Bluetooth Network Encapsulation Protocol (BNEP) Specification

Abstract:

The Personal Area Networking (PAN) Bluetooth Network Encapsulation Specification describes the protocol to be used by the Bluetooth PAN profiles. This document defines a packet format for Bluetooth network encapsulation used to transport common networking protocols over the Bluetooth media. Bluetooth network encapsulation supports the same networking protocols that are supported by IEEE 802.3/Ethernet encapsulation. Packets from the supported networking protocols are contained in Bluetooth network encapsulation packets, which are transported directly over the Bluetooth L2CAP protocol.

Disclaimer and copyright notice

such compliance, authorizations or licenses**. NOTHING IN THE SPECIFICATION CREATES ANY WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING SUCH LAWS OR REGULATIONS.**

ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHTS OR FOR NONCOMPLIANCE WITH LAWS, RELATING TO USE OF THE SPECIFICATION IS EXPRESSLY DISCLAIMED. BY USE OF THE SPECIFICATION, EACH MEMBER EXPRESSLY WAIVES ANY CLAIM AGAINST BLUETOOTH SIG AND ITS PROMOTER MEMBERS RELATED TO USE OF THE SPECIFICATION.

Bluetooth SIG reserve the right to adopt any changes or alterations to the Specification as it deems necessary or appropriate and to adopt a process for adding new Bluetooth™ profiles after the release of the Specification.

Contents

Revision History

Revision	Date	Comments
0.0	April 4, 2000	Original Document Started
0.1	May 11, 2000	Added Ethernet encapsulation protocol proposal
0.45	July 24, 2000	Added Monte Carlo F2F feedback
0.46	Aug 16, 2000	Added Pittsburgh F2F feedback
0.5	September 26, 2000	Editorial changes Released to Adopters
0.6	October 12, 2000	Editorial changes Change to the protocol on the filters options
0.7	November 27, 2000	Editorial changes Add extension bit and extension header definitions
0.9	February 6, 2001	Minor Editorial changes
0.95	June 12, 2001	Editorial changes and changes based on BARB, BTI, BQRB, BTAB feedback. Support for 802.1p Filter Message clarification Filter Message for Multicast Connection Setup Network control message added
0.95a	June 26, 2001	Editorial changes and changes based on BARB & BTI adoption review.

Contributors

Name	Company
David Moore	3COM Corporation
Tom Scribner	3COM Corporation
Barry Corlett	Agere Systems
Willy Sagefalk	Axis Communications
Dan Willey	Certicom Corporation
Horia Balog	Classwave Wireless Inc.
Conrad Maxwell	Conexant Systems
Allan Bogeskov	Telefonaktiebolaget LM Ericsson
Theo Borst	Telefonaktiebolaget LM Ericsson
Per Johansson	Telefonaktiebolaget LM Ericsson
Tero Kauppinen	Telefonaktiebolaget LM Ericsson
Martin Kitchen	Telefonaktiebolaget LM Ericsson
Jesper Krogh	Telefonaktiebolaget LM Ericsson
Tony Larsson	Telefonaktiebolaget LM Ericsson
Johan Sorensen	Telefonaktiebolaget LM Ericsson
Dave Suvak	Extended Systems Inc.
Jean Tourrilhes	Hewlett Packard Corporation
Toru Aihara	International Business Machines Corporation
Chatschik Bisdikian	International Business Machines Corporation
Kris Fleming (Editor)	Intel Corporation
Robert Hunter	Intel Corporation
Jon Inouye	Intel Corporation
Diego Melpignano	Philips Inc.
Eiji Kato	Matsushita Electric Industrial
Mike Foley	Microsoft Corporation

Billy Brackenridge	Microsoft
Dale Farnsworth	Motorola Inc.
Carmen Kuhl	Nokia Corporation
Jaakko Lipasti	Nokia Corporation
James Scales	Nokia Corporation
Markus Schetelig	Nokia Corporation
Sander van Valkenburg	Nokia Corporation
Steven Kenny	Norwood Systems
Rebecca Ostergaard	Norwood Systems
Graeme Reid	Norwood Systems
Darrell Goff	Rappore
Daniel Shaw	Red-M Inc.
Pravin Bhagwat	ReefEdge, Inc.
Daryl Hlasny	Sharp Laboratories of America Inc.
Leonard Ott	Socket Communications Inc.
Johannes Lobbert	Sony Corporation
Wilhelm Hagg	Sony Corporation
Mike Blackstock	Synchropoint Wireless, Inc.
Yosuke Tajika	Toshiba Corporation
Tatuya Junmei	Toshiba Corporation
Kazuo Nogami	Toshiba Corporation
Jim Hobza	Widcomm Inc.

1 Introduction

Bluetooth is a short-range wireless technology operating in the 2.4 GHz ISM band. Many devices such as notebook computers, phones, PDAs, Home Electric Appliances, and other computing devices will incorporate Bluetooth as a part of the device. Bluetooth enabled devices will have the ability to form networks and exchange information. For these devices to interoperate and exchange information, a common packet format must be defined to encapsulate layer 3 network protocols. This document defines the packet format used to transport common networking protocols over the Bluetooth media[5][6][7]. The packet format is based on EthernetII/DIX Framing as defined by IEEE 802.3[3][4].

Bluetooth Network Encapsulation Protocol (BNEP) encapsulates packets from various networking protocols, which are transported directly over the Bluetooth Logical Link Control and Adaptation Layer Protocol (L2CAP) protocol [2]. L2CAP provides a data link layer for Bluetooth.

The Bluetooth Personal Area networking profile [1] describes how BNEP shall be used to provide networking capabilities for Bluetooth devices.

1.1 Bluetooth Networking Encapsulation Protocol (BNEP) Functional Requirements

The functional requirements for Bluetooth networking encapsulation protocol includes the following:

- Support for common networking protocols such as IPv4, IPv6, IPX, and other existing or emerging networking protocols as defined by the Network protocol types [3]. Many protocols are used for networking various computing devices together. Although IPv4 and IPv6 are perceived as the most important networking protocols, it is a requirement that Bluetooth Networking is able to supports other popular protocols
- Low Overhead -- The encapsulation format should be bandwidth efficient.

1.2 Assumptions

1. This protocol is implemented on connection oriented L2CAP.

2. Bluetooth is considered to be a transmission media in the same OSI layer as Ethernet, Token Ring, ATM, etc.

3. L2CAP is considered to be the Bluetooth Data MAC (Media Access Control) Layer.

4. BNEP specifies a minimum L2CAP MTU of 1691 bytes[1]. This is also the default L2CAP MTU for BNEP, such as IEEE 802.3.

5. The accepted rules of network connectivity and topology as defined for IEEE 802.3 (e.g. switching and routing) must be applied to Bluetooth in a manner consistent with IEEE 802.3 media.

6. The Bluetooth BD_ADDR address space administered by the IEEE, which means that it is possible to build a Bluetooth network access point as a bridge between Bluetooth devices and an Ethernet network.

1.3 Scope

This document covers only the Bluetooth networking encapsulation packet format. The follow items are beyond the scope of the Bluetooth Networking encapsulation document:

- Address Allocation
- Address Resolution
- Name Resolution
- Networking Security
- Routing
- Network Discovery
- Network Formation

The above issues are addressed in the Bluetooth Personal Area Networking profile document [1]. The Bluetooth Personal Area Networking profile [1] describes how the Bluetooth networking encapsulation is used to provide networking support.

1.4 Byte Order and Numeric Values

All values contained in the document are represented in hexadecimal notation. Multiple-byte fields are drawn with the more significant bytes toward the left and

[1] The minimum MTU of 1691 was selected based on the payload of a maximum Ethernet packet payload (1500 bytes) + BNEP header (15 bytes) + L2CAP header (4 bytes) + possible extension header. This minimum MTU is required to prevent violating any higher layer protocol assumptions about an "EthernetII/DIX Framing like" layer provided by BNEP. 1691 = 5*339(size of DH5) – 4 (L2CAP header).

the less significant bytes toward the right. The multiple-byte fields in the Bluetooth Networking encapsulation header are in standard network byte order (big endian), with more significant (byte 0 is the most significant byte) bytes being transferred before less-significant (low-order) bytes.

2 BNEP over L2CAP

2.1 Stack Overview

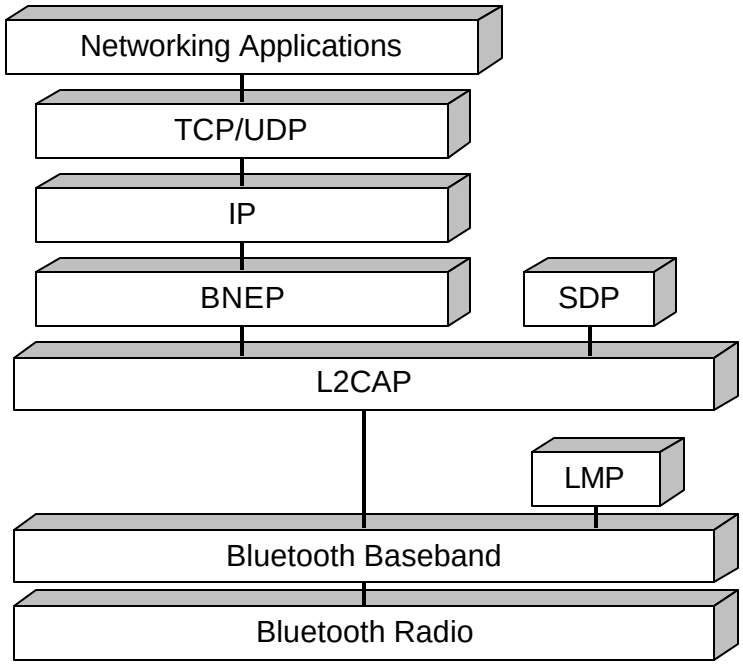

Figure 1: Stack Overview

2.2 Packet Encapsulation

The use of the BNEP for transporting an Ethernet packet is shown in Figure 2 on page 13. BNEP removes and replaces the Ethernet Header with the BNEP Header. The Ethernet Payload remains unchanged. Finally, both the BNEP Header and the Ethernet Payload is encapsulated by L2CAP and is sent over the Bluetooth media. The maximum payload that BNEP will accept from the higher layer is equal to the negotiated L2CAP MTU (minimum value: 1691), minus 191 (reserved for BNEP headers). This way it can be assured that enough frame buffer space is reserved to transmit all BNEP.

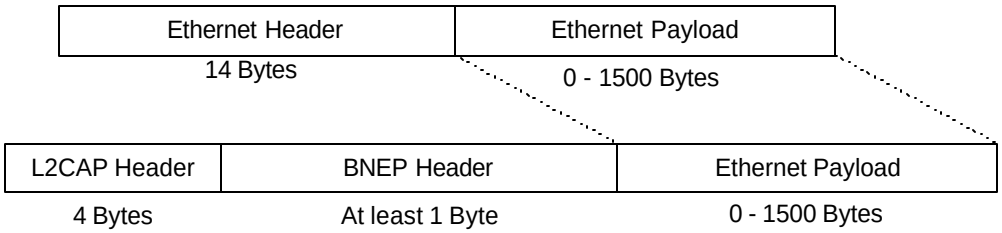

Figure 2: BNEP with an Ethernet Packet payload sent using L2CAP

2.3 BNEP Overview

BNEP is used for transporting both control and data packet over Bluetooth to provide networking capabilities for Bluetooth devices. BNEP provides capabilities that are similar to capabilities provided by Ethernet (EthernetII/DIX Framing /IEEE 802.3).

2.4 BNEP Header Formats

All BNEP Headers are in the following format as shown in Figure 4 on page 13. All devices supporting BNEP must be able to interpret all defined BNEP packet types. BNEP capable devices may or may not transmit the BNEP compressed headers. Any packet containing a reserved BNEP header packet type must be dropped. Processing of extension headers are defined in section 3 on page 36.

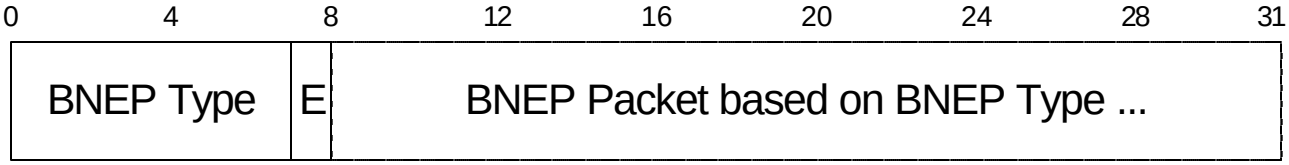

Figure 4 BNEP Header Format

BNEP Type: *Size: 7 Bits*

Value	Parameter Description
0x00 – 0x7F	Seven bit Bluetooth Network Encapsulation Protocol Type value identifies the type of BNEP header contained in this packet. Values are defined in Table 1 on page 14

Extension Flag (E): *Size: 1 Bit*

Value	Parameter Description
0x0 – 0x1	One bit extension flag that indicates if one or more extension headers follow the BNEP Header before the data payload. Extension headers are defined in section 3 on page 36. If the extension flag is equal to 0x1 then one or more extension headers follows the BNEP header. If the extension flag is equal to 0x0 then the BNEP payload follows the BNEP header.

BNEP Packet: *Size: Based on BNEP Type*

Value	Parameter Description
0xXX	Based on the BNEP Type

2.4.1 BNEP Type Values

The Table 1 on page 14 defines the various BNEP packet formats

Value	BNEP Packet Type
0x00	BNEP_GENERAL_ETHERNET
0x01	BNEP_CONTROL
0x02	BNEP_COMPRESSED_ETHERNET
0x03	BNEP_COMPRESSED_ETHERNET_SOURCE_ONLY
0x04	BNEP_COMPRESSED_ETHERNET_DEST_ONLY
0x05 – 0x7E	Reserved for future use
0x7F	Reserved for 802.2 LLC Packets for IEEE 802.15.1 WG

Table 1: BNEP Types

2.5 BNEP_GENERAL_ETHERNET Packet Type Header Format

The BNEP_GENERAL_ETHERNET packet type header format is shown in Figure 6 on page 15. This packet type shall be used to carry Ethernet packets to and from Bluetooth networks.

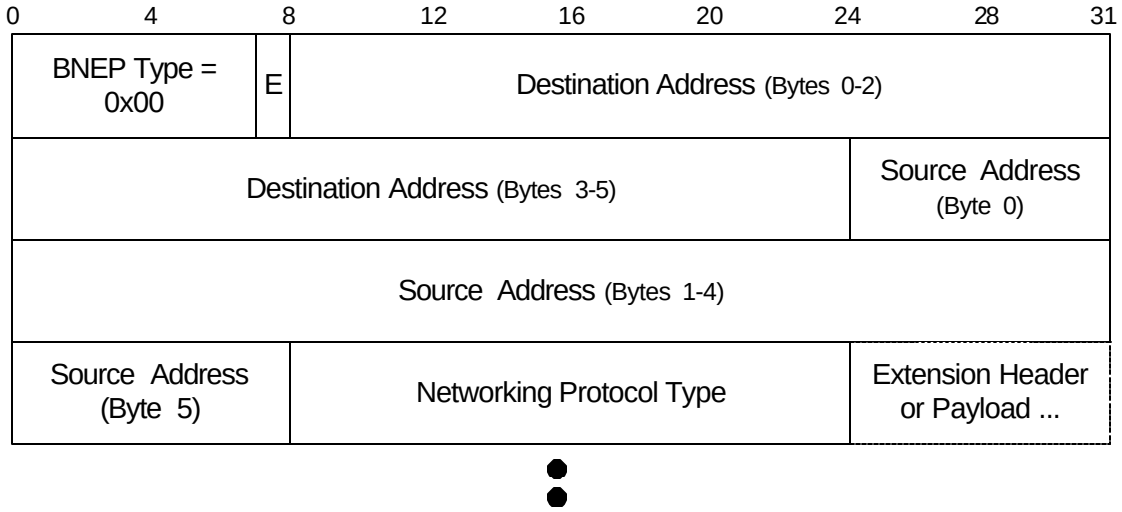

Figure 6: BNEP_GENERAL_ETHERNET Packet Type Header

BNEP Type: *Size: 7 Bits*

Value	Parameter Description
0x00	Seven bit Bluetooth Network Encapsulation Protocol Type value identifies the type of BNEP header contained in this packet. MUST be set to BNEP_GENERAL_ETHERNET

Extension Flag (E): *Size: 1 Bit*

Value	Parameter Description
0x0 – 0x1	One bit extension flag that indicates if one or more extension headers follow the BNEP Header before the data payload. Extension headers are defined in section 3 on page 36. If the extension flag is equal to 0x1 then one or more extension headers follows the BNEP header. If the extension flag is equal to 0x0 then the BNEP payload follows the BNEP header.

Destination Address: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	48 bit Bluetooth device address/IEEE address of the destination of the BNEP packet/Ethernet frame contained in the payload.

Source Address: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	48 bit Bluetooth device address/IEEE address of the source of the BNEP packet/Ethernet frame contained in the payload.

Networking Protocol Type: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	16 bit type field identifies the type of networking protocol contained in the payload. The values for this field are the same as defined for Ethernet types in [3]

Either the destination or the source address may be an IEEE Ethernet address, if the actual destination/source is an IEEE device and not a Bluetooth device. BNEP shall use IEEE Ethernet broadcast and multicast addresses for the source and/or destination addresses for broadcast and multicast packets.

2.6 BNEP_ CONTROL Packet Type Header Format

The BNEP_ CONTROL packet type header format is shown in Figure 8 on page 16. This packet type is mandatory to recognize and respond to accordingly. The BNEP_CONTROL packet type is used to exchange control information.

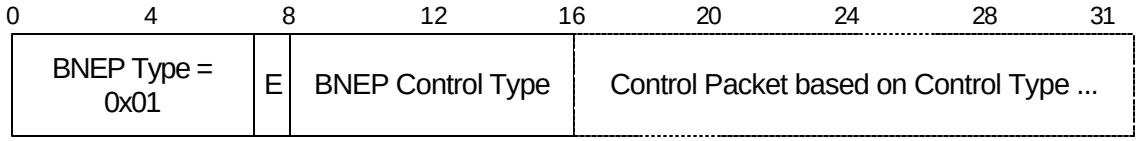

Figure 8: BNEP_ CONTROL Packet Type Header

BNEP Type: *Size: 7 Bits*

Value	Parameter Description
0x01	Seven bit Bluetooth Network Encapsulation Protocol Type value identifies the type of BNEP header contained in this packet. MUST be set to BNEP _CONTROL

Extension Flag (E): *Size: 1 Bit*

Value	Parameter Description
0x0 – 0x1	One bit extension flag that indicates if one or more extension headers follow the BNEP Header before the data payload. Extension headers are defined in section 3 on page 36. If the extension flag is equal to 0x1 then one or more extension headers follows the BNEP header. If the extension flag is equal to 0x0 then the BNEP payload follows the BNEP header.

BNEP Control Type: *Size: 1 Byte*

Value	Parameter Description
0x00 – 0xFF	Type of BNEP control message contained in the packet

2.6.1 BNEP Control Type Values

Table 3 on page 17 defines the various BNEP control and response message values to for the BNEP Control type field.

Value	BNEP Control Type
0x00	BNEP_CONTROL_COMMAND_NOT_UNDERSTOOD
0x01	BNEP_SETUP_CONNECTION_REQUEST_MSG
0x02	BNEP_SETUP_CONNECTION_RESPONSE_MSG
0x03	BNEP_FILTER_NET_TYPE_SET_MSG
0x04	BNEP_FILTER_NET_TYPE_RESPONSE_MSG
0x05	BNEP_FILTER_MULTI_ADDR_SET_MSG
0x06	BNEP_FILTER_MULTI_ADDR_RESPONSE_MSG
0x07 – 0xFF	Reserved for future use

Table 3: BNEP Control Types

2.6.2 BNEP_CONTROL_COMMAND_NOT_UNDERSTOOD Control Command Packet

This packet shall be used to reply to any control message received, which contains an unknown BNEP control type value. This allows devices to response to control message that may be used in the future.

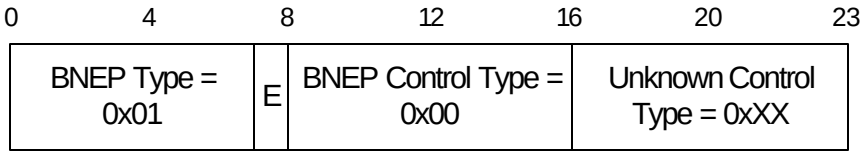

Figure 10: BNEP_CONTROL_COMMAND_NOT_UNDERSTOOD control message format

BNEP Type: *Size: 7 Bits*

Value	Parameter Description
0x01	Seven bit Bluetooth Network Encapsulation Protocol Type value identifies the type of BNEP header contained in this packet. MUST be set to BNEP_CONTROL

Extension Flag (E): *Size: 1 Bit*

Value	Parameter Description
0x0	One bit extension flag that indicates that no extension headers following the BNEP Header. For BNEP_CONTROL_COMMAND_NOT_UNDERSTOOD packets no extension headers are allowed.

BNEP Control Type: *Size: 1 Byte*

Value	Parameter Description
0x00	Type of BNEP control message contained in the packet. MUST be set to BNEP_CONTROL_COMMAND_NOT_UNDERSTOOD

Unknown Control Type: *Size: 1 Byte*

Value	Parameter Description
0xXX	Type of BNEP control message that was previous received and caused this message to be sent.

2.6.3 BNEP_SETUP_CONTROL Packets

This packet type shall contain control messages used to setup the initial connection information about the BNEP connection. All devices that support BNEP shall be able to recognize and respond to accordingly to all BNEP_SETUP_CONTROL packets.

BNEP_SETUP_CONTROL packet types must be processed in order that they are received. For each connection, only one outstanding BNEP_SETUP_CONTROL message is allowed. A response message must be used to respond to each control message received. If a response message is not received, after T_{crt} time has elapsed, then the outstanding BNEP_SETUP_CONTROL message can be assumed to be lost and the same BNEP_SETUP_CONTROL message can be retransmitted. The range for T_{crt} is from 1 second to 30 seconds, with a suggested timeout value to be 10 seconds. BNEP packets of type BNEP_SETUP_CONTROL are for the device with direct connection communication only, and must never be forwarded.

2.6.3.1 BNEP_SETUP_CONNECTION_REQUEST_MSG setup control message format

The BNEP_SETUP_CONNECTION_REQUEST_MSG setup control message format is shown in Figure 11 on page 19. The purpose of this control message is to inform the peer entity the destination and source SDP service UUIDs [8] which are being used for this BNEP connection. The device, which is establishing the L2CAP connection for BNEP, is required to send this packet and receive a successful response before sending any additional packets with other BNEP packet types. This setup packet SHALL be successfully transmitted to the other device and successfully responded to by that device before sending any other BNEP packets. In addition, the BNEP setup message can be used to switch the current roles for the BNEP connection.

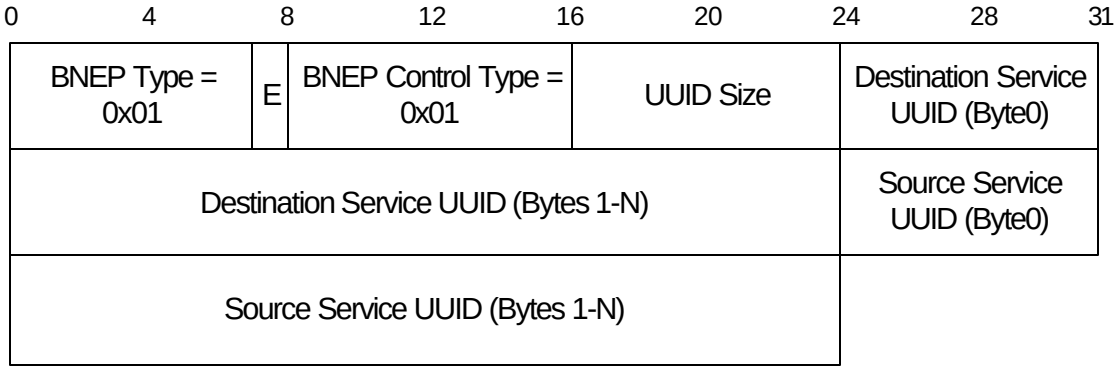

Figure 11: BNEP_SETUP_CONNECTION_REQUEST_MSG control message format

BNEP Type: *Size: 7 Bits*

Value	Parameter Description
0x01	Seven bit Bluetooth Network Encapsulation Protocol Type value identifies the type of BNEP header contained in this packet. MUST be set to BNEP_CONTROL

Extension Flag (E): *Size: 1 Bit*

Value	Parameter Description
0x0	One bit extension flag that indicates that no extension headers following the BNEP Header. For BNEP_SETUP_CONNECTION_REQUEST_MSG packets no extension headers are allowed.

BNEP Control Type: *Size: 1 Byte*

Value	Parameter Description
0x01	Type of BNEP control message contained in the packet. MUST be set to BNEP_SETUP_CONNECTION_REQUEST_MSG

UUID Size: *Size: 1 Byte*

Value	Parameter Description
0xX	1 byte field identifies the length of the SDP service UUIDs [8], measured in bytes. Note: The size of both the destination and source service UUID must be the same.

Destination Service UUID: *Size: 2-16 Bytes*

Value	Parameter Description
0xXX	Depending on the UUID Size parameter, this is a 2-16 byte field containing the destination (service which the source device is connecting to) SDP service UUIDs [8]. Note: The size of both the destination and source service UUID must be the same.

Source Service UUID: *Size: 2-16 Bytes*

Value	Parameter Description
0xXX	Depending on the UUID Size parameter, this is a 2-16 byte field containing the source (the service that the source device is using for the BNEP connection) SDP service UUIDs [8]. Note: The size of both the destination and source service UUID must be the same.

2.6.3.2 BNEP_SETUP_ CONNECTION_RESPONSE_MSG response message format

The BNEP_SETUP_ CONNECTION_RESPONSE_MSG response message format is shown in Figure 15 on page 25. The response message shall be used to respond to each BNEP_SETUP_CONNECTION_REQUEST_MSG control message. Each of the received setup control messages must be responded to by one response message.

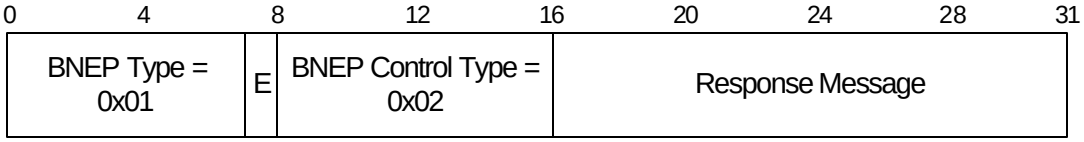

Figure 8: BNEP_SETUP_ CONNECTION_RESPONSE_MSG response message format

BNEP Type: *Size: 7 Bits*

Value	Parameter Description
0x01	Seven bit Bluetooth Network Encapsulation Protocol Type value identifies the type of BNEP header contained in this packet. MUST be set to BNEP_CONTROL

Extension Flag (E): *Size: 1 Bit*

Value	Parameter Description
0x0	One bit extension flag that indicates that no extension headers following the BNEP Header. For BNEP_SETUP_ CONNECTION_RESPONSE_MSG packets no extension headers are allowed.

BNEP Control Type: *Size: 1 Byte*

Value	Parameter Description
0x02	Type of BNEP control message contained in the packet. MUST be set to BNEP_SETUP_ CONNECTION_RESPONSE_MSG

Response Message: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	16 bit field identifies the response to the previous Setup Control Message. Valid responses are contained in Table 5 on page 22.

2.6.3.2.1 Response Messages

Table 5 on page 22 contains a list the valid response messages to be used to respond to setup control messages.

Value	Response Messages
0x0000	Operation Successful
0x0001	Operation Failed: Invalid Destination Service UUID
0x0002	Operation Failed: Invalid Source Service UUID
0x0003	Operation Failed: Invalid Service UUID Size
0x0004	Operation Failed: Connection not allowed
0x0004 – 0xFFFF	Reserved for future use

Table 5: Setup Connection Response Messages

2.6.4 BNEP_FILTER_CONTROL Packets

This packet type shall contain control messages used to control, which types of packets are to be transmitted over BNEP. Although, all devices that support BNEP shall be able to recognize and respond to accordingly to all BNEP_FILTER_CONTROL packets, the implied functionality to do filtering is optional and does not have to be supported by all devices.

BNEP_FILTER_CONTROL packet types must be processed in order that they are received. This allows Bluetooth devices to determine which types of packets are to be filtered to save networking bandwidth. For each connection, only one outstanding BNEP_FILTER_CONTROL message is allowed. A response message must be used to respond to each filter control message received. If a response message is not received, after T_{frt} time has elapsed, then the outstanding BNEP_FILTER_CONTROL message can be assumed to be lost and the same BNEP_FILTER_CONTROL message can be retransmitted. The range for T_{frt} is from 1 second to 30 seconds, with a suggested timeout value to be 10 seconds. BNEP packets of type BNEP_FILTER_CONTROL are for the device with direct connection communication only, and must never be forwarded.

Note that any new filter control message replaces the previous filter control settings.

2.6.5 BNEP Filter Network Protocol Type

This packet type shall contain control messages used to control, which Network Protocol types should be filtered and not transmitted over BNEP. By default all

Network Protocol types are not filtered, but network administrators may configure overriding default filters.

2.6.5.1 BNEP_FILTER_ NET_TYPE_SET_MSG filter control message format

The BNEP_FILTER_ NET_TYPE_SET_MSG filter control message format is shown in Figure 9 on page 24. The purpose of this control message is to inform the peer entity of the set of Networking Protocol Types that the sender whishes to receive[2]. Note that the filter control message does not change the settings, unless its response message returns Operation Successful.

The length (in octets) of this message is 2+4*N, where N is the number of disjoint ranges of Networking protocol types that form the complete set. Note that N=0 (empty set) denotes a reset to default filters (if any) supported by the remote device.

When the filtering is enabled and supported, only packets containing networking protocol types that are within the indicated Networking Protocol Type range(s) are sent using BNEP. See section 5.4 for an example of how filters may be applied. Note: Some BNEP packets will contain IEEE 802.1p header and all devices supporting BNEP shall be able to understand IEEE 802.1p, see section 4.1 on page 39. In addition, depending on the value of the network protocol type field, the actual network protocol type embedded in the payload may be contained at a known offset, which can be determined based on the network protocol type field contained in the BNEP header. Note: BNEP implementations should be aware of the fact that in case of 802.1p and 802.3, the actual 'type' field that is being used in filtering, has to be found inside the Ethernet payload, offset by 4 and 8 bytes respectively.

[2] In Network access points, the actual filters being applied may be further affected by policies set by a network administrator.

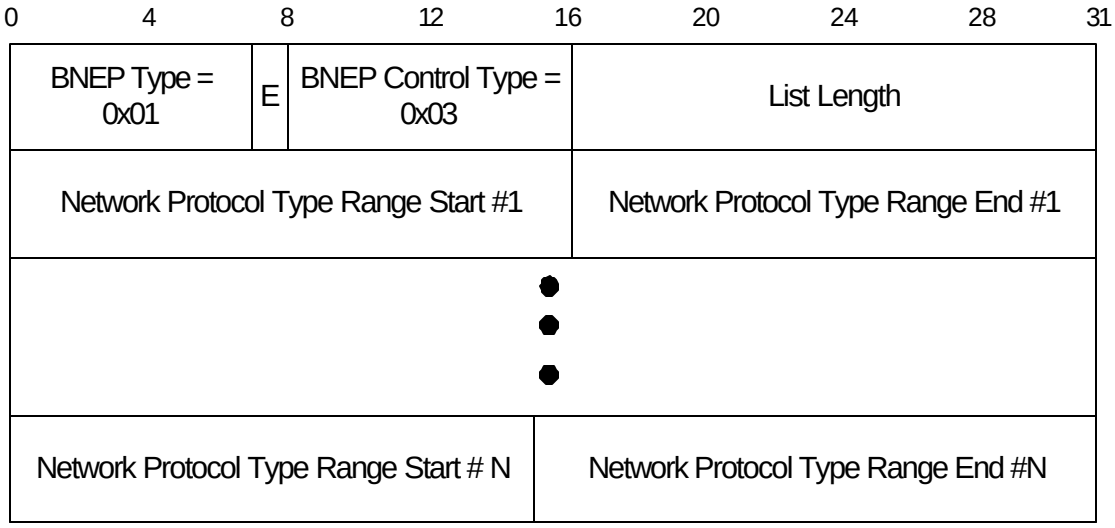

Figure 9: BNEP_FILTER_ NET_TYPE_SET_MSG control message format

BNEP Type: *Size: 7 Bits*

Value	Parameter Description
0x01	Seven bit Bluetooth Network Encapsulation Protocol Type value identifies the type of BNEP header contained in this packet. MUST be set to BNEP_ CONTROL

Extension Flag (E): *Size: 1 Bit*

Value	Parameter Description
0x0 – 0x1	One bit extension flag that indicates if one or more extension headers follow the BNEP Header before the data payload. Extension headers are defined in section 3 on page 36. If the extension flag is equal to 0x1 then one or more extension headers follows the BNEP header. If the extension flag is equal to 0x0 then the BNEP payload follows the BNEP header.

BNEP Control Type: *Size: 1 Byte*

Value	Parameter Description
0x03	Type of BNEP control message contained in the packet. MUST be set to BNEP_FILTER_ NET_TYPE_SET_MSG

List Length: *Size: 2 Byte*

Value	Parameter Description
0xXXXX	Length of the start and end range of network protocol types list. This field defines the number of bytes contain in the entire list.

The following two fields are repeated together N times, N=0…421.

Networking Protocol Type Start #N: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	16 bit type field identifies the start of a range of type of networking protocols to be enabled. The values for this field are the same as defined for Ethernet types in [3]

Networking Protocol Type End #N: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	16 bit type field identifies the end of the range of type of networking protocols to be enabled. The values for this field are the same as defined for Ethernet types in [3]

2.6.5.2 BNEP_FILTER_ NET_TYPE_RESPONSE_MSG response message format

The BNEP_FILTER_ NET_TYPE_RESPONSE_MSG response message format is shown in Figure 15 on page 25. The response message shall be used to respond to each BNEP_FILTER_NET_TYPE_SET_MSG filter control message. Each of the received filter control messages must be responded to by one response message. Note that the filter control message does not change the settings, unless its response message returns Operation Successful.

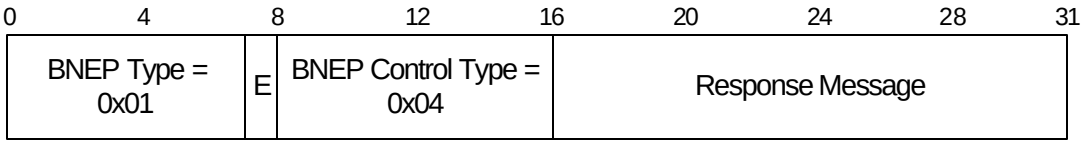

Figure 15: BNEP_FILTER_ NET_TYPE_RESPONSE_MSG response message format

BNEP Type: *Size: 7 Bits*

Value	Parameter Description
0x01	Seven bit Bluetooth Network Encapsulation Protocol Type value identifies the type of BNEP header

	contained in this packet. MUST be set to BNEP_ CONTROL

Extension Flag (E): *Size: 1 Bit*

Value	Parameter Description
0x0 – 0x1	One bit extension flag that indicates if one or more extension headers follow the BNEP Header before the data payload. Extension headers are defined in section 3 on page 36. If the extension flag is equal to 0x1 then one or more extension headers follows the BNEP header. If the extension flag is equal to 0x0 then the BNEP payload follows the BNEP header.

BNEP Control Type: *Size: 1 Byte*

Value	Parameter Description
0x04	Type of BNEP control message contained in the packet. MUST be set to BNEP_FILTER_ NET_TYPE_RESPONSE_MSG

Response Message: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	16 bit type field identifies the response to the previous Filter Control Message. Valid responses are contained in Table 7 on page 26.

2.6.5.2.1 Response Messages

Table 7 on page 26 contains a list the valid response messages to be used to respond to filter control messages.

Value	Response Messages
0x0000	Operation Successful
0x0001	Unsupported Request
0x0002	Operation Failed: Invalid Networking Protocol Type Range
0x0003	Operation Failed: Maximum Filter Limited Reached
0x0004	Operation Failed: Unable to fulfill request due to security reasons.
0x0005 – 0xFFFF	Reserved for future use

Table 7: Network Protocol Type Filter Response Messages

2.6.6 BNEP Filter Multicast Address Type

This packet type shall contain control messages used to control, which multicast destination addresses should not be filtered and transmitted over BNEP. By default all Multicast Addresses are not filtered.

2.6.6.1 BNEP_FILTER_ MULTI_ADDR _SET_MSG filter control message format

The BNEP_FILTER_ MULTI_ADDR _SET_MSG filter control message format is shown in Figure 12 on page 28. The purpose of this control message is to inform the peer entity of the set of Multicast Addresses that the sender whishes to receive and all other Multicast Addresses not contained in the set shall be filtered[3]. Note that the filter control message does not change the settings, unless its response message returns Operation Successful.

The length (in octets) of this message is 2+(2*6)*N, where N is the number Multicast Addresses that form the complete set. Note that N=0 (empty set) denotes a reset to default filters (if any) supported by the remote device.

When the filtering is enabled and supported, only packets with the Destination Address field contained in one of the Multicast Addresses range contained in BNEP_FILTER_MULTI_ADDR_SET_MSG are sent using BNEP. See section 5.5 for an example of how Multicast Address filters may be applied.

[3] In Network access points, the actual filters being applied may be further affected by policies set by a network administrator.

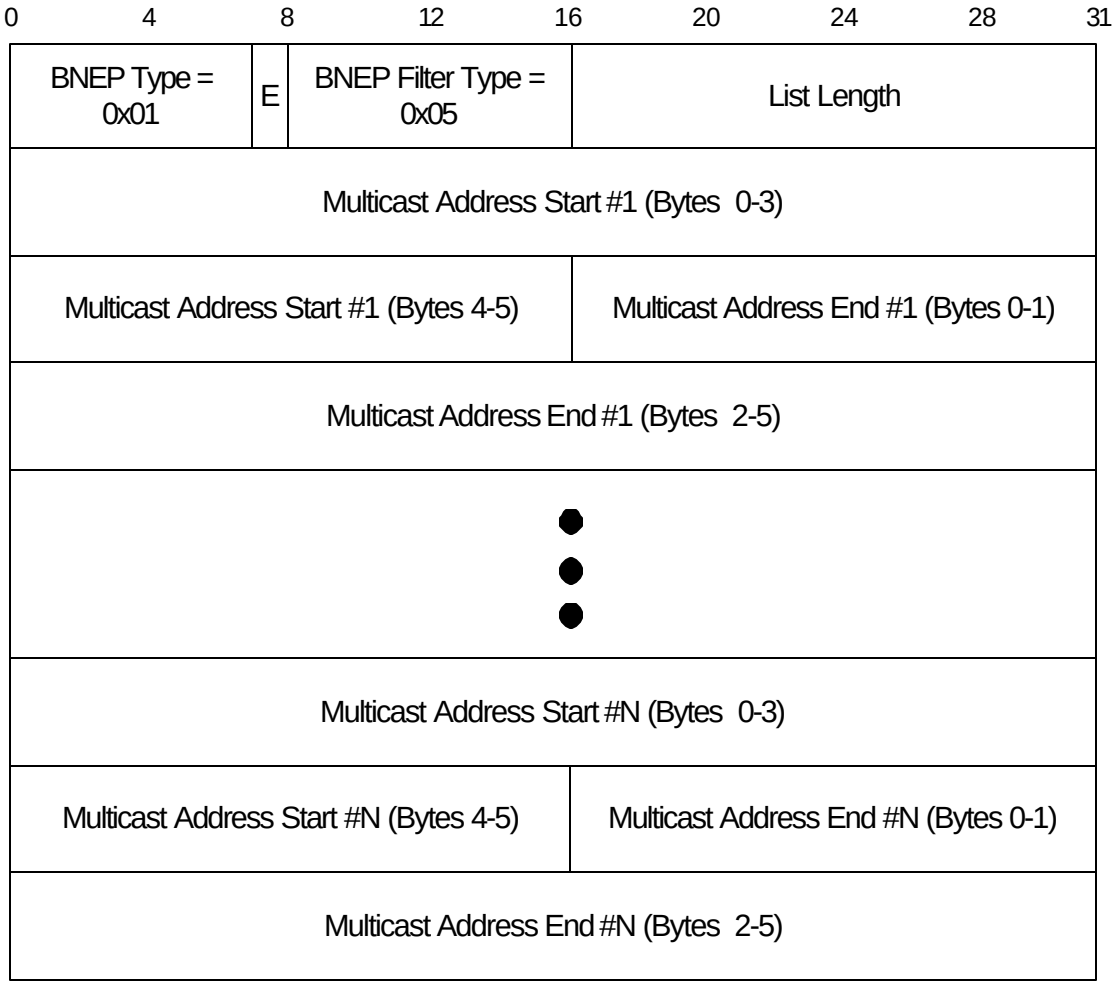

Figure 12: BNEP_FILTER_MULTI_ADDR_SET_MSG control message format

BNEP Type: *Size: 7 Bits*

Value	Parameter Description
0x01	Seven bit Bluetooth Network Encapsulation Protocol Type value identifies the type of BNEP header contained in this packet. MUST be set to BNEP_ CONTROL

Extension Flag (E): *Size: 1 Bit*

Value	Parameter Description
0x0 – 0x1	One bit extension flag that indicates if one or more extension headers follow the BNEP Header before the data payload. Extension headers are defined in section 3 on page 36. If the extension flag is equal to 0x1 then

	one or more extension headers follows the BNEP header. If the extension flag is equal to 0x0 then the BNEP payload follows the BNEP header.

BNEP Control Type: *Size: 1 Byte*

Value	Parameter Description
0x05	Type of BNEP control message contained in the packet. MUST be set to BNEP_FILTER_ MULTI_ADDR_SET_MSG

List Length: *Size: 2 Byte*

Value	Parameter Description
0xXXXX	Length of the start and end range of multicast IEEE address range list. This field defines the number of bytes contain in the entire list.

The following two fields can be repeated together N times, N=0…140.

Multicast Address Start #N: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	Start of the range of 48 bit Multicast IEEE address not to be filtered.

Multicast Address End #N: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	End of the range of 48 bit Multicast IEEE address not to be filtered.

2.6.6.2 BNEP_FILTER_MULTI_ADDR_RESPONSE_MSG response message format

The BNEP_FILTER_MULTI_ADDR_RESPONSE_MSG response message format is shown in Figure 15 on page 31. The response message shall be used to respond to each BNEP_FILTER_MULTI_ADDR_SET_MSG filter control message. Each of the received filter control messages must be responded to by one response message. Note that the filter control message does not change the settings, unless its response message returns Operation Successful.

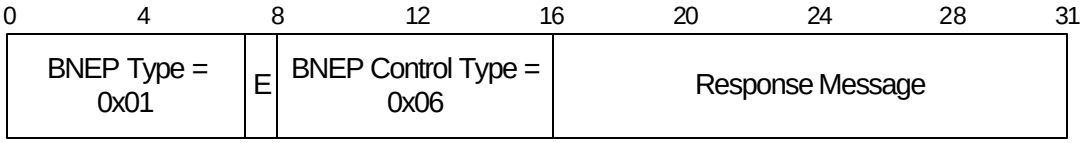

Figure 12: BNEP_FILTER_MULTI_ADDR_RESPONSE_MSG response message format

BNEP Type: *Size: 7 Bits*

Value	Parameter Description
0x01	Seven bit Bluetooth Network Encapsulation Protocol Type value identifies the type of BNEP header contained in this packet. MUST be set to BNEP_ CONTROL

Extension Flag (E): *Size: 1 Bit*

Value	Parameter Description
0x0 – 0x1	One bit extension flag that indicates if one or more extension headers follow the BNEP Header before the data payload. Extension headers are defined in section 3 on page 36. If the extension flag is equal to 0x1 then one or more extension headers follows the BNEP header. If the extension flag is equal to 0x0 then the BNEP payload follows the BNEP header.

BNEP Control Type: *Size: 1 Byte*

Value	Parameter Description
0x06	Type of BNEP control message contained in the packet. MUST be set to BNEP_FILTER_MULTI_ADDR_RESPONSE_MSG

Response Message: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	16 bit type field identifies the response to the previous Multicast Address Filter Control Message. Valid responses are contained in Table 9 on page 31.

2.6.6.2.1 Response Messages

Table 9 on page 31 contains a list the valid response messages to be used to respond to Multicast Address filter control messages.

Value	Response Messages
0x0000	Operation Successful
0x0001	Unsupported Request
0x0002	Operation Failed: Invalid Multicast Address
0x0003	Operation Failed: Maximum Multicast Address Filter Limited Reached
0x0004	Operation Failed: Unable to fulfill request due to security reasons.
0x0005 – 0xFFFF	Reserved for future use

Table 9: Multicast Address Filter Response Messages

2.7 BNEP_COMPRESSED_ETHERNET Packet Type Header Format

The BNEP_COMPRESSED_ETHERNET packet type header format is shown in Figure 15 on page 31. The header format is based on one of the compressed versions of the Ethernet header supported by BNEP. This packet type shall be used to carry Ethernet packets to and from devices that are directly connected at L2CAP level (have a valid L2CAP connection handle) using BNEP. This compressed header may be used when two Bluetooth devices are exchanging packets, in which the source address is set to the local device's address which is the source device sending the packet and destination addresses is set to the other device's address which is the final destination for the packet. Devices do not need to include the source or destination addresses in the packet because the destination address is always the device's address that received the packet and the source address is always the device's address that sent the packet.

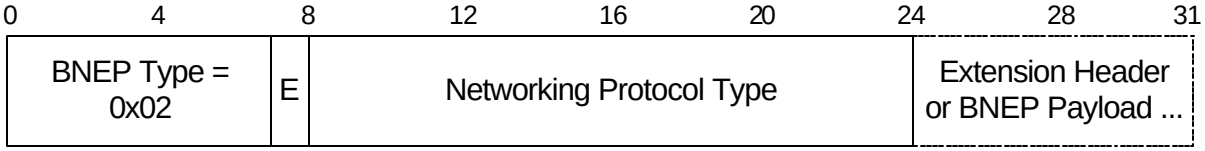

Figure 15: BNEP_COMPRESSED_ETHERNET Packet Type Header

BNEP Type: *Size: 7 Bit*

Value	Parameter Description

0x02	Seven bit Bluetooth Network Encapsulation Protocol Type value identifies the type of BNEP header contained in this packet. MUST be set to BNEP_COMPRESSED_ETHERNET

Extension Flag (E): *Size: 1 Bit*

Value	Parameter Description
0x0 – 0x1	One bit extension flag that indicates if one or more extension headers follow the BNEP Header before the data payload. Extension headers are defined in section 3 on page 36. If the extension flag is equal to 0x1 then one or more extension headers follows the BNEP header. If the extension flag is equal to 0x0 then the BNEP payload follows the BNEP header.

Networking Protocol Type: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	16 bit type field identifies the type of networking protocol contained in the payload. The values for this field are the same as defined for Ethernet types in [3]

2.8 BNEP_COMPRESSED_ETHERNET_SOURCE_ONLY Packet Type Header Format

The BNEP_COMPRESSED_ETHERNET_SOURCE_ONLY packet type header format is shown in Figure 14 on page 33. The header format is based on one of the compressed versions of the Ethernet header supported by BNEP. This packet type will be used to carry Ethernet packets to a device using BNEP, which is the final destination for that packet. Devices do not need to include the destination address in the packet, because the destination address of the BNEP packet is the same as the address corresponding to the L2CAP channel over which the packet is sent.

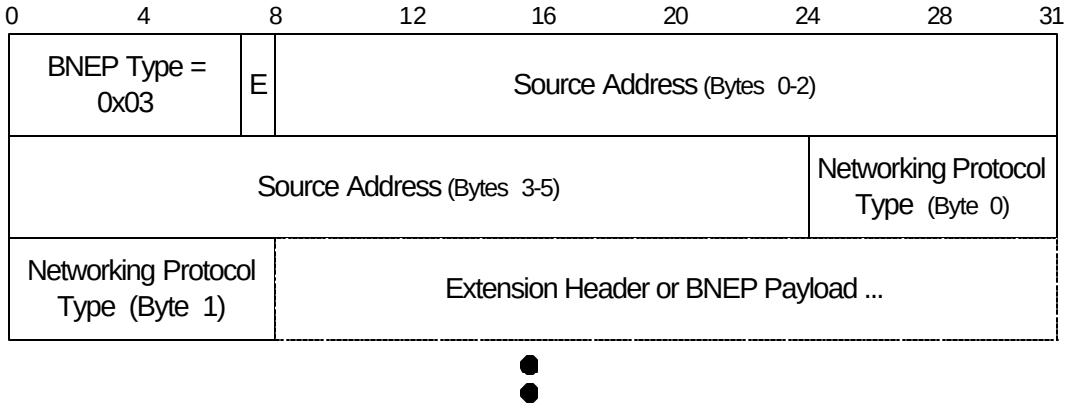

Figure 14: BNEP_COMPRESSED_ETHERNET_SOURCE_ONLY Packet Type Header

BNEP Type: *Size: 7 Bits*

Value	Parameter Description
0x03	Seven Bit Bluetooth Network Encapsulation Protocol Type value identifies the type of BNEP header contained in this packet. MUST be set to BNEP_COMPRESSED_ETHERNET_SOURCE_ONLY

Extension Flag (E): *Size: 1 Bit*

Value	Parameter Description
0x0 – 0x1	One bit extension flag that indicates if one or more extension headers follow the BNEP Header before the data payload. Extension headers are defined in section 3 on page 36. If the extension flag is equal to 0x1 then one or more extension headers follows the BNEP header. If the extension flag is equal to 0x0 then the BNEP payload follows the BNEP header.

Source Address: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	48 bit Bluetooth device address/IEEE address of the source of the BNEP packet/Ethernet frame contained in the payload.

Networking Protocol Type: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	16 bit type field identifies the type of networking

	protocol contained in the payload. The values for this field are the same as defined for Ethernet types in [3]

The source address may be an IEEE Ethernet address, if the actual source is an IEEE device and not a Bluetooth device.

2.9 BNEP_COMPRESSED_ETHERNET_DEST_ONLY Packet Type Header Format

The BNEP_COMPRESSED_ETHERNET_DEST_ONLY packet type header format is shown in Figure 16 on page 34. The header format is based on one of the compressed versions of the Ethernet header supported by BNEP. This packet type shall be used to carry Ethernet packets from a device using BNEP, which is the originator of that packet. Devices do not need to include the source address in the packet, because the source address can be determined from the L2CAP connection and which device sent the packet.

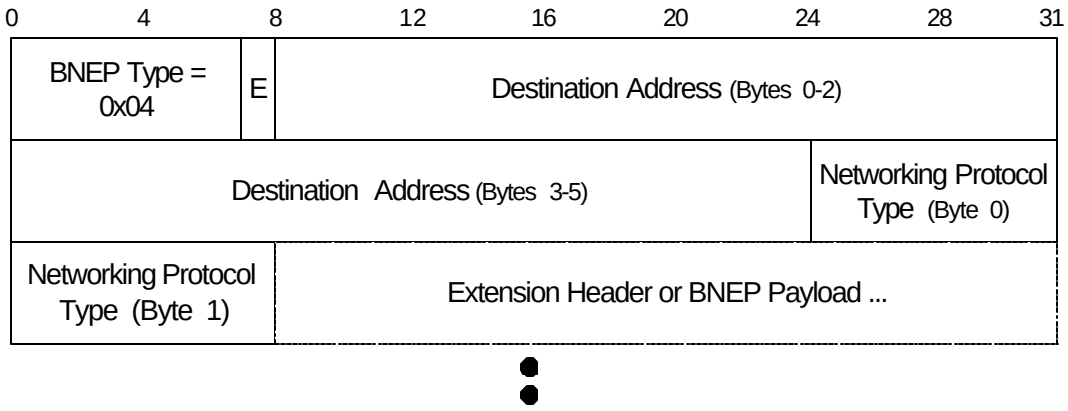

Figure 16: BNEP_COMPRESSED_ETHERNET_DEST_ONLY Packet Type Header

BNEP Type: *Size: 7 Bits*

Value	Parameter Description
0x04	Seven bit Bluetooth Network Encapsulation Protocol Type value identifies the type of BNEP header contained in this packet. MUST be set to BNEP_COMPRESSED_ETHERNET_DEST_ONLY

Extension Flag (E): *Size: 1 Bit*

Value	Parameter Description
0x0 – 0x1	One bit extension flag that indicates if one or more extension headers follow the BNEP Header before the

	data payload. Extension headers are defined in section 3 on page 36. If the extension flag is equal to 0x1 then one or more extension headers follows the BNEP header. If the extension flag is equal to 0x0 then the BNEP payload follows the BNEP header.

Destination Address: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	48 bit Bluetooth device address/IEEE address of the destination of the BNEP packet/Ethernet frame contained in the payload.

Networking Protocol Type: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	16 bit type field identifies the type of networking protocol contained in the payload. The values for this field are the same as defined for Ethernet types in [3]

The destination may be an IEEE Ethernet address, if the actual destination is an IEEE device and not a Bluetooth device.

3 Extension Header

3.1 Extension Header Overview

Extension headers are used as optional headers in addition to the BNEP header. One or more extension headers may be included after the BNEP header and before BNEP payload. If one or more extension headers are contained in the BNEP packet then, the Extension Flag in the BNEP must be used to indicate that extension header follows the BNEP header. If an additional extension header follows the current extension header, then the extension flag in the extension header must be used to indicate that an additional extension header follows. Every node must process every extension header. If the Extension Type is not understood, then that extension header must be skipped. If there are multiple similar or contradicting requests in one BNEP packet (either in the header or even in the payload), only the last one in the packet shall be processed.

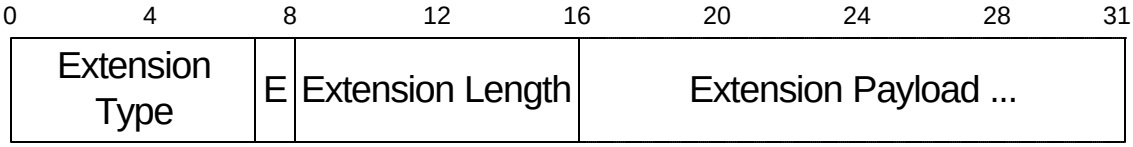

Figure 18 BNEP Extension Header Format

Extension Type: *Size: 7 Bits*

Value	Parameter Description
0x00 – 0x7F	One byte Bluetooth Network Encapsulation Protocol Extension Header Type value identifies the type of extension header contained in this packet. Values are defined in Table 11 on page 37.

Extension Flag (E): *Size: 1 Bit*

Value	Parameter Description
0x0 – 0x1	One bit extension flag that indicates if one or more extension headers follow the current extension header before the data payload. Extension headers are defined in section 3 on page 36. If the extension flag is equal to 0x1 then one or more extension headers follow the current extension header. If the extension flag is equal to 0x0 then the BNEP payload follows the current extension header.

Extension Length: *Size: 1 Byte*

Value	Parameter Description
0x00 – 0xFF	One byte extension length that defines the number of bytes contain in the extension payload. This byte count does not include bytes used for the extension type or the extension length.

Extension Payload: *Size: Based on Extension Type*

Value	Parameter Description
0xXX	Based on the Extension Type

3.2 Extension Type Values

The Table 11 on page 37 defines the various extension type formats

Value	Extension Packet Type
0x00	BNEP_EXTENSION_CONTROL
0x01 – 0x7F	Reserved for future use

Table 11: Extension Types

3.3 BNEP_EXTENSION_CONTROL Packet Type Header Format

The BNEP_EXTENSION_CONTROL packet type header format is shown in Figure 19 on page 38. This packet type is mandatory to recognize and respond to accordingly, whereas the implied functionality to do command is optional and does not have to be supported by all devices. The BNEP Control Type and the Control Packet parameters are defined in section 2.6 on page 16. BNEP extension headers of type BNEP_EXTENSION_CONTROL are for the device with the direct connection only, and must never be forwarded. BNEP_EXTENSION_CONTROL packet type headers in extension headers and BNEP_ CONTROL packets can be used interchangeably. Note: The size of each extension header is limited in length in terms of the maximum length of 255 bytes extension payload as well as the size of the entire BNEP packet, which is limited to the maximum MTU for the L2CAP connection.

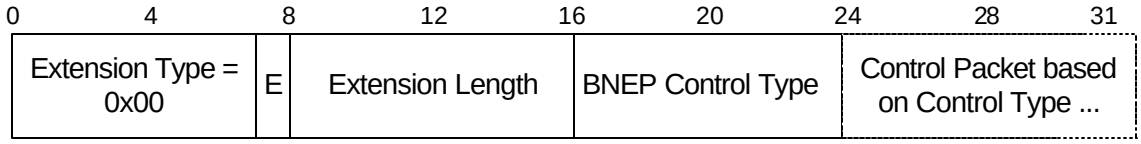

Figure 19: BNEP_EXTENSION_CONTROL Extension Header

4 Required Support for interpreting the IEEE 802.1p Header

4.1 IEEE 802.1p Support

IEEE 802.1p specification defines standardized frame prioritization tagging. In order to correctly determine the network protocol type, devices that implement the BNEP specification, MUST be able to interpret the IEEE 802.1p header, contained in the payload, in order to determine the actual network protocol type.

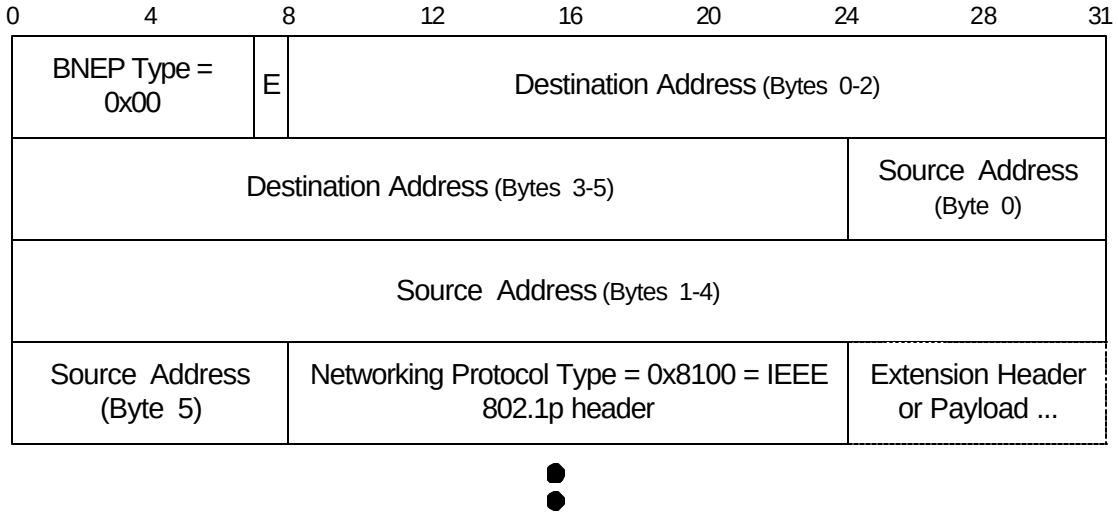

Figure 21: Example of a packet with an 802.1p protocol type contain in a BNEP header

The IEEE 802.1p header is contained in an additional 4 bytes in the Ethernet payload, as shown in the

Figure 22 on page 40 below, which is used for prioritization tagging. Devices should use the IEEE 802.1p header to determine prioritization of the packets.

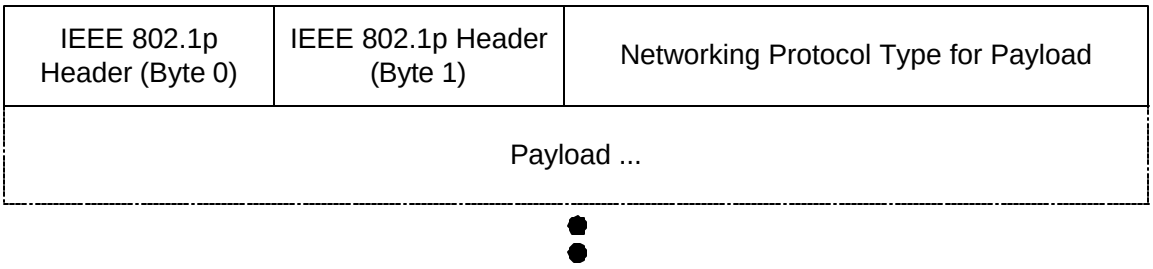

Figure 22: Ethernet payload for IEEE 802.1p packet

5 Examples

5.1 Example Overview

The following examples are used to illustrate some of the possible ways to use BNEP.

5.2 Sending an IP Packet Example

The following is a simple example in which an IP packet is sent using BNEP. The example illustrates an IPv4 packet sent from a device with 48 bit IEEE address of 00:AA:00:55:44:33 to a 48 bit Bluetooth address of 00:30:B7:45:67:89.

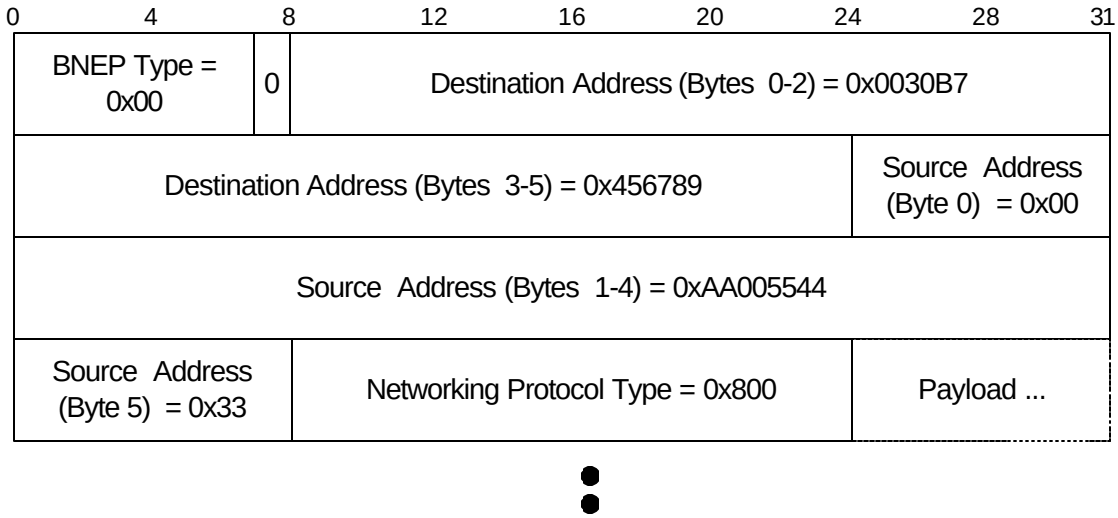

Figure 24: Sending an IP Packet Example

5.3 Sending an IP Packet between Bluetooth Master and Slave Example

The following is a simple example in which an IPv4 packet is sent using BNEP. In this example, the BNEP packet is sent from Device A, which is the master, to Device B, which is a slave of Device A. Device A has a 48 bit Bluetooth address of 00:AA:00:55:44:33 and Device B has a 48 bit Bluetooth Address of 00:30:B7:45:67:89.

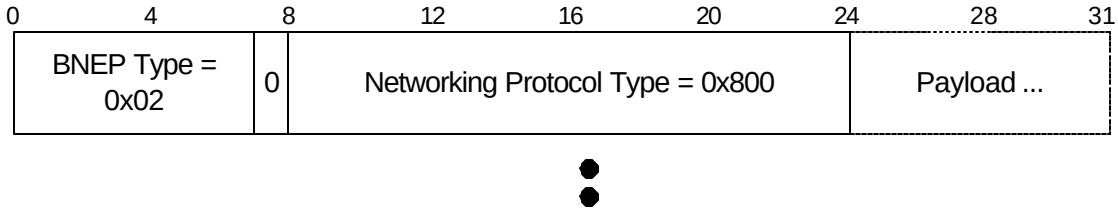

Figure 25: Sending an IP Packet between a Bluetooth Master and Slave Example

5.4 Setting Network Type Filter Examples

5.4.1 Enabling only IPv6

The following is a simple example for setting a filter to enable only IPv6.

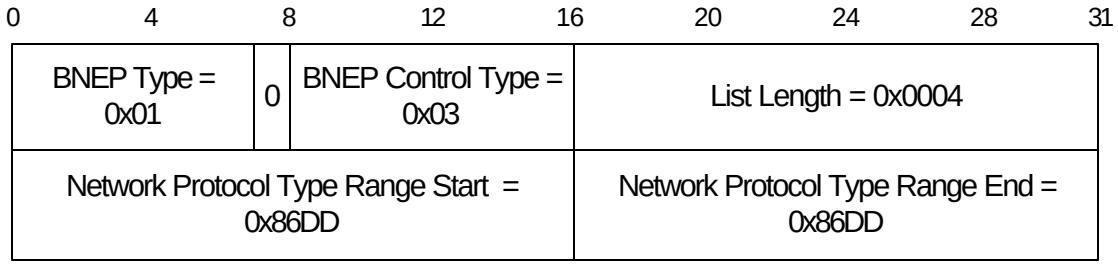

Figure 26: Setting Filter to Enable only IPv6 Example

5.4.2 Enabling only IPv4

The following is a simple example for setting a filter to enable only IPv4 (including ARP).

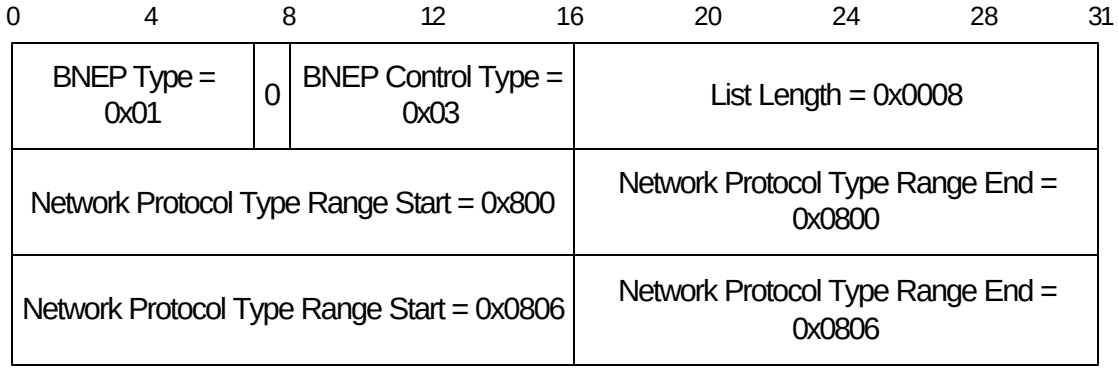

Figure 27: Setting Filter to enable only IPv4 and ARP

5.5 Setting Multicast Address Filter Examples

5.5.1 Enabling only IPv4 Multicast

The following is a simple example for setting a filter to enable only IPv4 IEEE Multicast Address (03-00-00-20-00-00).

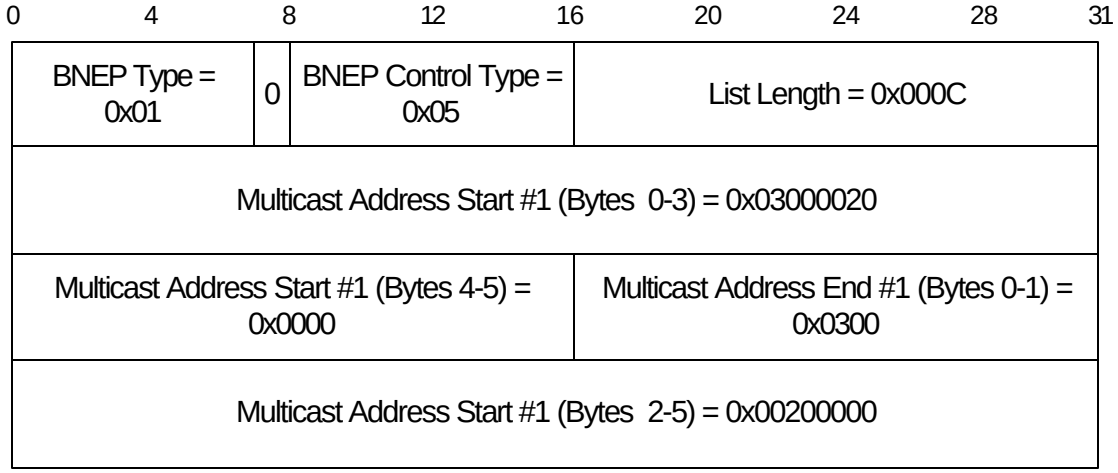

Figure 28: Setting Filter to Enable only IPv4 Multicast Example

5.5.2 Enabling only IPv6 Neighbor Discovery Multicast Address Range

The following is a simple example for setting a filter to enable only IPv6 neighbor discovery multicast address (33-33-00-00-00-00 to 33-33-FF-FF-FF-FF)

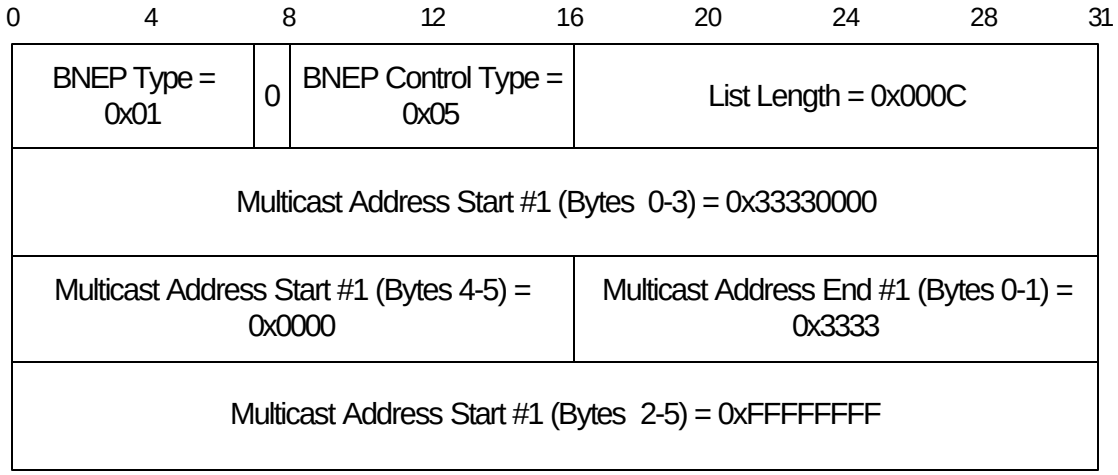

Figure 29: Setting Filter to enable only IPv6 Neighbor Discovery Multicast Address Range

5.6 Sending an IP Packet with one extension header Example

The following is an example extension in which an IPv6 packet is sent using BNEP and a BNEP filter extension header is also included in the packet. The example illustrates an IPv6 packet sent from a device with 48 bit IEEE address of 00:AA:00:55:44:33 to a 48 bit Bluetooth address of 00:30:B7:45:67:89. The filter control message in the extension header is set to enable only IPv6.

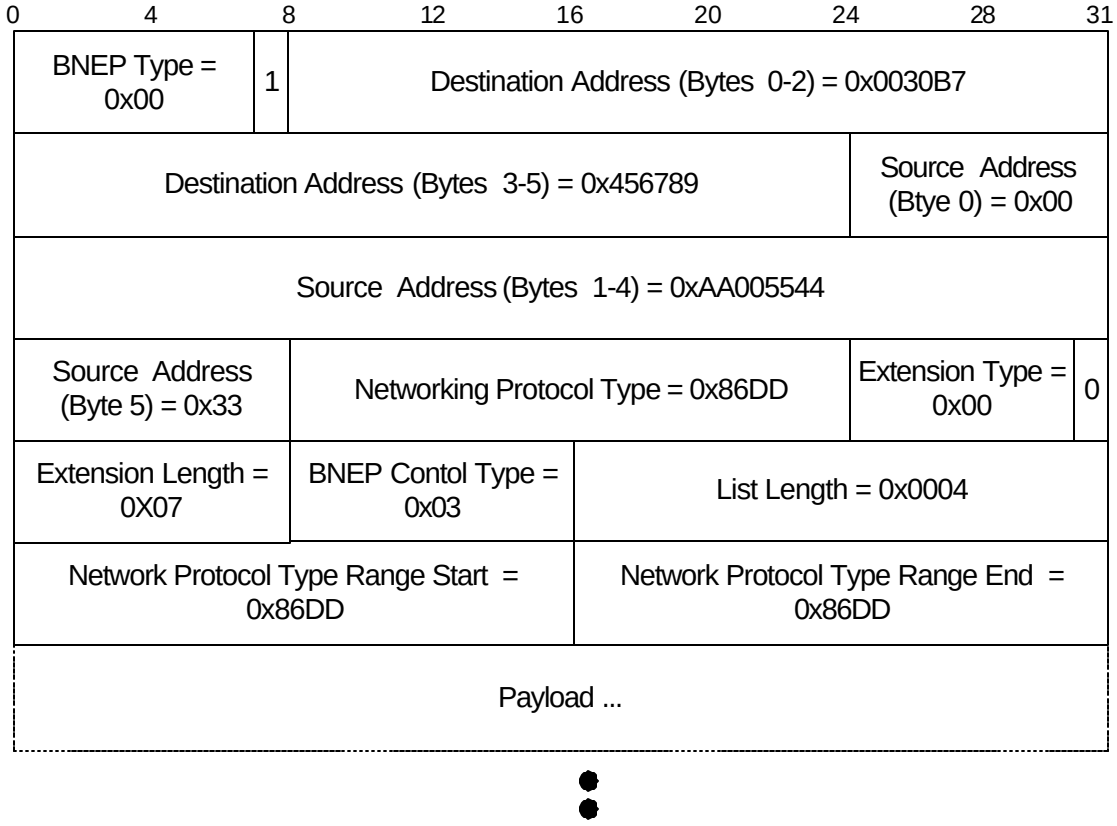

Figure 30: Sending an IP Packet with one extension Example

5.7 Sending an IP Packet between Bluetooth Master and Slave with one extension Example

The following is a simple example in which an IP packet is sent using BNEP using the BNEP_COMPRESSED_ETHERNET Packet Type Header format. In this example, the IPv4 packet is sent from Device A, which is the piconet master, to Device B, which is a slave of Device A. Device A has a 48 bit Bluetooth address of 00:AA:00:55:44:33 and Device B has a 48 bit Bluetooth Address of 00:30:B7:45:67:89. The filter control message in the extension header is set to enable only IPv4 (including ARP).

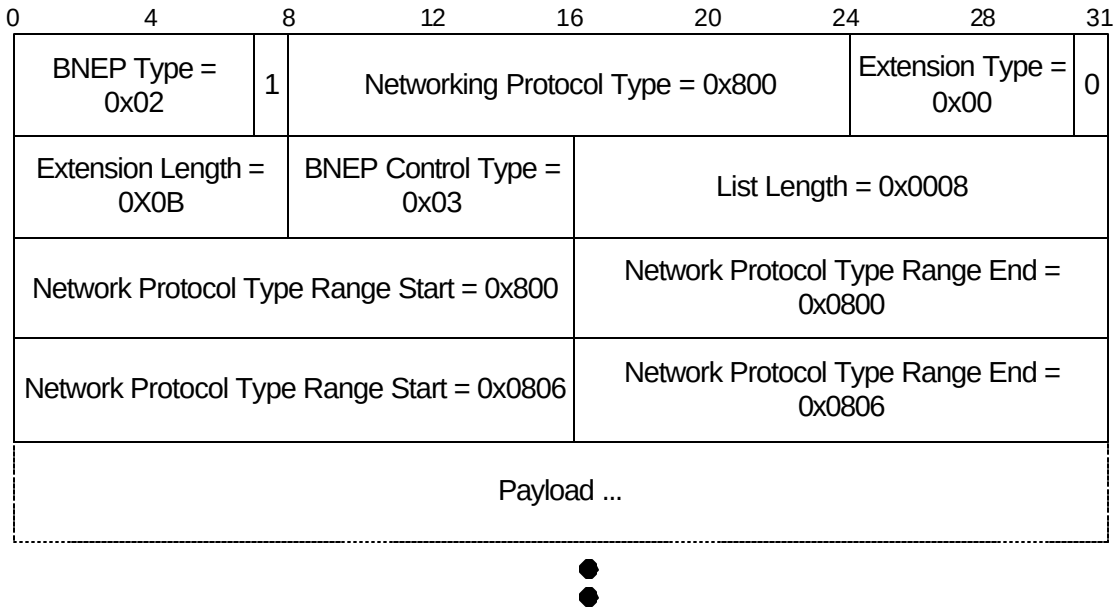

Figure 31: Sending an IP Packet between a Bluetooth Master and Slave with filter extension Example

6 Editor's Notes and Decision History

This section is to add a short history about key decisions. This section, then can be referred to prevent rehashing the same topics and to further progress.

1. For phase 1, of the PAN working group, Bluetooth Network Encapsulation Protocol as described in this document shall be used. For phase 2 (scatternet support), a layer 2 or layer 3 approach will be used.

2. The reserved BNEP packet types shall be used to extend the protocol.

3. The L2CAP packet length can be used to determine the total length of the BNEP packet. Therefore a length field in the BNEP header is redundant.

4. The extension header is defined for possible future extension of the BNEP.

7 References

[1] Bluetooth Special Interest Group, “Bluetooth Personal Area Networking Profiles”, Specification of the Bluetooth System, Version 0.95, May 6, 2001

[2] Bluetooth Special Interest Group, “Bluetooth Core”, Specification of the Bluetooth System, Version 1.1, February 22, 2001

[3] http://www.iana.org/assignments/ethernet-numbers

[4] “The Ethernet – A Local Area Network”, Version 1.0 Digital Equipment Corporation, Intel Corporation, Xerox Corporation. September 1980.

[5] Internet Engineering Task Force, “A Standard for the Transmission of IP Datagrams over Ethernet Networks”, RFC 894.

[6] Internet Engineering Task Force, “Classical IP and ARP over ATM”, RFC 2225, April 1998

[7] Internet Engineering Task Force, “IPv4 over IEEE 1394”, RFC2734, December 1999.

[8] Bluetooth Special Interest Group, “Bluetooth Assigned Number”, Specification of the Bluetooth System, Version 1.1 December 1, 2000

8 Acronyms and Abbreviations

List of abbreviations necessary for the understanding BNEP.

Abbreviation or Acronym	Meaning
BNEP	Bluetooth Network Encapsulation Protocol
IP	Internet Protocol
L2CAP	Logical Link Control and Adaptation Protocol
MTU	Maximum Transmission Unit
OSI	Open Systems Interconnect (model)
PAN	Personal Area Network

Table 13: Acronyms and Abbreviation Table

9 List of Figures

10 List of Tables