for leased line connections

Backup operation is also available for leased line connections. Should a leased line fail, for example, a dialup connection is dynamically initiated.
There is no necessity to define another interface for the backup case. IPX backup configuration is also possible.

1.12.3 Bandwidth on Demand when leased line is down

Bandwidth on Demand is also available for backup connections, the modes used can be either *BOD_active*, *BOD_passive* or *BOD_backup*. To avail of BOD when the leased line is down, the maximum number of switchable B-channels in the **biboPPPTable** (**biboPPPMaxConn)** must be correspondingly configured, i.e. set to greater than *1*.
If the leased line fails, the first switchable B-channel is used as the backup channel, so a second is required to provide this channel with Bandwidth On Demand. Load-dependent switching then occurs from the side that established the link

1.12.4 Bandwidth on Demand for pure dialup lines

The extensions to the ***pppExtIfTable*** can also be used for pure dial-up interfaces. In contrast to the behaviour of leased-line interfaces, there is no static configuration of the partner to activate switching. The **pppExtIfBodMode** variable should always be set to *BOD_active* (in Setup Tool: enabled) to activate BOD for a dialup interface. Load-dependent switching then always occurs from the side that established the initial connection.

The MIB table variables in the PPP group (***pppExtIfTable***) with example and default values:**.**

```
mybrick : > pppExtIfTable
inx IfIndex(rw)          BodMode(-rw)        Algorithm(rw)
    Interval             Load(ro)            MlpFragmentation(rw)
    MlpFragSize(rw)

    5000                 backup              equal
    5                    0                   proportional
    50
```

The meanings of the different variables:

Variable	Meaning
IfIndex	Interface index of the relevant leased line interface.

Variable	Meaning
BodMode	Contains the following values: *backup – only for leased line bundles* If the leased line fails, backup operation is activated. When the leased line is available again, the backup connection is terminated. BOD is also available for this mode, provided **biboPPPMaxConn** is set to more than 1. *bod_active* Only one partner should be configured as the active partner. Switching on and off of additional channels then only occurs from this side. *bod_passive* No switching on and off of additional B-channels occurs from the side of the partner configured as *bod_passive*. He participates as a passive partner in the channel bundle. *disabled* No extra channels are opened to support leased lines.
Algorithm	An algorithm for the weighting of throughput within a specified time frame for the calculation of capacity utilization (load). The following values are possible: *equal*: constantly weighted load over the given time frame. This means that all values considered in the time frame contribute in equal measure to the calculation of load. *proportional*: current proportional weighting within the time frame in favour of the latest throughput values. This means the calculation of **load** is influenced least by the first throughput value in the time frame and most by the most recent value added to the time frame.
Interval	Maximum period in seconds for a throughput measurement sample which is used to calculate the load.
Load	A calculation of the capacity utilization of the bundle in % depending on the selected algorithmus and used to determine whether another line should be switched either on or off.

Variable	Meaning
MlpFragmentation	A mode according to which MLP fragments are formed. *proportional*: the fragment size taken from the available bandwidth of the individual links in relation to the total bandwidth of the bundle. This means that if, for example, an X.21 link has 128 000 kbit/s bandwidth and dialup lines 2 and 3 have 64 000 kbit/s each, the X.21 link will receive a greater proportion of each packet, i.e. a larger fragment, than lines 2 and 3. *equal*: as far as is possible, fragments of equal size are formed, the weighting of the bandwidth of a link is made by the number of fragments to be sent on the link. Using the above example, the leased line simply receives more fragments of packets; the fragments, however, remain of equal size.
MlpFragSize	Here you can configure the size of the fragment to be sent. If you have left the setting *proportional* above (default), the maximum size is used of the number you have configured (you can, of course, leave the default value of 50) and the calculation according to the **MlpFragmentation** mode. If you have selected equal above, the number you configure represents the size of the fragments.

1.12.5 Setup Tool configuration

The structure and content of the Setup Tool menues vary slightly from router to router (a pure dialup-line bundle configuration on BinGO!, for example, will differ in structure from the BRICK X21), the basic functionality remains the same, however.
The following configuration of a leased line with a dialup line bundle via Setup Tool is an example using an X.21 leased line. X.21 is only supported by routers with a CM-X21 module. Configuration of X.21 leased line and dialup line bundles is the same as for ISDN leased line and dialup line bundles.

1. In the main menu of Setup Tool, go to **WAN Partner.**

```
BIANCA/BRICK-X21 Setup Tool                BinTec Communications AG
(WAN): WAN Partners                                        BRICK-X21


Current WAN Partner Configurtion
Partnername          Protocol              State
Leased, Slot 2 (0)   ppp                   down
Partner_1            ppp                   down
xi2                  ppp                   up

ADD          DELETE                EXIT


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

2. Select the configured WAN partner you want Bandwidth On Demand for, in the example below xi2. A leased line exists for this WAN partner

```
BIANCA/BRICK-X21 Setup Tool                BinTec Communications AG
[WAN][EDIT]: Configure X.21 Leased Line                    mybrick


   Partner Name                              xi2


   Encapsulation                        ppp
   Compression                          none
   Encryption                           none

   PPP >
   Advanced Settings >

   IP >
   IPX >
   Bridge >

                       SAVE                CANCEL
Press, to scroll, tag/untag DELETE, to edit
```

3. Select Advanced Settings

```
BIANCA/BRICK-X21 Setup Tool                BinTec Communications AG
[WAN][EDIT][ADVANCED]:Advanced Settings                      mybrick

    Extended Interface Settings (optional)

                        OK                    CANCEL

Press, to scroll, tag/untag DELETE, to edit
```

4. Select **Extended Interface Setting**

```
BIANCA/BRICK-X21 Setup Tool                BinTec Communications AG
[WAN][EDIT][ADVANCED][EXTIF] :Extended Interface Settings(xi2) mybrick

    Extended Interface Settings not configured yet!

    Mode                          Bandwidth on Demand active
    Line Utilization Weighting    equal
    Line Utilization Sample (sec)5
    Maximum Number of Dialup Channels2

    SAVE          DELETE CONFIGURATION     CANCEL

Press, to scroll, tag/untag DELETE, to edit
```

The variables from the new MIB table, **pppExtIfTable,** can be identified now on the above Setup Tool page as follows:

- **BodMode** is configured under **Mode**.
- **Algorithm** is configured under **Line Utilization Weighting**.
- **Interval** is configured under **Line Utilization Sample**.

Additionally on this Setup Tool page, **Maximum Number of Dialup Channels** (**PPPMaxConn** of the **biboPPPTable**), corresponds to the number of channels to be dynamically switched.
If you have configured **Mode** to *Bandwidth On Demand active, Bandwidth On Demand passive* or *Bandwidth On Demand backup* and then you press **SAVE**, you will return to the previous page, which is now supplemented by the WAN Numbers variable, see below.

```
BIANCA/BRICK-X21 Setup Tool              BinTec Communications AG
[WAN] [EDIT]: Configure X.21 Leased Line                 mybrick

    Partner Name            xi2

    Encapsulation           ppp
    Compression             none
    Encryption              none

    PPP >
    Advanced Settings >
    WAN Numbers >

    IP >
    IPX >
    Bridge >

              SAVE                    CANCEL

Press, to scroll, tag/untag DELETE, to edit
```

In order for the dialup connection to be dynamically switched to your partner, it is now necessary to enter the WAN number of the partner and direction, either incoming if you have configured *Bandwidth on Demand passive*, outgoing if you have configured Bandwidth on Demand active or both (these settings correspond to the **biboDialDirection** or **biboDialNumber** variables in the **biboDialTable**).

This concludes the configuration of dynamic channel bundling for leased lines over Setup Tool.

Switchover thresholds for BOD

BOD is activated by setting the **pppExtIfBodMode** variable to *BOD-active* or *BOD-passive*, depending on which side should actively switch on and bear the costs. The maximum number of B-channels to be dynamically switched corresponds to the value of the variable **biboPPPMaxConn** of the **biboPPPTable**; this is configured in Setup Tool under **Maximum Number of Dialup Channels**.

- Switching on of B-channels:
  If the **pppExtIfLoad** corresponds to the value 90 (%) or more for at least 5 seconds, a B-channel is switched on.
- Switching off of a B-channel:
  The current value of **pppExtIfLoad** does not serve as the basis for switching off, the calculated (fictitious) bundle load after switching off of a B-channel does. For example, the fifth dialup line is switched off if the remaining four lines in the bundle would have a load of less than 80% for 10 seconds.
  There are three mechanisms for deciding when a dialup connection is switched off. The first is fixed and is a precondition for the third to take effect, the second and third can be configured separately.

1. If this value drops below 80 (%) for at least 10 seconds, a B-channel is switched off.
2. Static Short Hold: terminates all BOD/backup links after expiry of the inactivity timeout configured. Static Short Hold always takes priority over the load utilization calculation. If, for example, static Short Hold is set to 2 seconds and there is no more

data exchange on a channel bundle, the dialup line is terminated after the two seconds and not after the 10 seconds of number 1.

3. Dynamic Short Hold: if the **PPPTable** is correspondingly configured and AOCD (advice of charging during the call) is available, a B-channel is switched off just shortly before the beginning of the next charging unit, provided that the terms of number 1 are fulfilled, i.e. the current capacity utilization is less than 80% for 10 seconds.

Authentication

On establishing a PPP leased line, though accepted, no authentication is required by the partner. Authentication is essential, however, for the dialup link bundled with the leased line and should be configured in the **biboPPPTable** accordingly. Authentication over Setup Tool is set in the menu

In this case, authentication of the partner is requested for incoming BOD/backup calls.

LCP echo requests (*PPP keepalive*)

LCP Echo Requests are only generated on existing leased lines, not, however, on the switched B-channels.

X.21 leased lines

The setting for **X.21IfLeads** can have a significant bearing on costs incurred and is thus worthy of some attention. When **IfLeads** is set to *enabled*, a switched backup connection is immediately initiated on the failing of an X.21 leased line. This can, however, lead to excessive and undesired dialup connections if, for example, a flickering leased line sends repeated signals to establish dialup connections. On

the other hand, it provides a means of assuring the speedy transmission of data.
By setting **X.21IfLeads** to *disabled*, the backup connection is only established after a period of time set in **biboPPPTimeOut**. This is set to 10 x 3000 milliseconds, which is equal to 30 seconds by default. You can thus be sure that due to an unsteady leased line, a series of unwanted dialup connections is not established, and that a backup connection is only made when the leased line has been down for a set time.

The remaining variables in the biboPPPTable and their relevance for leased with BOD/backup lines.

Variable	Leased with BOD/BACKUP
Encapsulation	only ppp, x75_ppp, x75btx_ppp
Timeout	Supported
IpAddress	Not supported
RetryTime	Not supported
BlockTime	Supported
MaxRetries	Supported
ShortHold	Supported
InitConn	Not supported
MaxConn	Supported
MinConn	Not supported
CallBack	Not supported
Layer1Protocol	Supported
LoginString	Not supported

Variable	Leased with BOD/BACKUP
VJHeaderComp	Supported
Layer2Mode	Not supported
DynShortHold	Supported

1.13 Routing with OSPF

OSPF on the BinTec router consists of 11 system tables and the ospfmon application, see External Commands in the chapter on the SNMP Shell. An overview of the 10 OSPF tables (from the **`ospf`** group) and the ***ipImportTable*** (**`ip`** group) from the SNMP shell are shown below.

1.13.1 OSPF System Tables

- ***ospfGeneralGroup***
  Global settings used by the OSPF protocol including the ***ospfAdminStat*** object (must be enabled to use OSPF).
- ***ospfStatTable***
  Status information about Link-State advertisements and OSPF protocol packets that have been sent or received.
- ***ospfErr***
  Status information about bad OSPF packets (bad checksum, incorrect field values, etc.) that have been received.
- ***ospfAreaTable***
  Identifies OSPF areas the BinTec router's interfaces are assigned to and logs statistics for each.

- ***ospfLsdbTable***
  Contains header information from the BinTec router's Link State DB.
- ***ospfIfTable***
  Lists all OSPF interfaces, their current state, and settings specific to that OSPF interface.
- ***ospfIfMetricTable***
  Lists the actual metric values being used for each OSPF interface.
- ***ospfNbrTable***
  Lists the neighbor routers that have been identified via then HELLO protocol and their respective OSPF states.
- ***ospfAreaAggregateTable***
  Specifies IP address ranges for route condensation (also called: inter-area route summarization) among areas.
- ***ospfStubAreaTable***
  Generates a default route for Stub Areas.
- ***ipImportTable***
  Specifies how routes from one routing protocol are imported into another routing protocol.

1.13.2 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 BinTec router includes support for Demand Circuits (RFC 1793) the dialup link is only opened when changes in routing information must be propagated.

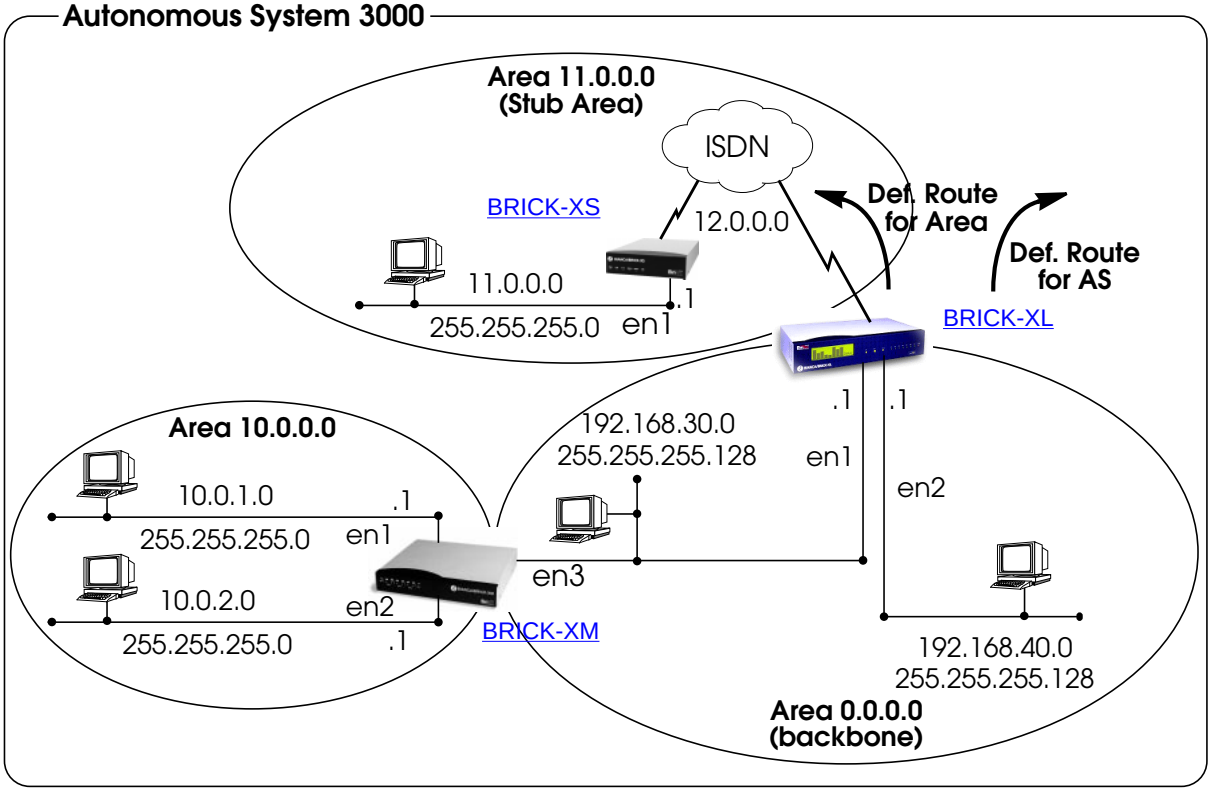
Area 0.0.0.0 (backbone)

- 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.



Configuration Overview

All BinTec routers:

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.

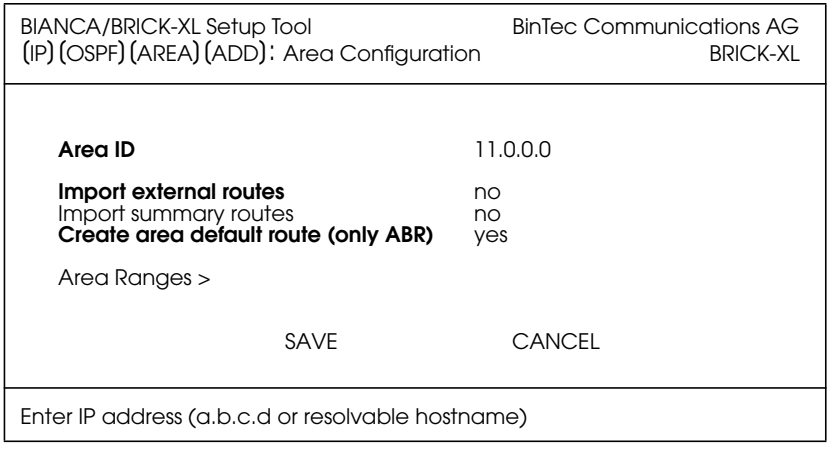
Configuration Steps for BRICK-XL

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

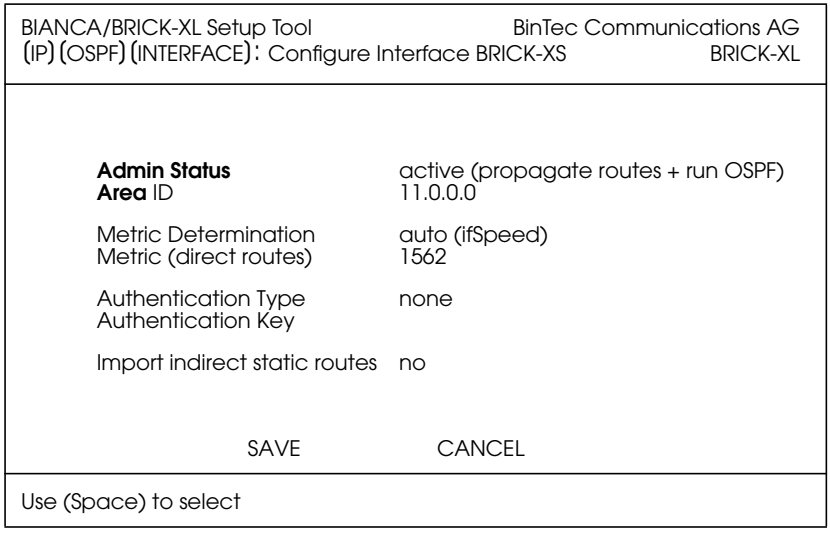
Area and have BRICK-XL generate the default route for this area.



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 .



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 BinTec router) 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 BinTec router goes into Overflow State. In Overflow State two things happen:

1. The BinTec router begins to flush all external advertisements generated locally.

2. The BinTec router 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 BinTec router 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 BinTec router then 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.

1.13.3 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.

ipImportSrcProto \ ipImportDstProto	rip	ospf
default_route		3[1]
direct		
static		3[2]
rip	–	
ospf	3[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.

1.14 Advanced IP Features

Several advanced IP features which are described below are available via settings in the ***ipExtIfTable*** shown below.

```
mybrick: admin> ipExtIfTable

inx Index(*ro)          RipSend(rw)          RipReceive(rw)
    ProxyArp(rw)        Nat(rw)              NatRmvFin(rw)
    NatTcpTimeout(rw)   NatOtherTimeout(rw)  NatOutXlat(rw)
    Accounting(rw)      TcpSpoofing(rw)

mybrick: ipExtIfTable >
```

1.14.1 Access Lists

NOTE:

Updates and Access Lists
If you are updating from a software release equal to or older than 4.7.x to 5.1.1 or later, it is necessary to firstly upgrade to 4.9.3, save the configuration (cmd = save), then upgrade to 5.1.1. If you update directly from 4.7.x to 5.1.x, the old access lists (***ipAllowTable***, ***ipDenyTable***) can not be automatically converted and are lost.

The IP Access List methodology used on the BinTec router is based upon a concept of Rules, Filters, and so-called Chains.

Suggested Method for Configuring Access Lists

Because the potential danger exists of “locking oneself out of the system” when configuring IP Access Lists the following order of events should be used.

1. Define the set of filters to use.
2. Disable access lists for all interfaces by setting the “FirstRule” to “0 (no access rules)”.
3. Define the complete set of rules.
4. Enable the rule(s) for the desired interfaces.

Access List Methodolgy

An Access Filter simply describes a subset of IP traffic and may be based upon one or more of the following attributes.

- Source and/or Destination IP address.
- Source and/or Destination Port.
- Source and/or Destination Protocol.

An Access Rule defines an:

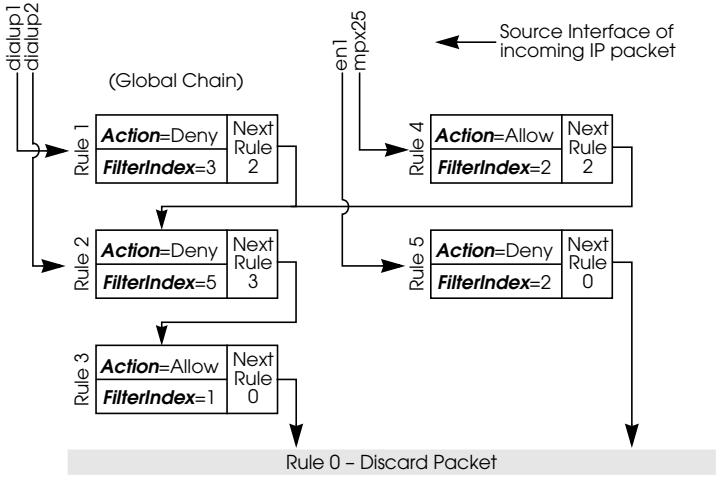
1. Access Filter to compare the packet to.
2. Action to take if a packet matches/doesn’t-match a filter.
3. Index of the next rule to use if no action was taken.

Each Rule references a NextRule allowing different *Chains* (sequence of Rules) to be defined. For each interface a separate starting rule must be defined (via the ***ipExtIfRuleIndex*** field) that determines which Rule chain is applied. Rule 1 has special meaning: it is used by default for all newly created interfaces.
Rules are applied until one of the following events occurs:

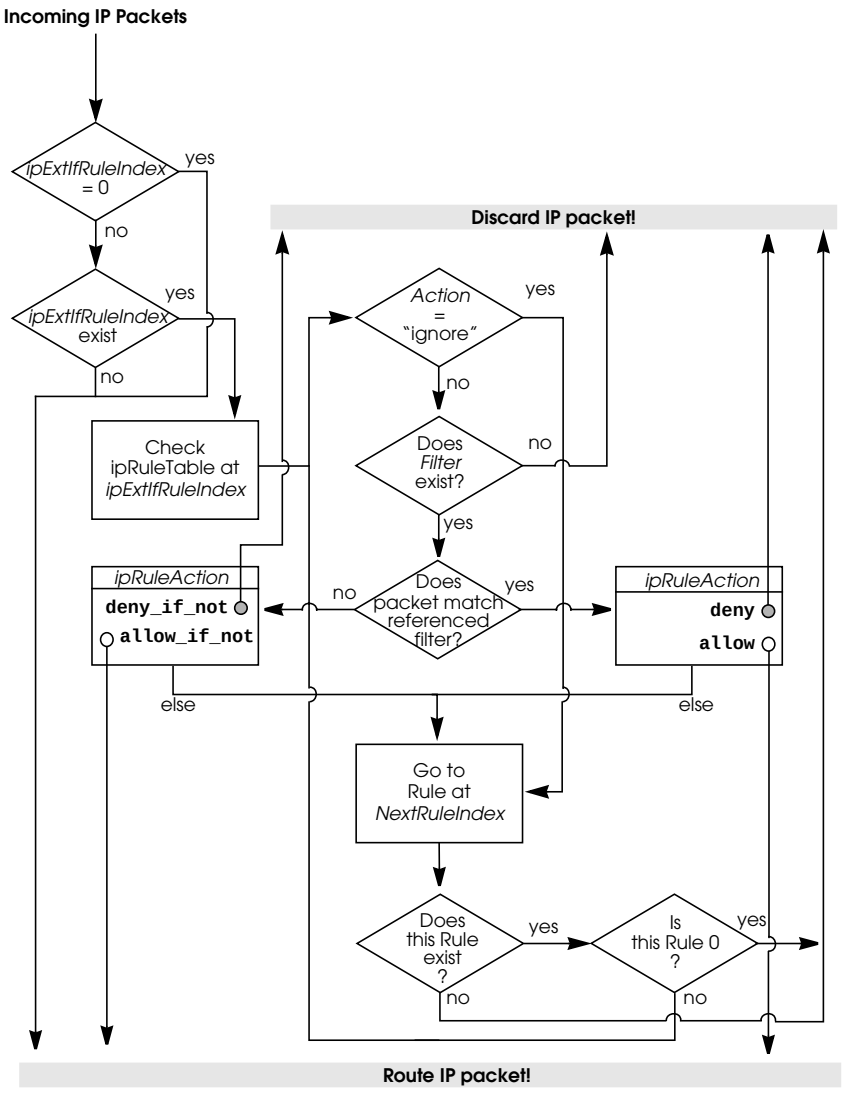
- The packet matches and the ***Action*** is “match” based OR the packet doesn’t match and the ***Action*** is “if_not” based.

- The packet is discarded if the end of the chain or Rule 0 is reached.

In the diagram below, packets arriving via the “dialup1” interface are compared to Rules 1–2–3 while packets arriving on the “mpx25” are applied to Rules 4–2–3.



The diagram below shows in detail how Access List, Rules and Filters are applied to incoming IP traffic.


Incoming IP Packets
ipExtIfRuleIndex = 0
yes
no
ipExtIfRuleIndex exist
yes
no
Discard IP packet!
Action = "ignore"
yes
no
Does Filter exist?
no
yes
Check ipRuleTable at ipExtIfRuleIndex
ipRuleAction
deny_if_not
allow_if_not
Does packet match referenced filter?
no
yes
ipRuleAction
deny
allow
else
else
Go to Rule at NextRuleIndex
Does this Rule exist ?
yes
Is this Rule 0 ?
yes
no
no
Route IP packet!

Setup Tool Menus

Go to IP → ACCESS LISTS →

The IP–>Access Lists menu displays three submenus where IP Access Lists settings are configured.

```
Setup Tool                                          BinTec Communications AG
[IP][ACCESS]: IP Access Lists

                              Filters
                              Rules
                              Interfaces

                              EXIT

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

The FILTERS menu is used to configure filters. Each filter describes a subset of IP traffic and may be address, protocol, source or destination port based.
The RULES menu is used to configure rules. Rules can be ordered, or "chained" to control the order in which the filters are applied.
The INTERFACES menu is used to define which rule is used first for traffic arriving on that interface.

Go to IP → ACCESS LISTS → FILTERS →

This menu lists the currently configured IP Access Filters and shows the Index number, Description, and Conditions for each filter. In the Conditions column, abbreviations (explained in the menu) are used to describe the type of filter (i.e. address or port based filter).

To add a new filter select ADD. The menu shown below will be displayed.

```
Setup Tool                                          BinTec Communications AG
[IP][ACCESS][FILTER][ADD]: Configure IP Access Filter

    Description             no http
    Index                   4

    Protocol                any

    Source Address          192.168.50.5
    Source Mask             255.255.255.0
    Source Port             any


    Destination Address
    Destination Mask
    Destination Port        specify
    Specify Port            80

                  SAVE                CANCEL

Enter integer range 0..65535
```

Description A text string can be entered here to describe the filter. Note that in other menus only the first 15 characters of the description may be displayed.

Index The index field can't be changed. The BinTec router assigns a new filter number here automatically as new filters are added.

Protocol Select a predefined protocol or "any" to match all protocols.

Source/Destination Address (optional) Enter the source (or destination) IP address to match IP packets from.

Source/Destination Mask (optional) Apply an optional mask.

Source/Destination Port The range of port numbers to apply. Use “specify” to select a pecific port number, “specify range” to selec a range of port numbers by entering the first and the last port to be included in the range, “any” to match all ports numbers, or one of the predefined ranges, as explained in the table below.

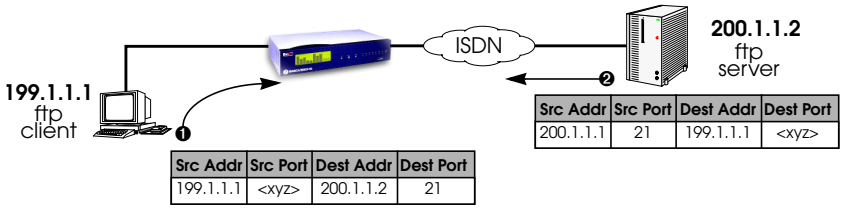
Source Port Ranges

0 ...1023	1024...4999	5000...32767	32768...65535
privileged	unprivileged		
server	clients	server	clients
specify / specify range			

Specify Port If “specify” or “specify range” is set in the previous field, the port number or port number range must be set here.

Using Source and Destination Port Numbers

Along with the source and destination addresses, the Internet Protocol uses source and destination ports numbers to identify data connections uniquely. The client side generates a number (xyz) which is used as the source port for the destination port it uses the number the server offers the service on. The server sends IP packets with the port numbers reversed in respect to the client. A simplified ftp connection might look like this.

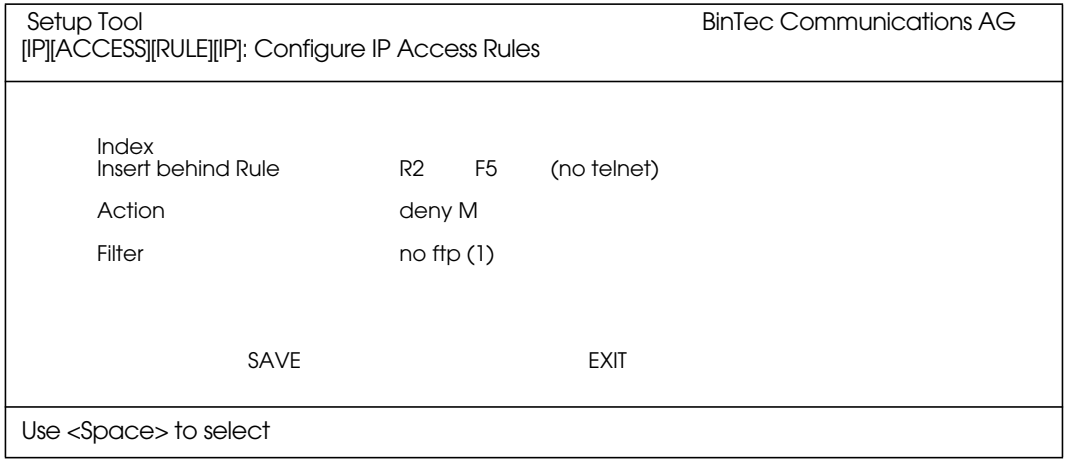


Go to IP → ACCESS LISTS → RULES →.

This menu lists configured Rule Chains (individual chains are separated by a line). For each rule the Rule Index, Filter Index, Next Rule Index, Action, Filter, and Conditions are shown.
If a Rule (i.e., a link in the chain) is deleted from the list all neighbouring rules in the chain are automatically relinked.

1. Select ADD to create new rules. The menu below will be displayed. For each rule an Action and Filter must be defined that defines what to do when a packet matches that filter.
2. Select DELETE to remove an existing Rule that has been marked for deletion (Using the spacebar.).

3. Select REORG to reorganize the order of the rules in a chain. See the following page.



Index This value can not be changed but is displayed when editing an existing rule. When creating new rules this field is empty until the rule is saved.

Insert behind Rule (only shown when creating new rules) Use the scrollbar to select the location in the chain where this new rule should be inserted.
For example: If you already have a global rule chain 1–3–2–0, selecting 3 here results in the chain 1–3–4–2–0. To start a new (separate) rule chain,use the scrollbar and select “none” in this field.

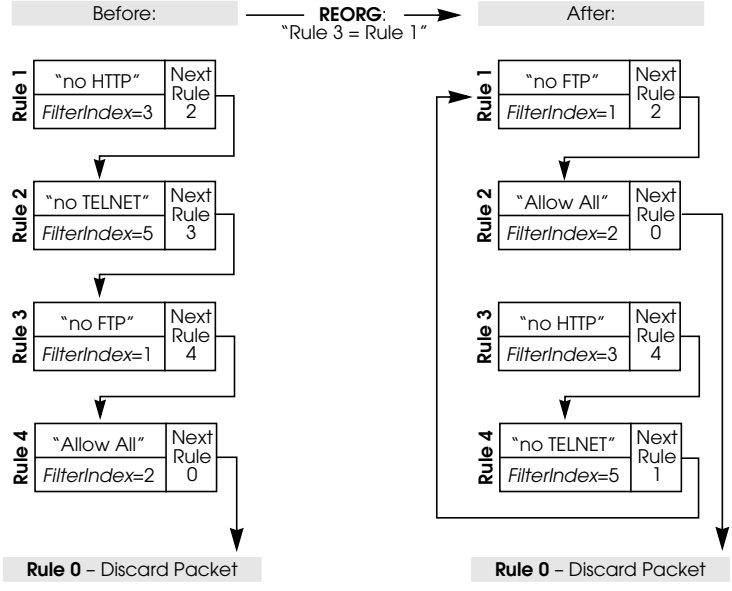
Action The action field defines whether to allow or discard the packet based on whether or not the packet matches the filter (defined in the following field) or not.

Filter The Filter to test IP packets against;

use the spacebar to scroll through the list of currently configured filters.

Reorganizing Rules in a Chain

The REORG menu allows you to change the order of Rules in an Access Rule chain.
After selecting the Rule that should be placed at the beginning of the chain (the "Index of Rule that gets Index 1" field), remaining Rules are automatically relinked. The appropriate Rule Index and Next Rule Index numbers are reassigned in the ***ipRuleTable*** and the interface-specific Start Rules are updated in the ***ipExtIfTable***.



NOTE: The appropriate indicies are renumbered but the access semantics remain the same.

Go to IP → ACCESS LISTS → INTERFACES →.

This menu is used to control which Rule Chain(s) are used for packets arriving via the BinTec router interface. This menu lists all IP capable interfaces and the First Rule that is currently being used for this interface.
To change the First Rule for any interface, highlight the entry and hit the Return key; otherwise select Exit to accept the displayed settings.

NOTE: By default Rule 1 is always used for newly created BinTec router interfaces.

```
Setup Tool                                          BinTec Communications AG
[IP][ACCESS][INTERFACES]: Configure First Rules

   Configure first rules for interfaces

   Interface        First Rule              First Filter
   en1              0 (no access rules)
   sales1           2                       3 (all else)
   sales2           2                       3 (all else)
   branch           2                       3 (all else)



   EXIT

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

In the EDIT/ADD menu the following fields are displayed.

Interface This value can not be changed but is displayed for reference.

First Rule Use the scrollbar to select the Rule to use first for packets arriving on this interface. Setting this field to “none”

disables the Access List mechanism for this interface.

NOTE: If the referenced Rule doesn't exist (in ***ipRuleTable***) then all packets arriving on this interface will be allowed.

1.14.2 Local TCP/UDP Service Access Rules

For additional security, access to specific TCP or UDP services on the BinTec router can be controlled using the ***localTcpAllowTable*** and ***localUdpAllowTable*** described below.

Access rules for TCP and UDP services are "Service" based. Access to a service can be based upon combinations of the following criteria:

- The BinTec router interface the TCP connection request (or UDP packet) arrived on.
- The IP address of the originating host.

The general rule for accepting/denying access to BinTec router TCP/UDP services is as follows.

If an access rule exists for a TCP or UDP service, incoming connections to that service are allowed ONLY if:

1. The source address is 127.0.0.1, OR
2. No access rule exists for the requested service, OR
3. The incoming packet matches at least one access rule.
   i.e. source address = AllowAddr/AllowMask or
   source interface = AllowIfIndex

Examples

localTcpAllowTable

The ***localTcpAllowTable*** defines access rules for TCP services. Each entry defines an access rule for a specific TCP service. Controllable TCP services (***Service*** field) include:

telnet trace snmp capi

tapi rfc1086 http

The ***localTcpAllowTable*** entries shown below:

1. Limit access to the BinTec router´s HTTP service to a single host (at IP address 192.168.12.2), and
2. Limit access to the BinTec router´s telnet service to all hosts on the 192.168.5.0

```
mybrick:>localTcpAllowTable

inx AddrMode(-rw)      Addr(*rw)       Mask(rw          IfMode(rw)
    IfIndex(rw)        Service(rw)

00  verify             192.168.12.2    255.255.255.255  dont_verify
    0                  http
01  verify             192.168.5.0     255.255.255.0    dont_verify
    0                  telnet
```

localUdpAllowTable

The ***localUdpAllowTable*** defines access rules for UDP services. Each entry defines an access rule for a specific UDP service. Controllable UDP services (Service) include

snmp rip bootps dns

The following ***localUdpAllowTable*** entry limits access to the SNMP service on the BinTec router on the LAN.

```
mybrick:>localUdpAllowTable

inx AddrMode(-rw)    Addr(*rw)      Mask(rw        IfMode(rw)
    IfIndex(rw)      Service(rw)

00  dont_verify      0.0.0.0        0.0.0.0        verify
    1000             snmp
```

1.14.3 IP Session Accounting

In many cases, the logging of each individual TCP/IP session is desirable. IP accounting can be turned on for an interface by setting the interface's *Accounting* object in the ***ipExtIfTable*** to "on". Then, for each TCP, UDP or ICMP session that is routed over the interface, an entry in the ***ipSessionTable*** is created. Since this table is updated dynamically, viewing this table allows one to monitor all active sessions.

Once a session is terminated, either by disconnection of the TCP session or timeout, an accounting record is written to the ***biboAdmSyslogTable*** (***Subject***=acct, ***Level***=info). Records can also be sent to remote log hosts using the syslog protocol (see the section Logging with Remote LogHosts, in Chapter 7, System Administration on the BRICK.

NOTE:

Logging with the Syslog protocol is unreliable. In rare cases, accounting records may get lost. Generally, it's best to configure log hosts that are on the same LAN segment as the BinTec router, use caution when configuring hosts that can only be reached via ISDN.

1.14.4 Network Address Translation

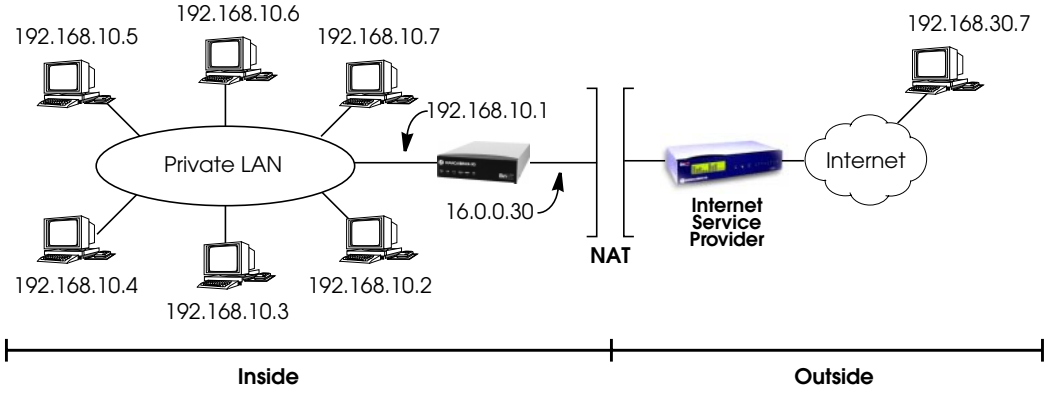
NAT (Network Address Translation) allows the BinTec router to hide a complete LAN behind one IP address and may be useful in different installations where:

- Security is an issue.
  (controlling access to a limited number of hosts)
- The number of available IP addresses is limited.
- Monitoring of incoming connections is desired.

The BinTec router performs NAT by keeping track of all TCP/IP sessions and manipulating all incoming/outgoing IP packets to reflect different source and destination addresses.

In the diagram shown below, the BinTec router's ISDN interface can be configured for NAT. All hosts on the Private LAN at 192.168.10.0 can still access external networks, but will appear to external hosts as the same host (16.0.0.30). Connections initiated from outside the private LAN can access local hosts only after local hosts have been explicitly configured (on the BinTec router) to accept connections from external hosts. Finally, it is also possible to

specify the IP address translation and remote address(es) for outgoing sessions.



Enabling NAT

Step 1 First, a route for the externally visible IP address is needed. For our example shown above, we would create a route to use our ISDN interface with:

```
mybrick: admin> ipRouteIfIndex=10001 ipRouteDest=16.0.0.30 ipRouteNexthop=16.0.0.30
                ipRouteMask=255.255.255.255 ipRouteType=direct

02: ipRouteIfIndex.16.0.0.30.2( rw):      10001
02: ipRouteDest.16.0.0.30.2( rw):         16.0.0.30
02: ipRouteNextHop.16.0.0.30.2( rw):      16.0.0.30
02: ipRouteMask.16.0.0.30.2( rw):         255.255.255.255
02: ipRouteType.16.0.0.30.2(-rw):         direct

mybrick: ipRouteTable >
```

NOTE: This example NAT configuration assumes that the LAN is properly configured and that the appropriate routes are present to allow hosts to connect to external networks.

Step 2 Next, enable the interface the BinTec router should perform NAT for; this is “10001” for our example from step 1. Locate the appropriate table entry in the ***ipExtIfTable***, and set ***Nat*** to “on”.

```
mybrick: ipExtIfTable> ipExtIfTable

inx Index(*ro)              RipSend(rw)             RipReceive(rw)
    ProxyArp(rw)            Nat(rw)                 NatRmvFin(rw)
    NatTcpTimeout(rw)       NatOtherTimeout(rw)     NatOutXlat(rw)
    Accounting(rw)          TcpSpoofing(rw)         AccessAction(rw)
    AccessReport            Ospf(rw)                OspfMetric(rw)
    TcpCKsum(rw)            BackRtVerify(rw)        RuleIndex(rw)
    Authentication(rw)      RouteAnnounce(rw)
...
02  10001                   ripV1                   both
    off                     off                     yes
    3600                    30                      on
    off                     off                     ignore
    info                    active                  auto
    check                   off                     0
    off                     strict                  3600
    60                      up_dormant
...
mybrick: admin> ipExtIfNat:02=on
02: ipExtIfNat.10001.2( rw):      on

mybrick: ipExtIfTable >
```

At this point, all hosts on the LAN are inaccessible from external networks but can continue to establish external connections at will.

Allowing Incoming Connections

To allow network connections to hosts on the private LAN (from external networks/hosts), explicit permission must be configured in the ***ipNatPresetTable***. Packets arriving from external networks and addressed to different (official) addresses can be linked to various (internal) addresses. Each entry in this table defines a specific port on a specific host that can be accessed from outside the private LAN.

For each entry, the following variables must be defined:

Ifindex	The *Ifindex* NAT is being performed on,
IntAddr	The host that is to be accessed
Protocol	The protocol (TCP, UDP, ICMP) to allow.
IntPort	The port on the specified host to allow (ftp, telnet, nntp, etc). Required only if the Protocol uses ports (TCP and UDP). –

In the following example, several servers can be entered for the same service in the ***ipNatPresetTable.*** In oder that each of these can be reached with a different external address, the external addresses must be entered in the ***ipNatPresetTable*** and the external mask set to ***255.255.255.255.***

```
inx IfIndex(*rw)        Protocol(*-rw)     RemoteAddr(rw)     RemoteMask(rw)
    ExtAddr(rw)         ExtMask(rw)        ExtPort(*rw)       ExtPortRange(rw)
    IntAddr(rw)         IntPort(rw)

00  10001               tcp                0.0.0.0            0.0.0.0
    192.1.1.1           255.255.255.255    80                 -1
    10.1.1.1            -1

01  10001               tcp                0.0.0.0            0.0.0.0
    192.1.1.2           255.255.255.255    80                 -1
    10.1.1.2            -1

02  10001               tcp                0.0.0.0            0.0.0.0
    192.1.1.3           255.255.255.255    80                 -1
    10.1.1.3

mybrick: ipNatPresetTable >
```

We can use the special internal address 0.0.0.0 to allow access to all hosts on our LAN. The entry below would be used to allow all ICMP packets to enter the private LAN.

```
mybrick: admin>ipNatPrIfIndex=10001 ipNatPrIntAddr=0.0.0.0 ipNatPrProtocol=icmp ipNat-
PrExtPort=-1

inx IfIndex(*rw)      Protocol(*-rw)   RemoteAddr(rw)   RemoteMask(rw)
    ExtAddr(rw)       ExtMask(rw)      ExtPort(*rw)     ExtPortRange(rw)
    IntAddr(rw)       IntPort(rw)

1   0001              icmp             0.0.0.0          0.0.0.0
    0.0.0.0           0.0.0.0          -1               -1
    0.0.0.0           -1
mybrick: ipNatPresetTable >
```

Mapping Addresses for Outgoing Traffic

The ***ipNatPresetTable***, however, only controls incoming traffic initiated outside the LAN, allowing access to all or just to specified internal hosts.

In order to map outgoing traffic, i.e. internal addresses initiated within the LAN, to specified and different, external IP addresses, and to specify the remote address(es) packets with these NAT addresses should be sent to, a further table, the ***ipNatOutTable***, is available. In the following example, packets from the hosts with the internal addresses 10.1.1.1 to 10.1.1.3 are to be sent with NAT with the external IP addresses 192.1.1.1. to 192.1.1.3.

```
inx IfIndex(*rw)          Protocol(*-rw)   RemoteAddr(rw)
    RemoteMask(rw)    ExtAddr(rw)          RemotePort(rw)
    RemotePortRange(rw)IntAddr(rw)         IntMask(rw)

00  10001                 any              0.0.0.0
    0.0.0.0               192.1.1.1        -1
    -1                    10.1.1.1         255.255.255.255

01  10001                 any              0.0.0.0
    0.0.0.0               192.1.1.2        -1
    -1                    10.1.1.2         255.255.255.255

02  10001                 any              0.0.0.0
    0.0.0.0               192.1.1.3        -1
    -1                    10.1.1.3         255.255.255.255

mybrick: ipNatOutTable >
```

If no matching entry is found, the IP address is set to the IP address defined on the interface configured for NAT. If a matching entry is found, the source IP address of outgoing IP packets is set to the value of ***ipNatOutExtAddr***. This table is only used if the outgoing address translation is activated (if ***ipExtIfNatOutXlat*** from the ***ipExtIfTable*** is set to on).

Entries in the table are created and removed manually by network management.

The ***ipNatOutTable*** consists of the following variables:

ipNatOutIfIndex This variable specifies the interface index, for which the table entry shall be valid. If set to 0, the entry will be valid for all interfaces configured to use NAT.

ipNatOutProtocol Possible values: icmp (1), tcp (6), udp (17), any (255), delete (256)
This variable specifies the protocol, for which the table

entry shall be valid.
Default value: any

ipNatOutRemoteAddr Together with ***ipNatOutRemoteMask,*** this variable specifies the set of target IP addresses for which the table entry is valid. If both variables are set to 0.0.0.0, the table entry will be valid for any target IP address.

ipNatOutRemoteMask Together with ***ipNatOutRemoteAddr,*** this variable specifies the set of target IP addresses for which the table entry is valid. If both variables are set to 0.0.0.0, the table entry will be valid for any target IP address.

ipNatOutExtAddr This variable specifies the external IP address to which the internal IP address is mapped.

ipNatOutRemotePort Together with ***ipNatOutRemotePortRange***, this variable specifies the range of port numbers for outgoing calls, for which the table entry shall be valid. If both variables are set to -1, the entry is valid for all port numbers. If ***ipNatOutPortRange*** is set to -1, the entry is only valid, when the port number of an outgoing call is equal to ***ipNatOutRemotePort***. Otherwise, the entry is valid, if the called port number is in the range RemotePort .. RemotePortRange.
Default value: -1
Possible values: –1..65535

ipNatOutRemotePortRange Together with ***ipNatOutRemotePort,*** this variable specifies the range of portnumbers for outgoing calls, for which the table entry shall be valid. If both variables are set to –1, the entry is valid for all portnumbers. If ***ipNatOutPortRange*** is set to -1, the entry is only valid, when the portnumber of an

outgoing call is equal to ***ipNatOutRemotePort***. Otherwise, the entry is valid, if the called portnumber is in the range RemotePort .. RemotePortRange.
Default value: -1
Possible values: –1..65535)

ipNatOutIntAddr Together with ***ipNatOutIntMask***, this variable specifies the internal host's IP address for outgoing calls matching the table entry. If both variables are set to 0.0.0.0, the table entry will be valid for any source IP address.

ipNatOutIntMask Together with ***ipNatOutIntAddr,*** this variable specifies the internal host's IP address for outgoing calls matching the table entry. If both variables are setto 0.0.0.0, the table entry will be valid for any source IP address.

Session Monitoring

While NAT is operating, you can see a list of established connections in the ***ipNatTable***. This table changes dynamically as sessions from local hosts are opened/closed. A session may be either a tcp connection, a udp connection or an icmp connection with icmp-echo messages (ping). A valid session is either an outgoing session or an incoming session

specified in the ***ipNatPresetTable***. An example ***ipNatTable*** for our installation might look as follows.

```
mybrick: admin>ipNatTable

inx IfIndex(*ro)        Protocol(*ro)      IntAddr(*ro)       IntPort(*ro)
    ExtAddr(ro)         ExtPort(ro)        RemoteAddr(ro)     RemotePort(ro)
    Direction(ro)       Age(ro)

00  10001               tcp                192.168.10.5       21
    16.0.0.30           456                192.168.19.19      80
    incoming            0 00:01:28.00

01  10001               tcp                192.168.10.2       80
    16.0.0.30           456                10.0.70.123        1605
    outgoing            0 00:18:08:40

02  10001               tcp                192.168.10.3       23
    16.0.0.30           456                192.168.55.10      77
    outgoing            0 00:00:05.00

mybrick: ipNatTable >
```

Here we see that three hosts have active sessions.

Entry 00 shows:

The host at 192.168.19.19 has been connected to the FTP service, port 21 on 192.168.10.5.

Entry 01 shows:

Our local host at 192.168.10.2 has an HTTP connection (port 80) open with the host at 10.0.70.123.

Entry 02 shows:

Our local host at 192.168.10.3 has a telnet session (port 23) opened with the host at 192.168.55.10.

The ***Age*** field specifies the period of time since the last packet was sent or received for this session.

1.14.5 NetBIOS over NAT

In networks using Windows computers, several network functions such as domain registration, access to drives and printers of other computers are based on the NetBIOS protocol.

WAN connections with NetBIOS over IP

NetBIOS was actually designed for use in LANs. NetBIOS addressing constraints make it impossible to split up a network into subnets with different locations and thereby establishing a hierarchical construction, for example.
In order to avail of the network services mentioned above over WANs, the NetBIOS packets are packed in IP packets, for example, (NBT or NetBIOS over TCP/IP). This can be activated for Windows computers under Start → Settings → Control Panel → Network → Protocols → Properties → where TCP/IP is installed.
Once packed in IP packets, NetBIOS packets can also be sent to destinations over routers and WAN connections. In this way, hierarchies are formed.

Heavy traffic loads with NetBIOS over IP

The amount of data traffic between two computers or applications connected by NetBIOS can often be unexpectedly heavy. Frequently, data exchange can take place even when no activity is apparent. For "on demand" WAN connections for which costs are charged, even in local networks as is usual in Europe, the switching of NetBIOS over IP involves an element of risk in terms of costs.

The reduction of traffic

In order to reduce traffic between locations, Microsoft recommends a concept by which a domain controller is used

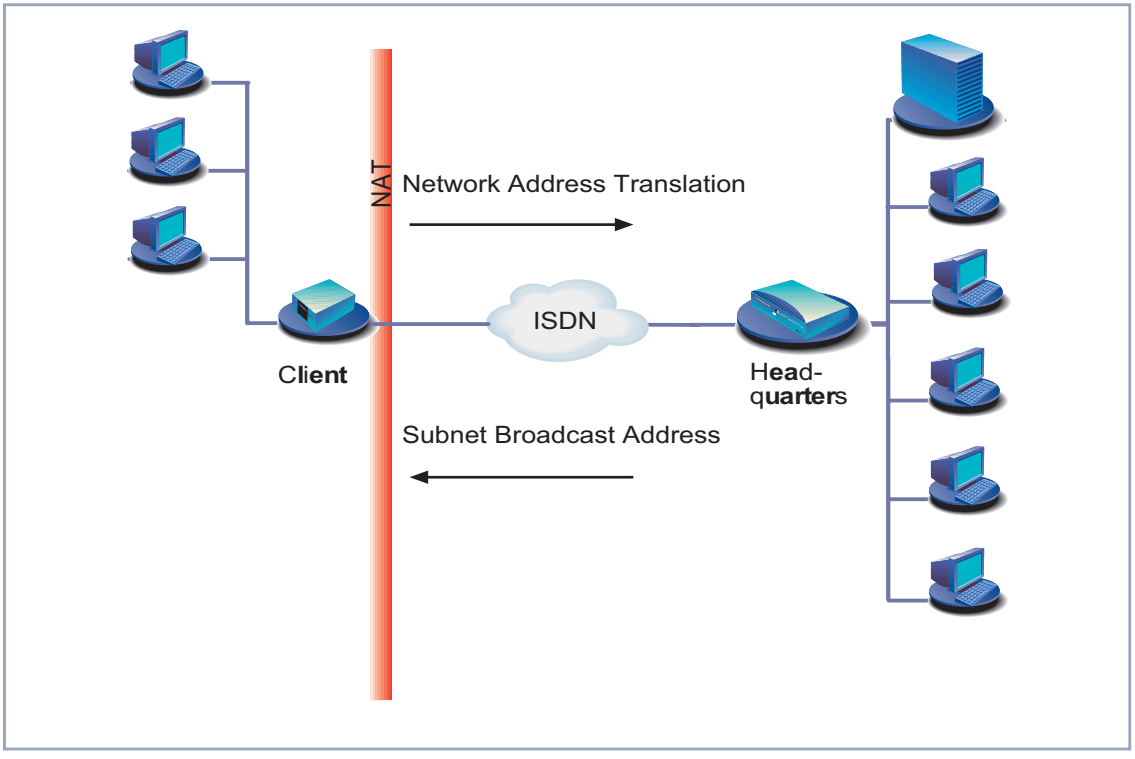
at each location (Windows NT server configured as primary and backup domain controller). This already reduces traffic levels.
In addition, on the support side, Microsoft provides a number of tuning measures (Microsoft Knowledgebase) which help to reduce traffic levels even further. To explain these changes (mostly registry changes), would exceed the scope of this document.

Network Construction without an additional domain controller

In many scenarios, e.g. teleworkers or smaller branch offices using their own router, a concept with several domain controllers is inappropriate due to the increased financial and administrative costs involved.
The construction of such networks without the use of domain controllers was already possible using BinTec routers, provided all computers, including those on the central side, had different network addresses.

Establishing a network with NAT

BinTec routers can be configured with NAT for WAN connections with NetBIOS over IP. The BinTec router thus independently manipulates not only the IP packets themselves, but also the NetBIOS packets contained within them, enabling domain registration at the central side and access to central computers, printers and drives.
The use of NAT simplifies routing to the central side as each location is represented by just the one IP address. As the illustration below shows, packets returning from the WAN partner are sent to all computers in the LAN with a subnet broadcast address.



Prerequisite – Switching Access Lists

If the resources of a central side are to be accessed over WAN links and remote access servers, all routers involved must be configured to route NetBIOS over IP traffic (TCP and UDP packets from and to ports 137 to 139). This is frequently prevented by access lists or packet filters.

Caution

By the switching of NetBIOS over IP, unexpectedly high connection costs can be incurred on WAN links. The real volume depends on the applications and services used. Consequently, the costs of the connections should be regularly monitored (in the beginning daily). To guard against

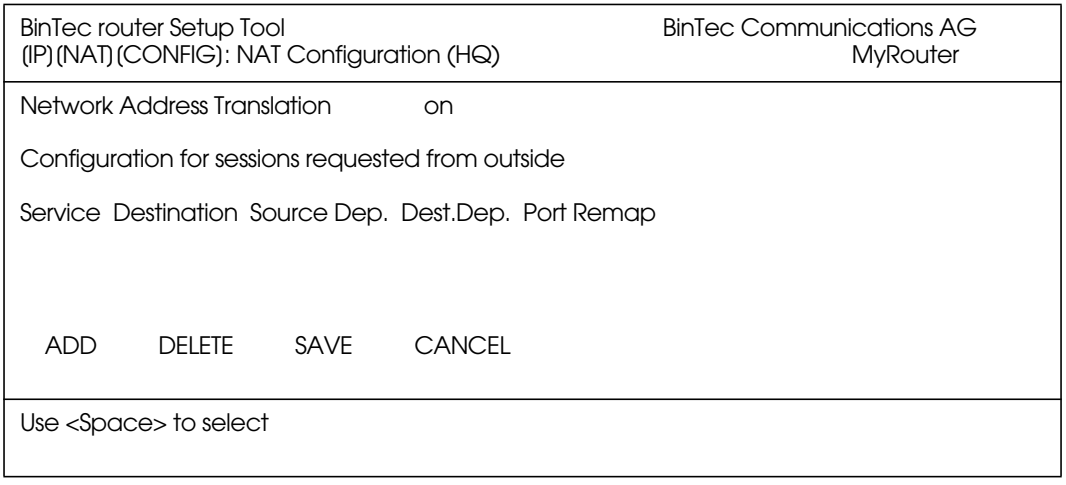
such unintentional costs, you can avail of the Credits Based Accounting System feature which is available with all BinTec routers and with which connections and costs can be limited.

Activating NAT

- In order to activate NAT over Setup Tool, go to **IP** –> **Network Address Translation**

```
BinTec router Setup Tool                          BinTec Communications AG
(IP) (NAT): NAT Configuration                              MyBinGO!

Select IP Interface to be configured for NAT

HQ
en1
en1-snap


EXIT

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

- Mark the interface or the WAN partner for which you want to activate NAT (e.g. HQ) and press **Return**.
- Another menu window opens:



Make the following entries:

- Select Network Address Translation: **ON**.
- Press **SAVE.**
  Network Address Translation is activated for the selected interface or WAN partner.

Now on routing, all source IP addresses in the LAN are replaced by this address. The NAT address is assigned to the BinTec router by IPCP from the WAN partner. The client PC appears on the WINS server on the central side in the WINS databank under this address.

Entering a subnet broadcast address

Essentially, there are three TCP/UDP ports involved in the process of domain registration over NAT: 137, 138 and 139. UDP port 138 is of special interest. Here an additional entry in the **ipNatPresetTable** is essential for NetBIOS over NAT to function properly. The reason is this. According to Microsoft´s implementation, the source and destination ports

of these packets destined for the domain controller are always port 138. This means that it is not possible to change and link the source port number with the client PC´s IP address, thus specifying the PC to which the packet should be sent back within the LAN. The domain controller will always return the packet over the port number 138. The solution to this is simply to enter a subnet broadcast address under **Destination** so that all PCs in the LAN receive the packets. This can also be done over Setup Tool in the menu IP –> Network Address Translation –> ADD

BinTec routerSetup Tool (IP) (NAT) (CONFIG) (ADD): NAT Configuration	BinTec Communications AG MyRouter
Service	user defined
Protocol	udp
Port (-1 for any)	138
Destination	172.16.98.255 (e.g. a broadcast address)
SAVE	CANCEL
Use <Space> to select	

Finally, actual domain registration and access to the external resources in the **Network Neighbourhood** take place over the NetBIOS session service over port 139. For each of these TCP/IP connections an entry is automatically made in the ***ipNatTable*.**

Example:

```
mybrick: ipNatTable

inx IfIndex(*ro)        Protocol(*ro)     IntAddr(*ro)      IntPort(*ro)
    ExtAddr(ro)         ExtPort(ro)       RemoteAddr(ro)    RemotePort(ro)
    Direction(ro)       Age(ro)

00  10001               tcp               172.16.100.99     687
    172.16.200.20       1023              172.16.201.10     139
    outgoing            50


mybrick: ipNatTable >
```

Note:

If you want to save yourself the trouble or time of calculating the broadcast address yourself, you can use a subnet calculator tool like the one you can find at this address: http://www.cci.com/tools/subcalc/index.html

Determining a subnet broadcast address

In order to determine the broadcast address, you need to know the IP address of the BinTec router and the subnet mask. It is then necessary to identify which portion of the IP address is the host number. Each IP address consists of a network portion that identifies the network number and a host portion that identifies the host´s number on that network. The dividing line between the two portions of the address depends on the network "Class" the address belongs to. We can pinpoint this dividing line for all classes, however, by examining the bit values in the netmask as follows:

1. If the bit in the mask is ON (=1), the respective bit in the IP address belongs to the network portion.
2. If the bit in the mask is OFF (=0), the respective bit in the IP address belongs to the host portion.

Example 1: **Class B: IP address 128.66.12.1, mask 255.255.255.0, subnet broadcast address 128.66.12.255.**

Here is an example of a class B subnet mask. In a standard Class B netmask, the last two bytes identify the host portion of the IP address; in the case of this subnet mask, all eight bits of the third byte define the subnet part of the address, while all eight bits of the fourth byte correspond to the host portion.
In this case, the first three bytes identify the network, including subnets, and the last byte identifies the host. Consequently, the host number to use below is 1.

Address	(dec.):128.	66.	12.	1
	(bin.): 1000 0000	0100 0010	0000 1100	0000 0001
SubNetmask(dec.)255		255	255	0
	(bin.): 1111 1111	1111 1111	1111 1111	0000 0000

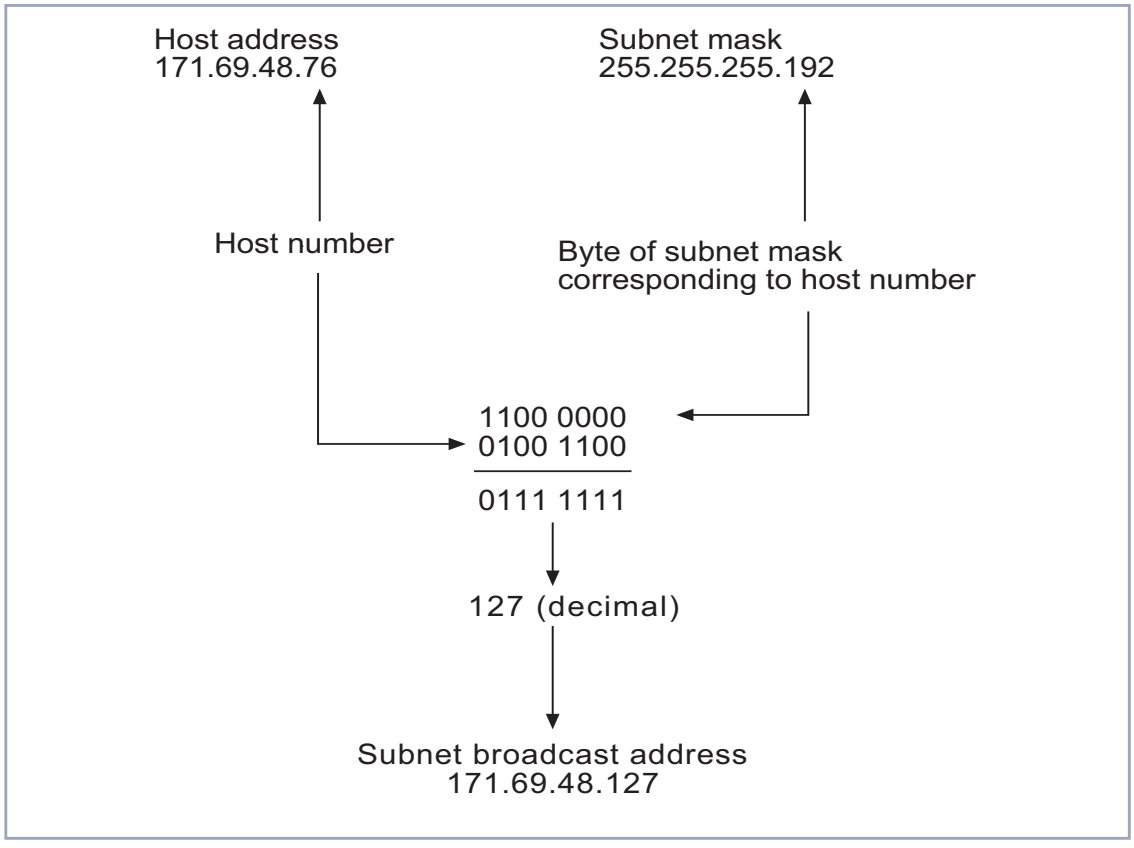
Example 2: **Class C: IP address 192.178.16.66, mask 255.255.255.192, subnet broadcast address 192.178.16.66.127.**

Next, an example of a subnet netmask from Class C. In a standard netmask, the last byte indicates the host portion of the IP address; in this case, the first two (or high-order) bits of the fourth byte (11) define the subnet part of the address, and the last six bits of that same byte (00 0000) correspond to the host portion.
The host number to use below is 66:

Address	(dec.):192.	178.	16.	66
	(bin.):1100 0000	1010 1010	0001 0000	0100 0010
Subnetmask(dec.):255		255	255	192
	(bin.):11111111	11111111	11111111	1100 0000

- Convert the host number (IP address) from decimal to binary. To do this, you can use a calculator tool such as the one supplied with Windows NT: Click the Windows Start button and point to Programs –> Accessories.
- Convert the corresponding bytes of the subnet mask from decimal to binary.
- Combine the binary subnet mask and the host number as follows:
  If the subnet mask bit is 1 and the host number bit is 0, place a 0 in that position.
  If the subnet mask bit is 1 and the host number bit is 1, place a 1 in that position.
  If the subnet mask bit is 0, place a 1 in that position.
- Convert the resulting binary number back to its decimal equivalent.
- Join the decimal number you have just determined with the network number.

The following figure is an illustration of the sequence of commands leading to the calculation of a subnet broadcast address outlined above.



Here are some more examples of IP addresses, standard and subnet masks and their corresponding broadcast addresses from classes A, B and C.

Class	IP Address	Standard / subnet mask	Broadcast Address
A	18.20.16.91	255.0.0.0	18.255.255.255
A (subnet)	18.20.16.91	255.255.0.0	18.20.255.255

Class	IP Address	Standard / subnet mask	Broadcast Address
B	171.69.48.18	255.255.0.0	171.69.255.255
B (subnet)	171.69.48.18	255.255.255.248	171.69.48.23
C	192.178.16.66	255.255.255.0	192.178.16.255
C (subnet)	192.178.16.66	255.255.255.192	192.178.16.127

Bidirectional access

After domain registration has been successfully negotiated, the PDC (Primary Domain Controller) in turn tries periodically to establish a TCP connection with the client over port 139. These connection attempts should normally fail if no additional entry is made in the **ipNatPresetTable**. Thus, only one-way access is configured from the client to the resources of the central network.

You may, however, want to permit bidirectional access (and the costs that might involve). This permission can be given in Setup Tool by adding the **Destination** of the IP address of the client PC in IP –> Network Address Translation –> ADD:

```
BinTec routerSetup Tool                          BinTec Communications AG
(IP)(NAT)(CONFIG)(ADD): NAT Configuration                        MyRouter

Service                     user defined
Protocol                    tcp
Port (-1 for any)           139


Destination                 172.16.100.99 (IP address of client PC)

Use <Space> to select
```

Proxies

In some cases NAT will not work when port numbers and/or IP addresses are transmitted in the data part of a TCP or UDP session. This is the case for some of the standard Internet Protocols (IP). To allow these protocols to work with NAT, so-called "proxies" have been implemented within the NAT software.

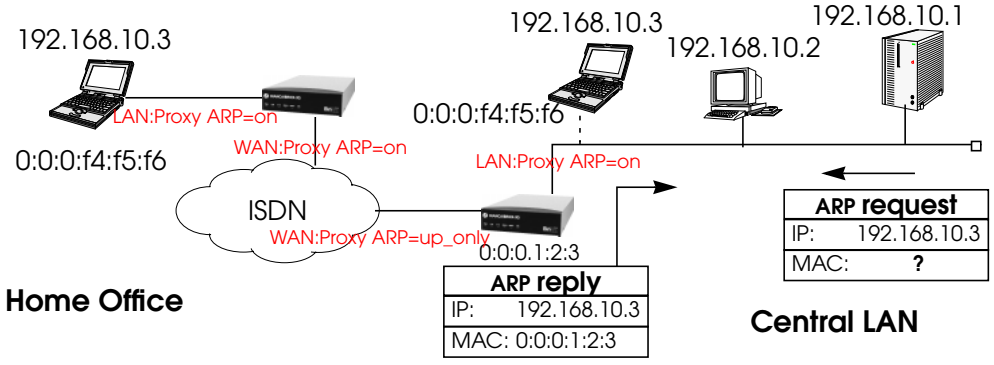
These proxies know how IP addresses and port numbers are transmitted within the data-portion of a connection. The proxy tracks the data sent/received, detects the addresses/port numbers used, and translates the information according to the NAT translation software. Currently, internal proxies have been implemented for the following services:

- FTP
- IRC
- Real Audio
- VDOLive Audio
- rlogin
- RCP
- rsh
- VDOLive Video

1.14.6 Proxy ARP

ARP (Address Resolution Protocol) is a technique used to map an IP address to a physical network address, or MAC address. Normally, ARP requests for the hardware address of a particular IP address are answered by the station the IP address is assigned to. With proxy ARP, the request can be

alternatively answered by the BinTec router. This is useful when a host on your network is connected via an ISDN line.



Our example above shows a setting, where a laptop is used in the Home Office and is connected to the LAN via ISDN, but may also be connected to the LAN directly. In this example ARP requests from the LAN for the laptop's physical address, are answered by the BinTec router, as long as the laptop is connected via ISDN. When the laptop is connected to the LAN, it answers ARP requests itsself.

To activate Proxy ARP it must be turned on for the LAN interface and the destination WAN interface, via which the requested IP address would be routed. The Proxy ARP settings for the WAN interface work in dependence of the operation state of the respective interface, when turned on (***on*** or ***up_only***). IP datagrams from the LAN destined for these hosts are sent directly to the BinTec router and are forwarded to the real host. The benefit of proxy ARP is that no routing entries need to be made for such hosts.

For the LAN interface the variable ***ipExtIfProxyArp*** (***ipExtIfTable***) can receive the values off and on:

- ***off***
  Proxy ARP is turned off, which is the default value.

- ***on***
  Proxy ARP is turned on.

In Setup Tool Proxy ARP for the LAN can be configured in the Advanced Settings for the LAN interface.

For the WAN interface the configuration of the variable ***ipExtIfProxyArp*** (***ipExtIfTable***) differs. When proxy ARP is turned on, ARP requests are answered in dependence of the ***ifOperStatus*** (***ifTable***) of the interface, via which the requested host can be reached. Possible values are ***off***, ***on*** and ***up_only***.

Values for ***ipExtIfProxyArp*** on the WAN interface:

- ***off***
  Proxy ARP is turned off, which is the default value.
- ***on***
  The request is only answered, when the WAN interface has the ***ifOperStatus up*** or ***dormant***. When the interface was in the state ***dormant***, a connection is setup after the ARP request.
- ***up_only***
  The request is only answered, when the WAN interface has the ***ifOperStatus up***. This value makes sense, when ARP requests should only be answered in case there is already an existing connection to the requested host.

In Setup Tool Proxy ARP for the WAN interface can be configured in the WAN Partner menu for the respective host in the Advanced Settings of the IP submenu.

The requirements for an answer to a ARP request from the LAN by the BinTec router are that the destination address would be routed to a different but the LAN interface and that on both interfaces (LAN and destination WAN interface) proxy ARP is turned on (***on*** for the LAN interface and

on or ***up_only*** for the respective WAN interface). Beyond that the ***ifOperStatus*** of the WAN interface must have the demanded state.

When you want to use Proxy ARP on a RADIUS interface, the variable ***ipExtIfProxyArp*** must be set via the BinTec-specific RADIUS attributes. On using BinTec-specific RADIUS attributes see the Extended Feature Reference available via the BinTec FTP server at http://www.bintec.de.

NOTE:

Proxy ARP may cause problems on systems that check for security violations where two IP addresses map to the same physical address.

1.14.7 RIP Options

RIP (Routing Information Protocol) is used by IP routers to learn of new IP routes (see RIP for a brief description). To enable the RIP on the BinTec router the ***biboAdmRipUdpPort*** field must be set to 520. This is the default setting and specifies the UDP-port to exchange RIP messages over. RIP can be disbled completely by assigning UDP port 0 to this variable.

The BinTec router supports both versions 1 and version 2 of RIP. Using the ***RipSend*** and ***RipReceive*** variables in the ***ipExtIfTable***, the BinTec router can be configured to separately send/receive either version, both versions or no RIP packets over selected interfaces. ***RipReceive*** defines the types of RIP packets that are accepted (will use for dynamically learning of new routes) over the interface.

ipExtIfRipSend can be assigned the values:

`ripV1`	Send only RIP V1 packets.
`ripV2`	Send only RIP V2 packets.

both Send a RIP V1 packet, and a V2 packet.
none Do not send RIP packets.

ipExtIfRipReceive can be assigned the values:
ripV Accept only RIP V1 packets.
ripV2 Accept only RIP V2 packets.
both Accept both versions.
none Ignore RIP packets.

1.14.8 Back Route Verify

ipExtIfBackRtVerify

This variable activates a check for incoming packets. If set to on, incoming packets are only accepted if return packets sent back to their source IP address would be sent over the same interface. Otherwise, the packets are silently dropped. This prevents packets being passed from untrusted interfaces to this interface.
Possible values: off (1), on (2)
Default value: off