



Personal Access (BINGO!)

Release Notes

System Software Release 5.1 Revision 1

October 28, 1999



New System Software

Release 5.1 Revision 1

This document describes the new features, changes, bugfixes and known issues made to the system software in Release 5.1 Revision 1 for the Personal Access product group (BINGO!).



From Release 5.1 Revision 1, Release Notes of Bintec's software images will be divided into three categories based on the three product groups: Corporate Access, which includes BIANCA/BRICK-XL/XL2 and BIANCA/BRICK-XMP; Workgroup Access, which includes BIANCA/BRICK-XM2 and BIANCA/BRICK-XS2; and Personal Access, including BinGO!

1	Upgrading System Software	7
2	New Features in Release 5.1.1	9
2.1	Windows Activity Monitor	9
2.1.1	Configure the BRICK	10
2.1.2	Windows Application	13
2.2	Extended WAN Partner License for BinGO!	13
2.3	Keepalive Monitoring	13
2.4	ISDN Channel Reservation	21
2.5	Improved Bandwidth On Demand	24
2.6	VPN and NAT	38
2.6.1	Constellation	39
2.6.2	What's new?	40
2.6.3	Configuration	42
2.7	NetBIOS over NAT	45
2.8	NetBIOS Node Type by DHCP	55
2.9	MS-Callback Termination Option	57
3	Changes/Improvements	59
3.1	PPP	59
3.1.1	Permanent Connection	59
3.2	IP	59
3.2.1	No Longer Automatic IP Conversion	59
3.3	CAPI	60
3.3.1	CAPI Info Syslog Message	60
3.4	System	60

3.4.1	No Autologout during Update	60
3.4.2	Trace Application Modification	61
3.4.3	Setup Tool SYSTEM Menu	61
4	Bug Fixes	63
4.1	Setup Tool	63
4.1.1	Setup Tool Crash after SAVE in Extended Interface Settings	63
4.1.2	CLID Configuration in Setup Tool Flawed	63
4.2	CAPI	63
4.2.1	CAPI via X.31 on D-Channel	63
4.2.2	Connection Delays with CAPI or a TPO Bridge	64
4.2.3	Data_B3_IND Data Handle Counted up to 2	64
4.2.4	CAPI and Incorrect Bearer Capability	64
4.2.5	Connection Difficulties with CAPI 1.1 Applications	64
4.2.6	Transmission Failure after Protocol Switching	65
4.3	ISDN	65
4.3.1	Connected Address with 1TR6 Leading to Connection Failure	65
4.4	System	65
4.4.1	Multiple Simultaneous Logins Jam BRICK	65
4.4.2	Number of Columns in the SNMP Shell	66
4.5	PPP	66
4.5.1	Channel Bundling and Dynamic Short Hold Malfunction	66
4.5.2	MS-Callback (CBCP) Working on the 2nd Attempt	66
4.5.3	Charging Amounts Logged Incorrectly	67
4.5.4	Link Quality Monitoring Set to "0"	67
4.5.5	Problems Accessing Compuserve for the First Time	67
4.5.6	Termination of VPN Connections	68
4.5.7	Tracing a VPN Connection	69
4.5.8	IPX Compression Protocol has not been Rejected	69

4.5.9	Link Quality Monitoring (LQM)	69
4.5.10	Microsoft Point-to-Point Compression (MPPC)	69
4.6	HTTP Status Page	70
4.6.1	Internet Explorer 4.0	70
4.7	IP	71
4.7.1	IP-Address via DHCP	71
5	Known Issues	72
5.1	Outgoing FTP Connections via NAT	72
5.2	BOD not Activated due to Load Error	72
5.3	Secure VPN and the BRICK	72

1 Upgrading System Software

- Retrieve the current system software image from BinTec's WWW server at <http://www.bintec.de> (Section: Download).
- With this image you can upgrade the BIANCA/BRICK with the `update` command from the SNMP shell via a remote host (i.e. using telnet, minipad, or isdnlogin) or by using the BOOTmonitor, if you are logged in directly on the console.
Information on using the BOOTmonitor can be found in the BIANCA/BRICK User's Guides under Firmware Upgrades.



Caution!

- Do not update your Logic or BOOTmonitor images unless expressly instructed to do so. Normally, it is not necessary to upgrade these images. Only in exceptional cases is an upgrade explicitly recommended.
- If you are unsure whether to upgrade or not, read the BOOTmonitor and Firmware Logic Release Notes (available below the images on the FTP server, section: Download) where you will find tables that specify the appropriate Logic and BOOTmonitor versions available for your BinTec product, and if an update is recommended or not.



Please note that there is an update procedure in case there is not enough memory available to perform a software update via the `update` command from the SNMP shell. The incremental update loads the new software image in blocks of 64 KB via TFTP and writes it to the Flash ROM immediately.

Because this procedure offers no possibility to check the integrity of the image:

➤ first use the option “-v” that verifies the image file.

- Once you've installed Release 5.1 Revision 1, you may want to retrieve the latest documentation (in Adobe's PDF format), which is also available from BinTec's WWW server at the address noted above.



When upgrading system software, it is also recommended that you use the most current versions of BRICKware for Windows and UNIX Tools. Both can be retrieved from BinTec's WWW server.



If you are updating from a software release equal to or older than 4.7.x to the current version, 5.1.1 or later, it is necessary to firstly upgrade to 4.9.3, save the configuration, then upgrade to 5.1.1. If you update directly from 4.7.x to 5.1.x, the old IP access lists can not be automatically converted and are lost.

2 New Features in Release 5.1.1

2.1 Windows Activity Monitor

With Release 5.1.1, you can configure a **BRICK** to be monitored by the new Windows Activity Monitor which will be available with the new BRICKware Release 5.1.1.

Why the Activity Monitor?

With the Activity Monitor, Windows users can monitor the activities of a **BRICK**. Important information like system status of physical interfaces (e. g. ISDN line) and virtual interfaces (e. g. WAN partners) are easily available with ONE tool. A clear and complete overview of the load of a **BRICK**'s interfaces is possible at any time. The following illustration shows the state of a CM-PRI interface.

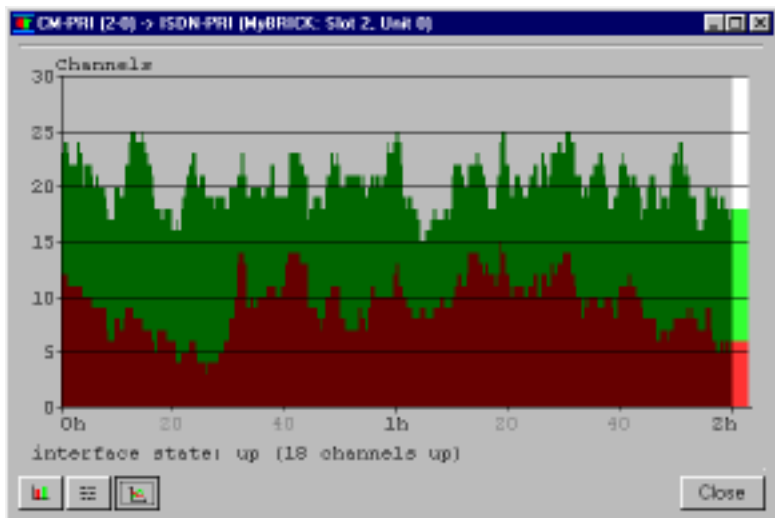


Figure 2-1: Activity Monitor display of a CM-PRI interface

How does it work?

A status daemon collects information about the **BRICK** and transmits it in the form of UDP packets to the LAN's broadcast address (default) or to a specified IP address. One packet per **BRICK** interface and time interval, which is individually adjustable from 1 to 60 seconds, is sent. All physical interfaces and up to 100 virtual interfaces can be monitored unless the packet size of approx. 4000 bytes is not exceeded. A Windows application on your PC, which will be available with the new BRICKware Release 5.1.1, receives the packets and displays the information in different ways.

To activate the Activity Monitor you have to

- configure the **BRICK(s)** to be monitored (this step is described in this Release Note)
- start the Windows application on your PC and use it (this step will be described in the new, updated **BRICKware for Windows**, available with the new BRICKware Release 5.1.1)

2.1.1 Configure the **BRICK**

You can perform the required configuration steps on the **BRICK** via

- MIB variables
- Setup Tool

MIB variables:

The configuration is made by entries in the MIB table **ExtAdmin** using the following MIB variables:

Variable	Meaning
ExtAdminMonPort	Number of port for the Activity Monitor (default: <i>2107</i> , registered by the IANA - Internet Assigned Numbers Authority).
ExtAdminMonAddress	IP address to which the BRICK sends the UDP packets. With the default value <i>255.255.255.255</i> the broadcast address of the first LAN interface is used. Be aware that if you enter the IP address of a WAN partner, connections liable to charges will be made at very regular intervals (default is every 5 seconds).
ExtAdminMonType	Type of information sent with the UDP packets to the Windows application. Possible values: <ul style="list-style-type: none"> ■ <i>off</i>: deactivates the Activity Monitor (default value) ■ <i>physical</i>: only information about physical interfaces ■ <i>physical_virt</i>: information about physical and virtual interfaces
ExtAdminMonUpdate	Update time in seconds. Possible values: <i>0</i> to <i>60</i> (default: <i>5</i>).

Table 2-1: **ExtAdmin**

To change the settings, make the following entry (as an example):

```
biboExtAdmMonAddress( rw):      192.168.1.1
biboExtAdmMonPort( rw):        2107
biboExtAdmMonType( rw):        physical_virt
biboExtAdmMonUpdate( rw):      5
```

Setup Tool

The configuration is made in **SYSTEM** ➤ **EXTERNAL ACTIVITY MONITOR**:

BRICK Setup Tool	BinTec Communications AG
[SYSTEM][ACTIVMON]:External Activity Monitor	MyBinGO!
Client IP Address	192.168.1.1
Client UDP Port	2107
Type	physical_virt
Update Interval (sec)	5
SAVE	CANCEL
Use <Space> to select	

Field	Meaning
Client IP Address	See ExtAdminMonAddress above.
Client UDP port	See ExtAdminMonPort above.
Type	See ExtAdminMonType above.
Update Interval (sec)	See ExtAdminMonUpdate above.

Table 2-2: **SYSTEM** ➤ **EXTERNAL ACTIVITY MONITOR**

Proceed as follows:

- Go to **SYSTEM** ➤ **EXTERNAL ACTIVITY MONITOR**.
- Enter **Client IP Address**, **Client UDP port**, **Type** and **Update Interval (sec)**.
- Press **SAVE**.

2.1.2 Windows Application

How to use the Activity Monitor on your PC will be described in the new, updated **BRICKware for Windows** (available with the new BRICKware Release 5.1.1).

2.2 Extended WAN Partner License for BinGO!

It is now possible to acquire a license to extend the number of WAN partners you want to configure. Thus, **BinGO!** users are no longer restricted to the previous number of four WAN partners. The license can not be activated, however, when running a system software image older than Release 5.1.1. For more details, consult your dealer.

2.3 Keepalive Monitoring

Keepalive Monitoring is a new feature that prevents unnecessary connections being made by a central server over the WAN to the router of a LAN. If that server regularly tries to transfer data to the router and its hosts, it may well happen that there are no computers turned on in the LAN and the calls made to the router as well as the costs incurred, of course, are in vain.

The solution to this problem is a monitoring system performed by the router which checks at regular definable intervals whether computers in the router's

LAN can be reached or not. By means of a new table, the **ipHostsAliveTable**, the state of specified IP addresses can be monitored by the router. The router tests the accessibility of the computers by pinging them. If, after three attempts, no computer in the LAN is reachable, PPP connections are deactivated by changing the state of the corresponding interface to *down*. If, on the other hand, at least one computer in the LAN responds to the ping, the state of the IP address is set to *alive* and the status of the corresponding interface (**ifAdminStatus**) is set to *up* and PPP connections can be made.

The reachability of the IP address is thus not initiated by the client PC itself. Therefore, the connection of router to client is not necessarily made immediately after the PC is turned on. The connection is made only after the router itself has established the reachability of the client, a process which is dependent on the monitoring times set.

WAN Partner Configuration

Essentially, Keepalive Monitoring oversees the accessibility of computers in a LAN and controls the states of the interfaces of WAN partners accordingly. An incoming call from a WAN partner can respond to the state of the interfaces or not, depending on how that WAN partner is configured. Let's look at some of the implications for the effectiveness of Keepalive Monitoring of different WAN partner configurations.

- WAN partners identifiable by CLID
If the WAN partner is configured and identifiable by CLID, the call is accepted if the state of that WAN partner's interface (**IfAdminStatus**) is *up*; the call is refused if the state of the interface is *down*. Thus, thanks to Keepalive Monitoring, no unnecessary connections are established.
- In the event of an inband call (non-CLID)
 - If just the one WAN interface is configured with non-CLID, the functionality of Keepalive is guaranteed. When the configured inband WAN partner calls, if the interface has been set to *up*, the call will be correctly taken as there is only one non-CLID interface to match the inband (non-CLID) call.

- If two or more WAN interfaces are configured with non-CLID and these interfaces are *up*, an inband call will be established although an unnecessary connection may be made. In such a case, the purpose of Keepalive Monitoring, i.e. to prevent unnecessary connections, is not served.
 - If ALL interfaces are down, an inband call is not established.
- Keepalive Monitoring has no effect for calls coming from partners configured on a RADIUS server. In this case, there is no static interface defined for that partner on the BRICK that can be influenced by the monitoring feature.

The MIB table variables in the new IP table (**ipHostsAliveTable**):

inx	Group (rw)	IPAddress (*rw)	State (ro)	Interval (rw)
	DownAction (rw)	FiirstIfIndex (ro)	Range (rw)	
00	0	123.124.125.126	down	300
	down	10002	0	

The meanings of the different variables from the ipHostsAliveTable:

Variable	Meaning
ipHostsAliveGroup	This is a group of IP addresses that can be monitored. The number of these groups is restricted to ten, the range of values spans from 0 to 9 (0 being the first group, 9 the last). IP addresses in the same group are combined with OR, which means that as long as at least one IP address is reachable in the group, the state of the group is set to <i>alive</i> . The state of the entire group is only set to <i>down</i> when none of the IP addresses in the group is reachable.
ipHostsAliveIPAddress	An IP address that is to be monitored or pinged by the router. The number of IP addresses per group is also restricted to 10, which means that, in total, up to 100 IP addresses can be monitored.
ipHostsAliveState	Here the state of the monitored IP addresses is maintained. The possible values are either <i>alive</i> or <i>down</i> . <i>Alive</i> if the IP address is reachable; <i>down</i> if the IP address is not reachable.

Variable	Meaning
ipHostsAliveInterval	<p>The interval calculated in seconds between which the IP addresses are pinged. The range of values spans from 1 to 65536. (The default value is set at 300 seconds = 5 minutes.)</p> <p>Each IP address is arranged in a group and is given an interval time. A monitoring time is then calculated for each group where the shortest interval of all is used. Then a global monitoring time is calculated using all the group values, again the shortest value is used. The global monitoring time is used to start the program, after booting it could take 2 minutes (if that is the shortest group interval) for the first ping to take place, during which the state of the IP addresses is set to 0. Hosts are then pinged according to the group monitoring time.</p> <p>If there is no reply, the state of the IP address can be reset after three pings. If there is a reply, the state of the IP address changes immediately.</p>
ipHostsAliveDownAction	<p>This variable defines the effect on the defined interfaces if ipHostsAliveState changes to the <i>down</i> state. The possible values are <i>up</i>, <i>down</i> or <i>delete</i> (<i>delete</i> does not affect the ifAdminStatus, but merely deletes the entries of that line in the MIB table), the default is down. The ifAdminStatus of the interface(s) is then changed according to the value in DownAction (see below).</p>

Variable	Meaning
ipHostsAliveFirstIfIndex	This is the first interface number in the index list that is to be affected by the execution of DownAction .
ipHostsAliveRange	This indicates the possible range of numbers of interfaces that can be affected by the execution of DownAction . So if you set FirstIfIndex to 10001 and Range to 0, only the interface 10001 is affected and none other. On the other hand, if, in the same case, you set Range to the default number 4999, all the interfaces with a number between 10001 and 15000 are affected by DownAction .

Table 2-3: **ipHostsAliveTable****Example of an entry in the new IP table (ipHostsAliveTable):**

inx	Group(rw) DownAction(rw)	IPAddress(*rw) FirstIfIndex(ro)	State(ro) Range(rw)	Interval(rw)
00	5 down	192.168.98.129 10003	alive 0	180

This means that the IP address *192.168.98.129* belongs to group 5. The interval between pings is set to 3 minutes. If this IP address is no longer reachable (and, if there are any, all other IP addresses in group 5 are also unreachable), the **ifAdminStatus** of the interface 10003 is set to the *down* state.

Possible states

Depending on the state of the group and what is specified in **DownAction**, the **ifAdminStatus** for the defined interfaces receive the following values:

Group State	AND	DownAction	=>	ifAdminStatus
<i>alive</i> (reachable)	+	<i>down</i>	=>	<i>ifAdminStatus_up</i>
<i>alive</i> (reachable)	+	<i>up</i>	=>	<i>ifAdminStatus_down</i>
<i>down</i> (not reachable)	+	<i>down</i>	=>	<i>ifAdminStatus_down</i>
<i>down</i> (not reachable)	+	<i>up</i>	=>	<i>ifAdminStatus_up</i>

Table 2-4: **ifAdminStatus**

As the table shows, when the group state is *alive* or *reachable*, the **ifAdminStatus** is set to the contrary of what is specified in **DownAction**. Conversely, when the group state is set to *down*, the **ifAdminStatus** conforms with the specification under **DownAction**.

One IP address in two groups?

If the same IP address is in two or more different groups with different monitoring times (can occur if the same IP address is allocated to different interfaces), it can happen that the group states of the one IP address are different, as well as the states of the respective interfaces.

For example: let's assume the same IP address appears in two groups. In the first of these groups, the monitoring time is set to 60 seconds, in the second to 300 seconds. If, in both cases, the IP address is not reachable, the state of the first group will be set to *down* after 3 x 60 seconds; the second, after 3 x 300 seconds. There could thus be an interim period when the state of group 1 is set

to *down* and the state of group 2 is set to *up*. This would mean in turn that the interface of the IP address in group 1 is set to *down* in the **ifAdminStatus**, the interface of the same IP address in group 2 is set to *up*.

However, once the IP address is *up* again, the state of the IP address changes in both groups immediately after the reply to the ping from the first group.

SysLog messages

The following is a list of relevant SysLog messages:

biboAdmSyslogMessage	Log Level
alivepacket sent to xxx.xxx.xxx.xxx (a packet has been sent to the following IP address xxx.xxx.xxx.xxx)	debug
xxx.xxx.xxx.xxx is alive (a packet has been received from the IP address xxx.xxx.xxx.xxx)	debug
no answer from xxx.xxx.xxx.xxx (after three ping attempts, no packet was received from the IP address xxx.xxx.xxx.xxx)	debug
only 10 entries per group are allowed (the user wanted to create more than ten entries in the one group. The entry last attempted is not included in the MIB table)	err
cannot send alivepacket to xxx.xxx.xxx.xxx (a packet could not be sent to the IP address xxx.xxx.xxx.xxx)	err
interface xxxxx set [<i>up</i> <i>down</i>] (the interface numbered xxxxx was set to the <i>up</i> or <i>down</i> state)	info

Table 2-5: **biboAdminSyslogMessage**

Changing the state of the interface over the SNMP shell

If the state of an interface is changed by the user in the SNMP shell, the program does not register the change. The following pattern of behaviour is possible:

If the state of the IP address is *down* and **DownAction** is also set to *down*, but the user manually sets the **ifAdminStatus** to *up* in the SNMP shell, the interface status remains *up* even if, subsequent to the next three pings, the program has established that the state of the IP address is *down*. The interface status will remain *up* as long as the IP address remains *down*.

However, once the IP address responds to a ping (i.e. the IP address is reachable and its state is thus set to *alive*), the program once again has control over the interfaces. Although this means that nothing changes in this case, i.e. the **ifAdminStatus** remains *up*, the next time this IP address can not be reached and its state is set to *down*, the **ifAdminStatus** of the corresponding interface is also set to *down*.

2.4 ISDN Channel Reservation

In order to give you even more control over the number and direction of your calls, the **isdnCreditsTable** has been extended. Up to now it was already possible to specify the maximum number of allowed incoming connections over a certain definable period of time (**isdnCreditsMeasureTime**). Once these limits were reached, they could not be exceeded and no more calls could be made in the direction in which the limitation was set. If a hundred outgoing calls could be made over the period of one day, in theory all those calls could be made in the first hour. This would have meant that during that hour it may have been very difficult to receive any incoming calls at all.

Now, however, this new feature allows you to set the maximum number of incoming calls, outgoing calls, as well as the total number of calls being made at the current moment in time.

Example 1:

This means that if it is very important for you that at least half of your PRI B-channels (30 in total) remain open for incoming calls, you can set one of the three new variables, **MaxCurrentOutCon** (the maximum number of outgoing calls), to 15, this would mean that at least 15 channels would be reserved for incoming calls.

Example 2:

You may only have two B-channels available. In such a case, it might be prudent to set the maximum number of outgoing calls to 1 as well as the maximum number of incoming calls to 1. You could thus limit the likelihood of missing any important incoming calls. It would, however, mean that, if one outgoing call were being made, an attempt to make a second outgoing call would not be possible.

Finally, you can limit the total number of both incoming and outgoing current connections. You could specify that only twenty of your B-channels are to be made available at any one given time. This last option can only be configured directly in the **isdnCreditsTable** not, however, over Setup Tool.

Configuration

The three new variables that have been added to the **isdnCreditsTable** can be configured in the MIB using the SNMP shell:

Variable	Meaning
MaxCurrentInCon	This variable allows you to set the maximum number of current incoming connections
MaxCurrentOutCon	This variable allows you to set the maximum number of current outgoing connections.
MaxCurrentCon	This variable allows you to set the maximum number of incoming as well as outgoing calls currently being made.

Table 2-6: **isdnCreditsTable**

MaxCurrentInCon and **MaxCurrentOutCon** can be configured in Setup Tool in the menu **ISDN ► CREDITS ► EDIT**.

BRICK Setup Tool	BinTec Communications AG
[ISDN][CREDITS][EDIT]: Configure ppp Credits	MyBinGO!
Surveillance	on
Measure Time (sec)	6400
Maximum Number of Incoming Connections	off
Maximum Number of Outgoing Connections	on
Maximum Charge	100
	off
Maximum Time for Incoming Connections (sec)	on
	28800
Maximum Time for Outgoing Connections (sec)	on
	28800
Maximum Number of Current Incoming Connections	on
	1
Maximum Number of Current Outgoing Connections	on
	1
SAVE	CANCEL
Enter integer range 0..2147483647	

- Activate **Maximum Number of Incoming/Outgoing Connections** by pressing the Spacebar and changing *off* to *on*.
- Enter the number of B-channels you want to reserve for that direction.
- Press **SAVE**.

In the above example, the two new variables at the bottom of the table are both set to one. This means that it is impossible to make more than one incoming or outgoing call at any one given time.

Surveillance

It is also possible to observe the number of connections currently being made by going to **MONITORING AND DEBUGGING ► ISDN CREDITS ► SELECT SUBSYSTEM**.

BRICK Setup Tool		BinTec Communications AG	
[Monitor][CREDITS][STAT]: Monitor ppp Credits		MyBinGO!	
	Total	Maximum	%reached
Time till end of measure interval (sec)	84400	86400	2
Number of Incoming Connections	1		1
Number of Outgoing Connections	1		1
Time of Incoming Connections	734		
Time of Outgoing Connections	244		
Charge	0		
Number of Current Incoming Connections	22	22	
Number of Current Outgoing Connections	6	12	
Number of Current Connections	28	28	
Exit			

In this case, the new variables show that twenty-two incoming calls are being made out of a possible total of twenty-two, only six of the permitted twelve outgoing calls are being made, however, because the maximum number of current connections is restricted to twenty-eight.

2.5 Improved Bandwidth On Demand

Dynamic channel bundling of leased lines and dialup lines

With Release 5.1 Revision 1, Bandwidth On Demand, BOD, over different media is now possible with the BIANCA/BRICK router family.

The bundling of leased lines with dialup lines gives you more control over the switching thresholds and times, bandwidth and costs of your connections.

You can now bundle channels over a new optional table, the **pppExtIfTable**, which can also be configured over Setup Tool, see [Setup Tool configuration, page 31](#) below. 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 algorithm used by which the load of your leased line or bundle is calculated or the number of seconds such a sample calculation should take, giving you more control over the switching of dialup lines.

Bandwidth On Demand for leased lines

In order to support high levels of traffic over leased lines, dynamic switching of dialup lines is now possible.

BOD is activated by setting the **pppExtIflBodMode** 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 (90% or more for 5 seconds to switch on and less than 80% of the remaining links for 10 seconds to switch off, see [Switchover thresholds for BOD, page 35](#) below).

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, as was necessary in the past. Now IPX backup configuration is also possible.

BOD 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 backup connection.

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

```
inx  Index (*rw)      BodMode (-rw)      Algorithm (rw)
     Interval (rw)   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	<p>Contains the following values:</p> <p><i>BOD_backup</i></p> <p>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 biboPPMaxConn is set to more than 1.</p> <p><i>BOD_active</i></p> <p>Only one partner should be configured as the active partner. Switching on and off of additional B-channels then only occurs from this side.</p> <p><i>BOD_passive</i></p> <p>No switching on and off of additional B-channels occurs from the side of the partner configured as <i>BOD_passive</i>. He participates as a passive partner in the channel bundle.</p> <p><i>BOD_disabled</i></p> <p>No extra channels are opened to support leased lines.</p>

Variable	Meaning
Algorithm	<p>An algorithm for the weighting of throughput within a specified time frame for the calculation of capacity utilization (load). The following values are possible:</p> <p><i>equal</i>: 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.</p> <p><i>proportional</i>: 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.</p>
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 algorithm and used to determine whether another line should be switched either on or off.

Variable	Meaning
MlpFragmentation	<p>A mode according to which MLP fragments are formed.</p> <p><i>proportional</i>: 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.</p> <p><i>equal</i>: 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.</p>
MlpFragSize	<p>Here you can configure the size of the fragment to be sent. If you have left the setting <i>proportional</i> 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 <i>equal</i> above, the number you configure represents the size of the fragments.</p>

Table 2-7: PPPExtIfTable

MlpFragmentation and **MlpFragSize** can not be configured in Setup Tool and should be left with the default values which are *proportional* and 50 respectively.

Setup Tool configuration

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:

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

BIANCA/BRICK-X21 Setup Tool		BinTec Communications AG
[WAN]: WAN Partners		mybrick
Current WAN Partner Configuration		
Partnername	Protocol	State
Leased, Slot 2 (0)	ppp	down
Partner_1	ppp	down
xi2	ppp	up
ADD	DELETE	EXIT
Press, to scroll, tag/untag DELETE, to edit		

- 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	

- 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	

➤ Select **Extended Interface Settings**

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 Channels	2
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.

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 **WAN PARTNER** ►

EDIT ► PPP. 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.

Switchover thresholds for BOD

BOD is activated by setting the **pppExtlfbodMode** 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 **pppExtlfbodLoad** 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 **pppExtlfbodLoad** 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.

- If this value drops below 80 (%) for at least 10 seconds, a B-channel is switched off.
- 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.

- 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.

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.

A tabular representation of 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
VJHeaderComp	Supported
Layer2Mode	Not supported
DynShortHold	Supported

Table 2-8: **biboPPPTable**

2.6 VPN and NAT

A typical application of VPN (Virtual Private Networks) with PPTP (Point-to-Point Tunneling Protocol) consists of using the Internet to make a connection to the headquarters. To do this, clients establishing those links use a connection to a local ISP (Internet Service Provider) with dynamic IP address assignment or static IP address. The **BRICK** establishes the VPN tunnel, the ISP is not involved. These constellations are called symmetric PPTP, the **BRICK** combines the functions of PAC (PPTP Access Concentrator) and PNS (PPTP Network Server).

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

Up to now BinTec's NAT implementation has supported connections with the protocols ICMP, TCP and UDP, but not with GRE (Generic Routing Encapsulation). All VPN connections using the protocol GRE for transport, e. g. PPTP and L2TP (Layer 2 Tunneling Protocol) could not be switched through a BinTec router when NAT was activated on the WAN interface.

With Release 5.1.1 BinTec's NAT implementation allows forwarding GRE packets to a specified endpoint. So it is possible now to establish VPN connections even with NAT.

This description is based on the example VPN LAN-LAN configuration from the **Extended Feature Reference**, which can be retrieved from BinTec's WWW Server at <http://www.bintec.de> (Section: Download). In this Release Note the required additional configuration steps, when activating NAT on the WAN interface, are explained. The following points will be covered:

- What could a typical constellation look like?
- What's new with Release 5.1.1?
- What additional configuration steps are required?

2.6.1 Constellation

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

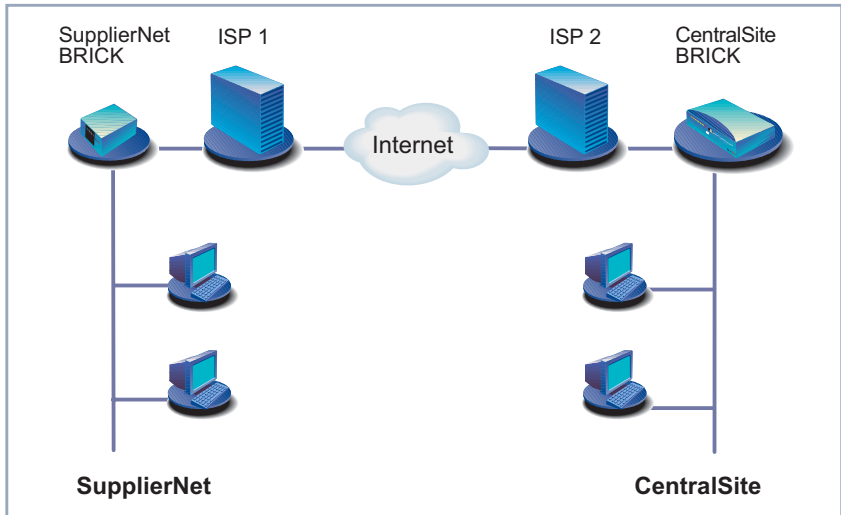


Figure 2-2: LAN-LAN connection using a VPN tunnel

Establishing and using a VPN connection with PPTP (see [figure 2-3, page 40](#) below) requires two protocols between the two tunnel endpoints – TCP (over PPTP) and GRE (over IP):

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

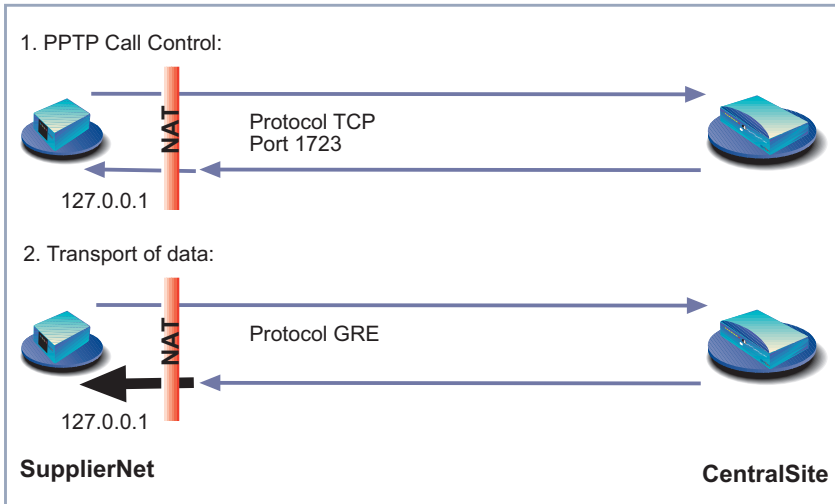


Figure 2-3: Establishing and using a VPN tunnel with PPTP (the changes in Release 5.1.1 allow incoming GRE packets to switch through the NAT firewall)

2.6.2 What's new?

BinTec Communications AG offers the following improvements to enable VPN connections using NAT:

Setup Tool

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

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

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

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

Table 2-9: **IP** ► **NETWORK ADDRESS TRANSLATION** ► **RETURN** ► **ADD**

MIB variables

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

The following values are available now:

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

Table 2-10: **IpNatTable**

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

Table 2-11: **IpNatPresetTable**

2.6.3 Configuration

The configuration of NAT for a VPN connection can be done

- via Setup Tool
- via MIB variables

In this Release Note the configuration is described via Setup Tool.



For the configuration of the VPN connection ([figure 2-2, page 39](#)), see **Extended Feature Reference**.

Client with dynamic IP address assignment

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



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

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

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

Client with static IP address

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



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

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

- Go to **IP** ➤ **NETWORK ADDRESS TRANSLATION**
- Select the interface to be configured for NAT (i. e. the interface to the ISP) and press **Return**

- Select **Network Address Translation**: *on*
- Press **ADD**
- Select **Service**: *user defined*
- Select **Protocol**: *gre*
- Enter **Port (-1 for any)**: *-1*
- Enter **Destination**: *127.0.0.1*
- Press **SAVE**
- Press **ADD**
- Select **Service**: *user defined*
- Select **Protocol**: *tcp*
- Enter **Port (-1 for any)**: *1723*
- Enter **Destination**: *127.0.0.1*
- Press **SAVE**

Central Site with static IP address

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

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

- Press **ADD**
- Select **Service**: *user defined*
- Select **Protocol**: *tcp*
- Enter **Port (-1 for any)**: *1723*
- Enter **Destination**: *127.0.0.1*
- Press **SAVE**



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



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

Testing the configuration

To test the VPN connection, e. g. with the `ping` command:

- do it from a host in your LAN to a host in the partner LAN or
- if you want to test from a **BRICK** to a partner **BRICK**, enter the **BRICK**'s LAN IP address as **Unique Source IP Address** in Setup Tool menu **IP** ➤ **STATIC SETTINGS** before testing. Otherwise, the **BRICK** will put the IP address of the WAN interface as source address into the ping packets originated by the **BRICK** and the consequence is that outgoing packets to the VPN partner are sent through the tunnel, but can only be returned outside the tunnel (on the underlying WAN connection).

2.7 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

Release 5.1.1 now allows that remote BinTec routers can be configured with NAT for WAN connections with NetBIOS over IP. The BRICK 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 shows, packets returning

from the WAN partner are sent to all computers in the LAN with a subnet broadcast address.

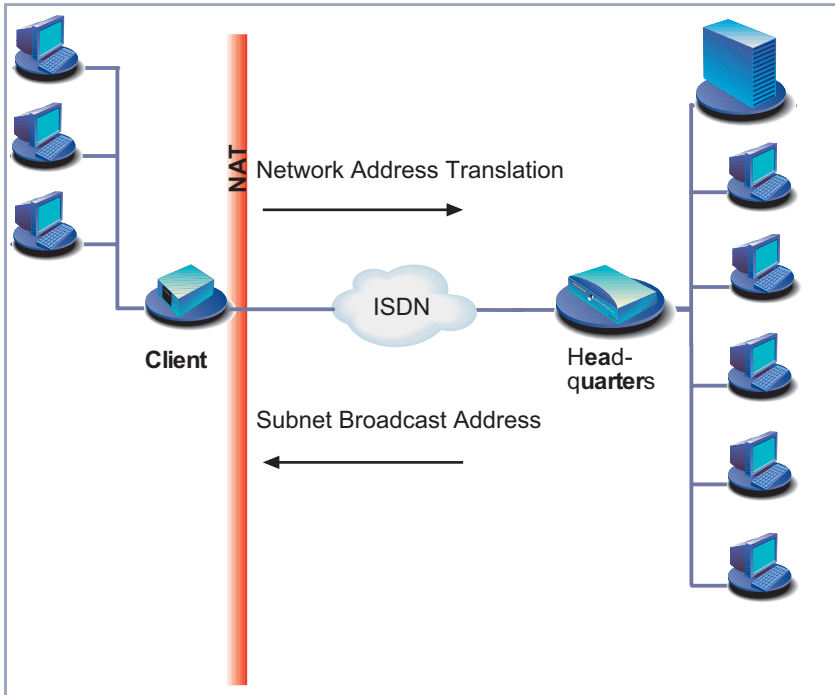


Figure 2-4: NetBIOS over NAT

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, proceed as follows:

- Go to **IP** ➤ **NETWORK ADDRESS TRANSLATION**

```
BRICK 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:

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**:

BRICKSetup Tool		BinTec Communications AG	
[IP][NAT][CONFIG][ADD]: NAT Configuration		MyBinGO!	
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:

```
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
```

Determining a subnet broadcast address



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>

In order to determine the broadcast address, you need to know the IP address of the **BRICK** 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 por-

tion 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:

3. If the bit in the mask is ON (=1), the respective bit in the IP address belongs to the network portion.
4. 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 ad-

dress, 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.

Figure 2-5 is an illustration of the sequence of commands leading to the calculation of a subnet broadcast address outlined above.

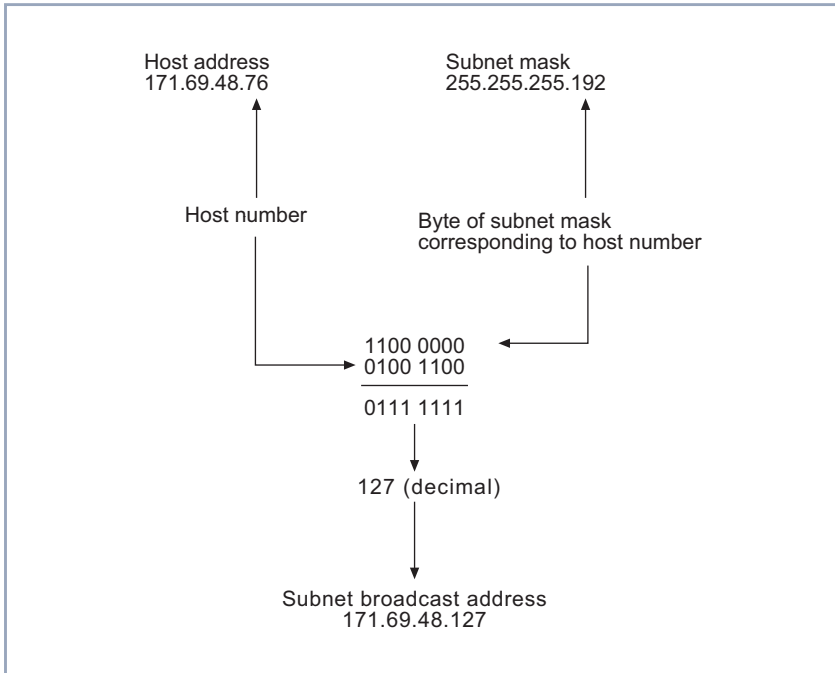


Figure 2-5: Determining a subnet broadcast address

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
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

Table 2-12: Three classes of IP addresses and broadcast addresses

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:**

BRICKSetup Tool	BinTec Communications AG
[IP][NAT][CONFIG][ADD]: NAT Configuration	MyBinGO!
Service	user defined
Protocol	tcp
Port (-1 for any)	139
Destination	172.16.100.99 (IP address of client PC)
Use <Space> to select	

2.8 NetBIOS Node Type by DHCP

Up to now the only way to define the NetBIOS node type was to set it directly in the registry under Windows 95/98/NT, now, however, configuration (via DHCP) on the BRICK is possible. The **ipDhcpTable** has been extended by the **ipDhcpNodeType** variable. This 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.

Methods of name resolution

■ 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

Configuration

It is now 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 BRICK 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:

- 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.
- 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:

- 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.
- 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.

3 Changes/Improvements

3.1 PPP

3.1.1 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 now 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 only be made in the MIB table and not in 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*.

3.2 IP

3.2.1 No Longer Automatic IP Conversion

If you are updating from a software release equal to or older than 4.7.x to the current version, 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.

3.3 CAPI

3.3.1 CAPI Info Syslog Message

Applications that started 30 CAPI listener(s) and received an incoming call also received very many Syslog messages, creating a considerable load. For this reason, the classification of the Syslog message "setup ignored" has been changed from "info" to "debug".

3.4 System

3.4.1 No Autologout during Update

In the past, it may have been necessary to disable autologout for the course of an update or to reset it to a higher value if a rather low time interval was set.

Now, however, autologout can not interrupt the installation of an image even if the autologout time interval is less than the time it takes to update.

Regardless of the value for autologout prior to the update, once the update is complete, autologout is reset to its default value of 900 or 15 mins. If, however, you want to change back to the value set prior to the update, type `t` in the SNMP shell and your old value will be reset.

3.4.2 Trace Application Modification

The output of the trace application has been slightly improved. The facilities traced in the D-channel can now be directly compared to their respective ASN.1 specifications.

3.4.3 Setup Tool *SYSTEM* Menu

With the implementation of the new Windows Activity Monitor feature, the Setup Tool *SYSTEM* menu has changed. This new feature is a surveillance tool for Windows users to oversee the functions of a **BRICK**, and is described above in [Setup Tool, page 12](#).

The following changes have been made:

- The *SYSTEM* ► *EXTERNAL ACTIVITY MONITOR* submenu has been added.
- The password settings are made in the new submenu *SYSTEM* ► *PASSWORD SETTINGS*.

The **SYSTEM** menu now appears as shown in the example below.

BRICK Setup Tool	BinTec Communications AG
[System]: Change System Parameters	MyBinGO!
System Name	brick
Local PPP ID (default)	bricklocal
Location	Country
Contact	BinTec
Syslog output on serial console	no
Message level for the syslog table	info
Maximum Number of Syslog Entries	20
External Activity Monitor >	
External System Logging >	
Password Settings >	
SAVE	CANCEL
Enter string, max length = 34 chars	

4 Bug Fixes

4.1 Setup Tool

4.1.1 Setup Tool Crash after SAVE in Extended Interface Settings

Running Release 4.9.5, if, in the Setup Tool menu **WAN PARTNER** ► **ADD**, no new WAN partner entry was made and one then went to the menu **ADVANCED SETTINGS** ► **EXTENDED INTERFACE SETTINGS** leaving via **SAVE**, Setup Tool crashed and if serially connected led to a stack trace.

This bug has been fixed.

4.1.2 CLID Configuration in Setup Tool Flawed

If two different numbers were configured for the one WAN partner with the **Directions both** and **outgoing** and if **LAPB Framing (only IP)** was selected under **Encapsulation**, the menu could not be exited via **SAVE**. Despite the fact that **Calling Line Identification** was set to **Yes**, the error message "**You must use Calling Line Identification (Direction both or incoming)**" was displayed.

This bug has been fixed.

4.2 CAPI

4.2.1 CAPI via X.31 on D-Channel

With an already existing CAPI connection via X.31 on the D-channel while other ISDN links were established and terminated, it could have occurred that CAPI

messages which had to establish the B-channel connection were delivered to the X.31 application. This could have led to the termination of the links.

This bug has been fixed.

4.2.2 Connection Delays with CAPI or a TPO Bridge

When there were particularly heavy loads of data traffic, transmission on the TCP connection may have been delayed by up to 15 seconds.

This bug has been fixed.

4.2.3 Data_B3_IND Data Handle Counted up to 2

Since Release 4.9.3 a valid Data Handle has been assigned. This was, however, counted up to 2 instead of 1.

This bug has been fixed.

4.2.4 CAPI and Incorrect Bearer Capability

Incoming CAPI calls, e.g. GSM calls, with incorrect bearer capability were not signalled to the CAPI application. When the bearer capability contained additional bytes to that contained in the CAPI specifications, no CIP value was recognised.

This bug has been fixed.

4.2.5 Connection Difficulties with CAPI 1.1 Applications

In some cases, when using CAPI 1.1 applications with Releases 4.9.1 and 4.9.3, outgoing connections could not be properly established.

This bug has been fixed.

4.2.6 Transmission Failure after Protocol Switching

There are a few fax and voice applications that use the Select_B_Protocol_REQ message to switch between protocols for fax and voice. Data could not be transmitted after this kind of switching.

This bug has been fixed.

4.3 ISDN

4.3.1 Connected Address with 1TR6 Leading to Connection Failure

In very few applications, if the presentation indicator was set in the connected address and the 1TR6 Protocol was being used, the connection could not be made. The 1TR6 does not allow the forwarding of the presentation indicator in the connected address in the D-channel.

Now, however, the connected address is filtered out by the BRICK and the call can be made.

4.4 System

4.4.1 Multiple Simultaneous Logins Jam BRICK

When a lot of people logged in over isdnlogin, it occurred that after a period of time no further logins were possible, and that no further processes could be started on the BRICK.

This bug has been fixed.

4.4.2 Number of Columns in the SNMP Shell

When the number of columns (i.e. each vertical line of character entries) was set to a value which fell below the number of characters required by the longest string using the command `u`, the SNMP shell crashed.

This bug has been fixed.

4.5 PPP

4.5.1 Channel Bundling and Dynamic Short Hold Malfunction

The channels of a bundle were disconnected at the end of each charging unit and then immediately reconnected, provided the load was high enough. This malfunction did not result in increased costs as the charging units were exploited to the full; there may, however, have been a momentary collapse in bandwidth.

This bug has been fixed.

4.5.2 MS-Callback (CBCP) Working on the 2nd Attempt

The MS-Callback Control Protocol initialized from a Windows client to the **BRICK** was known not to work under a combination of the following conditions:

- on the first attempt after a reboot of the **BRICK**
- no entry in the **biboDialTable**, which implies the "User Defined Numbers" mode.

This bug has been fixed.

4.5.3 Charging Amounts Logged Incorrectly

If the variables **biboPPPTotalCharge** and **biboPPPConnCharge** were set correctly and no value was given to the **biboPPPLinkCharge**, incorrect charging amounts in the syslog message (INFO-Level) may have occurred. If the problem occurred, the amount logged was higher than the real value.

This bug has been fixed.

4.5.4 Link Quality Monitoring Set to "0"

Concerning leased lines, when the variable **biboPPPLQMonitoring** in the **biboPPPTable** was set to the invalid value '0' instead of "off (01)", certain SNMP managers experienced problems.

This bug has been fixed.

4.5.5 Problems Accessing Compuserve for the First Time

If the **BRICK** was newly configured, used as DHCP server on the LAN and you had configured Compuserve as your ISP combined with encapsulation **x.75_PPP** or **x.75_BTX_PPP**, it was not possible to use a browser to establish a connection with your provider for the very first time.

In such a case, the router could not find a DNS server, necessary for connections using a name-based browser. Only connections to partners with PPP encapsulation were made and not to partners with **x.75_PPP** or **x.75_BTX_PPP**, as in the case of, for example, Compuserve access.

This bug has been fixed.

4.5.6 Termination of VPN Connections

If the **BRICK** is used to create a connection over a VPN tunnel to a PPTP-PNS (PPTP Network Server) via a PPTP-PAC (PPTP Access Concentrator), the **BRICK** typically has to perform two PPP authentication negotiations:

- an initial PPP negotiation with the PPTP-PAC
- a final PPP negotiation with the PPTP-PNS

If both PPP negotiations are successfully performed, a VPN connection between **BRICK** and PPTP-PNS is established.

It could have occurred that the **BRICK** terminated the link to PPTP-PAC and therewith the VPN connection to the PPTP-PNS. The following actions were performed:

1. The **BRICK** started an initial PPP negotiation with the PPTP-PAC.
2. If negotiated so, the **BRICK** started PPP authentication (for example by using PAP) with the PPTP-PAC sending a PAP authentication request.
3. The PPTP-PAC ignored this PAP authentication request, but established a VPN tunnel to the PPTP-PNS.
4. A new PPP negotiation between **BRICK** and PPTP-PNS was performed, but the **BRICK** did not terminate the PPP authentication negotiation with the PPTP-PAC as required in RFC 1661. The **BRICK** still tried to carry out step 2.
5. The **BRICK** successfully authenticated on the PPTP-PNS.
6. The **BRICK** tried to authenticate on the PPTP-PAC ten times. These trials failed and so the **BRICK** terminated the link to the PPTP-PAC and therewith the VPN connection to the PPTP-PNS.

This bug has been fixed, now the **BRICK** acts as required in RFC 1661.

4.5.7 Tracing a VPN Connection

Tracing the establishment phase of a VPN (PPTP) connection in a WAN as well as in a LAN may have caused a panic, leading to the termination of the trace. In very isolated cases, the BRICK rebooted.

This bug has been fixed.

4.5.8 IPX Compression Protocol has not been Rejected

The **BRICK** does not support IPX Compression Protocol presently. But if a WAN partner offered the option "IPX Compression Protocol" when performing the IPXCP negotiation, the **BRICK** did not reject it with an IPXCP Configure Reject.

This bug has been fixed.

4.5.9 Link Quality Monitoring (LQM)

If Link Quality Monitoring (LQM) had been successfully negotiated, a Link Quality Report (LQR) could be sent after the Link Control Protocol (LCP) was terminated but the ISDN B channel was still established. In consequence, the **BRICK** sometimes received an LCP Protocol Reject on this LQR which resulted in a restart of the **BRICK**.

This bug has been fixed.

4.5.10 Microsoft Point-to-Point Compression (MPPC)

If Microsoft Point-to-Point Compression (MPPC) was configured and successfully negotiated (MPPC is supported by BinTec's FM-STAC module with Hardware Release 2.1), the transmission of data which is difficult to be compressed could lead to inconsistent states of both partner's compression histories. Espe-

cially using Multilink PPP connections, termination of seeming redundant B-channels could occur as a result of stagnant or cancelled data transmission. In rare cases this could result in the failure of the FM-STAC module or the restart of the **BRICK**.

Additionally configured Microsoft Point-to-Point Encryption (MPPE) increased the problem.

This bug has been fixed.

4.6 HTTP Status Page

4.6.1 Internet Explorer 4.0

Using a **BRICK**'s HTTP status page with Internet Explorer 4.0, it was not possible to access sites restricted by password. For example, if one tried to show the MIB tables by clicking the link system tables, there occurred not the usual log-on dialog but the error message "401Unauthorized".

The reason is that the **BRICK**'s HTTP server did support the HTTP protocol versions 0.9 and 1.0, but not the latest version 1.1. As the Internet Explorer 4.0 uses the version 1.1, the HTTP server realized that it is not its latest protocol version (1.0) and therefore used its old version (0.9), which doesn't support authentication processes.

This bug has been fixed.

If one uses Internet Explorer 4.0 and doesn't want to upgrade the **BRICK** to Release 5.1.1, the following workaround is possible:

- Go to **View** ➤ **Internet Options**.
- Click the **Advanced** tab.
- Deactivate the option **Use HTTP 1.1**, click **OK**.

4.7 IP

4.7.1 IP-Address via DHCP

It could have occurred that assigning IP-addresses via DHCP in your LAN failed, for example, with the HP JetDirect print server. The reason was an invalid string length of the "Hostname" tag, i.e. there was one byte too many.

This bug has been fixed.

5 Known Issues

5.1 Outgoing FTP Connections via NAT

When outgoing FTP connections occur via NAT, data transfer does not work with some FTP servers. The connection is established, the FTP client can register with the server. Commands such as `cd` and `pwd` work, but others such as `dir` and `get` do not.

The problem can be dealt with if the client is switched to the passive mode. This is not, however, possible with all FTP clients.

5.2 BOD not Activated due to Load Error

When calculating the load of a channel bundle, no load (**pppExtIflLoad**) is calculated when **pppExtIflAlgorithm** is set to *proportional*. This means that BOD can not be activated when using this setting.

In order to ensure a load is calculated and BOD works correctly, make sure to leave **pppExtIflAlgorithm** set to the default value *equal*.

5.3 Secure VPN and the BRICK

The new Microsoft authentication procedure MSCHAP V2 included in Windows NT 4.0 Service Pack 4, the post-SP3 hotfix, and Windows 95 Dial-Up Networking 1.3 Upgrade is not yet supported by Bintec. This new protocol affects only VPN connections, not, however, dial-up lines. Thus, authentication on VPN connections using these upgrades or the hotfix may be affected. By making the following entries in the Windows Registry (if the entries do not already exist), VPN connections between Windows NT/95/98 servers and workstations can be ensured:

Windows NT

When the value below is set to one, the client is forced to use MSCHAP V2 for all VPN connections. When the default value of zero is set, authentication of VPN connections can take place using MSCHAP (or CHAP or PAP).

- Ensure the default value is set.

HKKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\RasMan\PPP

DWORD: SecureVPN

Value: 0x00000001 == force MSCHAP V2 for VPN connections

Value: 0x00000000 == do not force secure MSCHAP V2 (default)

Service Pack 4 or post-SP3 hotfix also includes a historyless mode for encryption and compression over PPTP connections.

- If you want normal MPPE compression and encryption negotiation, set the value below to "Disabled".

HKKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\NdisWan\Parameters

DWORD: Historyless

Value: 0x00000001 == Enabled (default)

Value: 0x00000000 == Disabled

Windows 95/98

For Windows 95/98 users, MSCHAP V2 could be upgraded with Windows 95 Dial-Up Networking 1.3 Upgrade. The same advice as above applies here, i.e. do not force secure mode.

HKKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\RemoteAccess

DWORD: SecureVPN

Value: 0x00000001 == Force secure mode (MSCHAP V2 plus data encryption) on all PPTP connections

Value: 0x00000000 == Do not force secure mode on PPTP connections (default)

For more information see:

<http://support.microsoft.com/support/kb/articles/q154/0/91.asp>

<http://support.microsoft.com/support/kb/articles/q189/5/94.asp>

<http://support.microsoft.com/support/kb/articles/q189/5/95.asp>

