

## The PPP NetBIOS Frames Control Protocol (NBFCP)

### Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

### Abstract

The Point-to-Point Protocol (PPP) [1] provides a standard method for transporting multi-protocol datagrams over point-to-point links. PPP defines an extensible Link Control Protocol, and proposes a family of Network Control Protocols for establishing and configuring different network-layer protocols.

The NBF protocol [3] was originally called the NetBEUI protocol. This document defines the Network Control Protocol for establishing and configuring the NBF protocol over PPP.

The NBFCP protocol is only applicable for an end system to connect to a peer system or the LAN that peer system is connected to. It is not applicable for connecting two LANs together due to NetBIOS name limitations and NetBIOS name defense mechanisms.

### Table of Contents

1.	Introduction .....	2
1.1	Specification of Requirements .....	2
1.2	Terminology .....	3
2.	A PPP Network Control Protocol for NBF .....	3
2.1	Sending NBF Datagrams .....	4
2.2	Bridging NBF Datagrams.....	5
2.3	NetBIOS Name Defense.....	5
3.	NBFCP Configuration Options .....	6
3.1	Name-Projection.....	6
3.2	Peer-Information.....	8
3.3	Multicast-Filtering.....	10
3.4	IEEE-MAC-Address-Required.....	11
	SECURITY CONSIDERATIONS .....	12
	REFERENCES .....	12

ACKNOWLEDGEMENTS .....	13
CHAIR'S ADDRESS .....	13
AUTHOR'S ADDRESS .....	13

## 1. Introduction

PPP has three main components:

1. A method for encapsulating multi-protocol datagrams.
2. A Link Control Protocol (LCP) for establishing, configuring, and testing the data-link connection.
3. A family of Network Control Protocols for establishing and configuring different network-layer protocols.

In order to establish communications over a point-to-point link, each end of the PPP link must first send LCP packets to configure and test the data link. After the link has been established and optional facilities have been negotiated as needed by the LCP, PPP must send NBFCP packets to choose and configure the NBF network-layer protocol. Once NBFCP has reached the Opened state, NBF datagrams can be sent over the link.

The link will remain configured for communications until explicit LCP or NBFCP packets close the link down, or until some external event occurs (an inactivity timer expires or network administrator intervention).

### 1.1. Specification of Requirements

In this document, several words are used to signify the requirements of the specification. These words are often capitalized.

- |          |   |
|----------|---|
| MUST     | This word, or the adjective "required", means that the definition is an absolute requirement of the specification.  |
| MUST NOT | This phrase means that the definition is an absolute prohibition of the specification.  |
| SHOULD   | This word, or the adjective "recommended", means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and carefully weighed before choosing a different course. |

MAY            This word, or the adjective "optional", means that this item is one of an allowed set of alternatives. An implementation which does not include this option **MUST** be prepared to interoperate with another implementation which does include the option.

## 1.2. Terminology

This document frequently uses the following terms:

peer           The other end of the point-to-point link.

silently discard  
This means the implementation discards the packet without further processing. The implementation **SHOULD** provide the capability of logging the error, including the contents of the silently discarded packet, and **SHOULD** record the event in a statistics counter.

end-system  
A user's machine. It only sends packets to servers and other end-systems. It doesn't pass any packets through itself.

router         Allows packets to pass through, usually from one ethernet segment to another. Sometimes these are called "intermediate-systems".

bridge         Allows packets to pass through with the data field unmodified. Usually from one ethernet segment to another or from one ethernet segment to a token-ring segment.

gateway        Allows packets to be sent from one network protocol to the same or different network protocol. For example, NetBIOS packets from an NBF network to a TCP/IP network which has implemented RFC 1001 and RFC 1002.

local access only server A server which does not pass any packets through itself to other servers.

## 2. A PPP Network Control Protocol for NBF

The NBF Control Protocol (NBFCP) is responsible for configuring, enabling, and disabling the NBF protocol modules on both ends of the point-to-point link. NBFCP uses the same packet exchange mechanism as the Link Control Protocol. NBFCP packets **MUST NOT** be exchanged until PPP has reached the Network-Layer Protocol phase. NBFCP packets received before this phase is reached should be silently

discarded.

The NBF Control Protocol is exactly the same as the Link Control Protocol [1] with the following exceptions:

#### Frame Modifications

The packet may utilize any modifications to the basic frame format which have been negotiated during the Link Establishment phase.

#### Data Link Layer Protocol Field

Exactly one NBFCP packet is encapsulated in the Information field of a PPP Data Link Layer frame where the Protocol field indicates type hex 803f (NBF Control Protocol).

#### Code field

Only Codes 1 through 7 (Configure-Request, Configure-Ack, Configure-Nak, Configure-Reject, Terminate-Request, Terminate-Ack and Code-Reject) are used. Other Codes should be treated as unrecognized and should result in Code-Rejects.

#### Timeouts

NBFCP packets MUST NOT be exchanged until PPP has reached the Network-Layer Protocol phase. An implementation should be prepared to wait for Authentication and Link Quality Determination to finish before timing out waiting for a Configure-Ack or other response. It is suggested that an implementation give up only after user intervention or a configurable amount of time. Also, because NetBIOS name defense takes time (typically a minimum of 3 seconds if names are added in parallel), it is suggested that if Name-Projection is negotiated, the timeouts are increased to 10 seconds.

#### Configuration Option Types

NBFCP has a distinct set of Configuration Options.

### 2.1. Sending NBF Datagrams

Before any NBF packets may be communicated, PPP must reach the Network-Layer Protocol phase, and the NBF Control Protocol must reach the Opened state.

Unless otherwise negotiated, exactly one NBF packet is encapsulated in the Information field of a PPP Data Link Layer frame where the

Protocol field indicates type hex 003f (NBF datagram).

Since NBF datagrams for PPP do not contain a datagram length field, the encapsulated NBF packet MUST NOT contain any extra octet padding except when Self-Defining-Padding is negotiated.

The maximum length of an NBF datagram transmitted over a PPP link is the same as the maximum length of the Information field of a PPP data link layer frame. Since there is no standard method for fragmenting and reassembling NBF datagrams, PPP links supporting NBF MUST allow at least 576 octets in the information field of a data link layer frame. It is recommended that an implementation allow 1500 octets in the information field unless the IEEE-MAC-Address-Required boolean option is negotiated (see below).

## 2.2 Bridging NBF Datagrams

There exist at least four different MAC header implementations for NBF packets: 802.3 Ethernet, 802.5 Token-Ring, DIX Ethernet, and FDDI. Because NBF is not a routable protocol, some PPP implementations may require IEEE MAC addresses to properly route or bridge NBF packets. Some PPP implementations may require the entire MAC media header in order to properly route or bridge NBF packets. Other smarter implementations may only require the IEEE MAC addresses, and still other implementations (such as NetBIOS gateways) may not require any MAC address fields. NBFCP implementations which require IEEE Addresses should negotiate the NBFCP IEEE-MAC-Address-Required boolean configuration option so that the MAC header can be provided in the NBF packet.

If IEEE-MAC-Address-Required boolean configuration option is negotiated, all NBF datagrams MUST be sent with the specified 12 octet IEEE MAC address header. Since negotiation of this option occurs after the LCP phase, NBF packets MAY exceed the negotiated PPP MRU size. A PPP implementation which negotiates this option MUST allow reception of PPP NBF packets 12 octets larger than the negotiated MRU size.

## 2.3 NetBIOS Name Defense

In order to guarantee uniqueness of NetBIOS Names on the network, NBFCP requires that end-system implementations MUST negotiate the Name-Projection configuration option.

### 3. NBFCP Configuration Options

NBFCP Configuration Options allow modifications to the standard characteristics of the network-layer protocol to be negotiated. If a Configuration Option is not included in a Configure-Request packet, the default value for that Configuration Option is assumed.

NBFCP uses the same Configuration Option format defined for LCP [1], with a separate set of Options.

Up-to-date values of the NBFCP Option Type field are specified in the most recent "Assigned Numbers" RFC [2]. Current values are assigned as follows:

- |   |                           |
|---|---------------------------|
| 1 | Name-Projection           |
| 2 | Peer-Information          |
| 3 | Multicast-Filtering       |
| 4 | IEEE-MAC-Address-Required |

#### 3.1. Name-Projection

##### Description

This Configuration Option provides a method for the peer to provide the NetBIOS names registered on its network. The sender of the Configure-Request states which NetBIOS names should be added by the remote peer. More than one Name-Projection option MAY appear in a single Configure-Request.

Implementations which do not attempt to add any NetBIOS names MUST Configure-Reject the Name-Projection Configuration Option.

If the Name-Projection Configuration Option is not offered by the remote peer, but is required by the local peer, the local peer should Configure-Nak the request and indicate that it wishes the remote peer to add zero NetBIOS names because it is the only known acceptable value. The remote peer may then terminate NBFCP, attempt to add zero NetBIOS names, or attempt add one or more NetBIOS names.

When the receiving peer cannot add all the requested names, it MUST Configure-Nak with the complete list of names requested. Those names which could be added should have the Added field set to zero. Those names which could not be added should have the Added field set to an appropriate non-zero return code. The sender of this Configuration Option SHOULD then resend the Configure-Request with the successfully added names.

The implementation may choose to fail configuration if the complete list of NetBIOS names is not accepted. By failing, the implementation should terminate NBFCP by sending a Terminate-Request packet.

Because adding NetBIOS names can take time (usually 3 seconds) and because PPP may default the restart timer to 3 seconds, the restart timer SHOULD default to 10 seconds when configuring NetBIOS names.

A summary of the Name-Projection Configuration Option format is shown below. The fields are transmitted from left to right.

```

      0                               1                               2                               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      Type      |      Length      |      1st NetBIOS-Name
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1st NetBIOS-Name (cont.)
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1st NetBIOS-Name (cont.)
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1st NetBIOS-Name (cont.)
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1st NetBIOS-Name (cont.)      |      Added      | 2nd NetBIOS Name...
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Type

1

Length

2 + (Number of NetBIOS names \* 17)

NetBIOS-Names

This group of zero or more sixteen octet NetBIOS-Name fields contains a list of all the NetBIOS names the peer wishes to add to the remote network if the packet is Configure-Request. If the packet is Configure-Reject, the peer does not support this configuration option and it can be assumed that no NetBIOS names were added.

Because the length field is only one octet, only 14 NetBIOS names can be added per Name-Projection option. If more than 14 NetBIOS names should be added, then more than one Name-Projection option packet will have to be sent in the Configure-Request packet.

## Added

This is a one octet field which plays a dual role. The Added field in the Name-Projection Request packet contains the type of NetBIOS name added. A summary of name types is listed below.

- 01 Unique Name.
- 02 Group Name.

If the packet is a Configure-Reject the Added field should contain the NetBIOS return code for the NetBIOS Add Name or NetBIOS Add Group Name command as defined in the NetBIOS 3.0 specification = [3].

A summary of common result codes is listed below in type hex.

- 00 Name successfully added.
- 0D Duplicate name in local name table.
- 0E Name table full.
- 15 Name not found or cannot specify "\*" or null.
- 16 Name in use on remote NetBIOS.
- 19 Name conflict detected.
- 30 Name defined by another environment.
- 35 Required system resources exhausted.

## 3.2. Peer-Information

### Description

This Configuration Option provides a way for the peer to communicate NetBIOS pertinent configuration information. Although negotiation of this option is not mandatory, it is suggested.

A summary of the Peer-Information Option format is shown below. The fields are transmitted from left to right.

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
Type										Length										Peer-class																			
Peer-version (major)										Peer-version(minor)																													
Peer-name ....																																							

### Type

2

### Length

$\geq 3D8$

If the length is 8, there is no Peer-name. If the length is greater than 8, the Peer-name's length is Length - 8.

### Peer-class

The Peer-class field is one octet. It identifies the sender's implementation type.

Initial values are assigned as follows:

Value	Class
1	Reserved for legacy implementations.
2	PPP NetBIOS Gateway Server.
3	Reserved for legacy implementations.
4	PPP Local Access Only Server.
5	Reserved for legacy implementations.
6	PPP NBF Bridge.
7	Reserved for legacy implementations.
8	PPP End-System.

### Peer-version

The Peer-version field is four octets and indicates the version of the communication peer providing one side of the PPP connection. The first two octets are the major version number and the last two octets are the minor version number. The major and minor version represent a 16 bit unsigned number sent with the most significant octet first.

### Peer-name

The name of the peer. A suggested name is the NetBIOS workstation name of the peer. If the length field is 8, no peer name is provided. The peer-name may not be greater than 32 octets in length.

### 3.3. Multicast-Filtering

#### Description

This Configuration Option provides a way to negotiate the use of the Multicast-Forward-Period and the Multicast-Priority. This Configuration Option provides a way to negotiate how to handle multicast packets. It allows the sender of the Configure-Request to state the current handling of multicast packets. The peer can request parameters by NAKing the option, and returning valid Multicast-Filtering parameters.

If negotiation about the remote Multicast-Filtering is required, and the peer did not provide the option in its Configure-Request, the option SHOULD be appended to a Configure-Nak.

Controlling the multicast rate is important because some NetBIOS applications use multicasts to communicate and withholding multicasts may prevent these applications from working. It is also true that other NetBIOS applications do not need to receive any multicast packets and therefore it is best to quench the rate at which the peer will send multicast packets.

By default, the peer is pre-configured to an administrator assigned Multicast-Forward-Period and Priority. A Multicast-Forward-Period specified as hex type FFFF in a Configure-Request is interpreted as requesting the receiving peer to specify a value in its Configure-Nak. A Multicast-Forward-Period value specified as hex type FFFF in a Configure-Nak is interpreted as agreement that no value exists. A Multicast-Forward-Period of zero indicates that all multicast packets SHOULD be forwarded.

Peers that rely on all multicast packets being forwarded SHOULD request a Multicast-Forward-Period of zero and a Multicast-Priority of one by NAKing the Configure-Request option and appending the proper parameters to a Configure-Nak.

A summary of the Multicast-Filtering Configuration Option format is shown below. The fields are transmitted from left to right.

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
Type										Length										Multicast-Forward-Period																			
Priority																																							

Type

3

Length

5

Multicast-Forward-Period

The Multicast-Forward-Period field is two octets and indicates the maximum period in seconds at which multicast packets can be sent. The maximum value for this field is 60 (one minute). A value of zero indicates that there is no maximum period at which multicast packets can be sent. A value of hex type FFFF indicates that the Multicast-Forward-Period is unknown. A value of five indicates that multicast packets will not be sent at a rate more frequent than once every five seconds. This two octet value represents a 16 bit unsigned number sent with the most significant octet first.

Priority

The Priority field is one octet long and indicates if multicast packets have priority over other packets when being sent. A value of 0 indicates that directed packets have priority. A value of 1 indicates that multicast packets have priority.

### 3.4. IEEE-MAC-Address-Required

Description

This boolean Configuration Option provides a method for the peer to require that all NBF datagrams be sent with a 12 octet IEEE MAC Address header. By default, it is assumed that no MAC header is required.

A summary of the IEEE-MAC-Address-Required Boolean Configuration Option format is shown below. The fields are transmitted from left to right.

0										1					
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---															
Type										Length					
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---															

Type

4

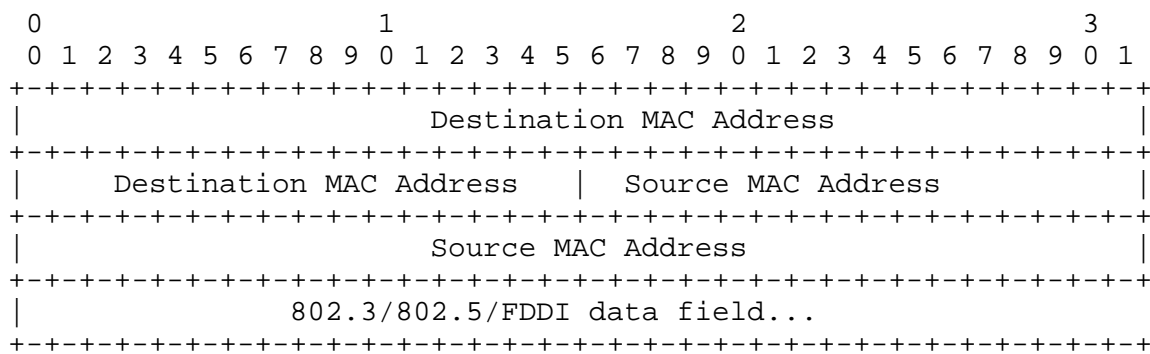
Length

2

## Requirements

By default the NBF datagram is sent without any MAC header information. The NBF datagram information field is equivalent to the data field in 802.3, 802.5, and FDDI frames.

If this option is negotiated successfully, each NBF datagram is sent with a 12 octet IEEE MAC Address header prepended to the information field. A summary of the information field when using 12 octet IEEE MAC Headers is shown below. The fields are transmitted from left to right. The MAC Address is in non-canonical form. This means that the first bit to be transmitted in every byte is the most significant bit.



## Security Considerations

Security issues are not discussed in this memo.

## References

- [1] Simpson, W., Editor, "The Point-to-Point Protocol (PPP)", STD 51, RFC 1661, July 1994.
- [2] Reynolds, J., and J. Postel, "Assigned Numbers", STD 2, RFC 1700, October 1994.
- [3] IBM Corp., "IBM Local Area Network Technical Reference", Third Edition, Document Number SC30-3383-2, November 4, 1988.

- [4] Baker, F., and R. Bowen "PPP Bridging Control Protocol (BCP)",  
Work in Progress.

#### Acknowledgments

Some of the text in this document is taken from previous documents produced by the Point-to-Point Protocol Working Group of the Internet Engineering Task Force (IETF).

Thomas J. Dimitri (previously at Microsoft Corporation) authored the original draft.

Special thanks go to coworkers at Microsoft, Bill Simpson (Daydreamer), Tom Coradetti (DigiBoard), Marty Del Vecchio (Shiva), Russ Gocht (Shiva) and several members of the IETF PPP Working Group.

#### Chair's Address

The working group can be contacted via the current chair:

Karl Fox  
Ascend Communications  
3518 Riverside Drive, Suite 101  
Columbus, Ohio 43221

karl@MorningStar.com  
karl@Ascend.com

#### Author's Address

Questions about this memo can also be directed to:

Gurdeep Singh Pall  
Microsoft Corporation  
1 Microsoft Way  
Redmond, WA 98052-6399

EMail: gurdeep@microsoft.com

