Network Working Group
Request For Comments: 1638
Category: Standards Track

F. Baker
ACC
R. Bowen
IBM
Editors
June 1994

PPP Bridging Control Protocol (BCP)

Status of this Memo

Abstract

The Point-to-Point Protocol (PPP) [6] 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.

This document defines the Network Control Protocol for establishing and configuring Remote Bridging for PPP links.

Table of Contents

1. Historical Perspective

Two basic algorithms are ambient in the industry for Bridging of Local Area Networks. The more common algorithm is called "Transparent Bridging", and has been standardized for Extended LAN configurations by IEEE 802.1. The other is called "Source Route Bridging", and is prevalent on IEEE 802.5 Token Ring LANs.

The IEEE has combined these two methods into a device called a Source Routing Transparent (SRT) bridge, which concurrently provides both Source Route and Transparent bridging. Transparent and SRT bridges are specified in IEEE standard 802.1D [3].

Although IEEE committee 802.1G is addressing remote bridging [2], neither standard directly defines the mechanisms for implementing remote bridging. Technically, that would be beyond the IEEE 802 committee's charter. However, both 802.1D and 802.1G allow for it. The implementor may model the line either as a component within a single MAC Relay Entity, or as the LAN media between two remote bridges.

2. Methods of Bridging

2.1. Transparent Bridging

As a favor to the uninitiated, let us first describe Transparent Bridging. Essentially, the bridges in a network operate as isolated entities, largely unaware of each others' presence. A Transparent Bridge maintains a Forwarding Database consisting of

{address, interface}

records, by saving the Source Address of each LAN transmission that it receives, along with the interface identifier for the interface it was received on. It goes on to check whether the Destination Address is in the database, and if so, either discards the message when the destination and source are located at the same interface, or forwards the message to the indicated interface. A message whose Destination Address is not found in the table is forwarded to all interfaces except the one it was received on. This behavior applies to Broadcast/Multicast frames as well.

The obvious fly in the ointment is that redundant paths in the network cause indeterminate (nay, all too determinate) forwarding behavior to occur. To prevent this, a protocol called the Spanning Tree Protocol is executed between the bridges to detect and logically remove redundant paths from the network.

One system is elected as the "Root", which periodically emits a message called a Bridge Protocol Data Unit (BPDU), heard by all of its neighboring bridges. Each of these modifies and passes the BPDU on to its neighbors, until it arrives at the leaf LAN segments in the network (where it dies, having no further neighbors to pass it along), or until the message is stopped by a bridge which has a superior path to the "Root". In this latter case, the interface the BPDU was received on is ignored (it is placed in a Hot Standby status, no traffic is emitted onto it except the BPDU, and all traffic received from it is discarded), until a topology change forces a recalculation of the network.

2.2. Remote Transparent Bridging

There exist two basic sorts of bridges -- those that interconnect LANs directly, called Local Bridges, and those that interconnect LANs via an intermediate medium such as a leased line, called Remote Bridges. PPP may be used to connect Remote Bridges.

The IEEE 802.1G Remote MAC Bridging committee has proposed a model of a Remote Bridge in which a set of two or more Remote Bridges that are

interconnected via remote lines are termed a Remote Bridge Group. Within a Group, a Remote Bridge Cluster is dynamically formed through execution of the spanning tree as the set of bridges that may pass frames among each other.

This model bestows on the remote lines the basic properties of a LAN, but does not require a one-to-one mapping of lines to virtual LAN segments. For instance, the model of three interconnected Remote Bridges, A, B and C, may be that of a virtual LAN segment between A and B and another between B and C. However, if a line exists between Remote Bridges B and C, a frame could actually be sent directly from B to C, as long as there was the external appearance that it had travelled through A.

IEEE 802.1G thus allows for a great deal of implementation freedom for features such as route optimization and load balancing, as long as the model is maintained.

For simplicity and because the 802.1G proposal has not been approved as a standard, we discuss Remote Bridging in this document in terms of two Remote Bridges connected by a single line. Within the 802.1G framework, these two bridges would comprise a Remote Bridge Group. This convention is not intended to preclude the use of PPP bridging in larger Groups, as allowed by 802.1G.

2.3. Source Routing

The IEEE 802.1D Committee has standardized Source Routing for any MAC Type that allows its use. Currently, MAC Types that support Source Routing are FDDI and IEEE 802.5 Token Ring.

The IEEE standard defines Source Routing only as a component of an SRT bridge. However, many bridges have been implemented which are capable of performing Source Routing alone. These are most commonly implemented in accordance either with the IBM Token-Ring Network Architecture Reference [1] or with the Source Routing Appendix of IEEE 802.1D [3].

In the Source Routing approach, the originating system has the responsibility of indicating the path that the message should follow. It does this, if the message is directed off of the local segment, by including a variable length MAC header extension called the Routing Information Field (RIF). The RIF consists of one 16-bit word of flags and parameters, followed by zero or more segment-and-bridge identifiers. Each bridge en route determines from this source route list whether it should accept the message and how to forward it.

In order to discover the path to a destination, the originating system transmits an Explorer frame. An All-Routes Explorer (ARE) frame follows all possible paths to a destination. A Spanning Tree Explorer (STE) frame follows only those paths defined by Bridge ports that the Spanning Tree Algorithm has put in Forwarding state. Port states do not apply to ARE or Specifically-Routed Frames. The destination system replies to each copy of an ARE frame with a Specifically-Routed Frame, and to an STE frame with an ARE frame. In either case, the originating station may receive multiple replies, from which it chooses the route it will use for future Specifically-Routed Frames.

The algorithm for Source Routing requires the bridge to be able to identify any interface by its segment-and-bridge identifier. When a packet is received that has the RIF present, a boolean in the RIF is inspected to determine whether the segment-and-bridge identifiers are to be inspected in "forward" or "reverse" sense. In its search, the bridge looks for the segment-and-bridge identifier of the interface the packet was received on, and forwards the packet toward the segment identified in the segment-and-bridge identifier that follows it.

2.4. Remote Source Route Bridging

There is no Remote Source Route Bridge proposal in IEEE 802.1 at this time, although many vendors ship remote Source Routing Bridges.

We allow for modelling the line either as a connection residing between two halves of a "split" Bridge (the split-bridge model), or as a LAN segment between two Bridges (the independent-bridge model). In the latter case, the line requires a LAN Segment ID.

By default, PPP Source Route Bridges use the independent-bridge model. This requirement ensures interoperability in the absence of option negotiation. In order to use the split-bridge model, a system MUST successfully negotiate the Bridge-Identification Configuration Option.

Although no option negotiation is required for a system to use the independent-bridge model, it is strongly recommended that systems using this model negotiate the Line-Identification Configuration Option. Doing so will verify correct configuration of the LAN Segment Id assigned to the line.

When two PPP systems use the split-bridge model, the system that transmits an Explorer frame onto the PPP link MUST update the RIF on behalf of the two systems. The purpose of this constraint is to ensure interoperability and to preserve the simplicity of the

bridging algorithm. For example, if the receiving system did not know whether the transmitting system had updated the RIF, it would have to scan the RIF and decide whether to update it. The choice of the transmitting system for the role of updating the RIF allows the system receiving the frame from the PPP link to forward the frame without processing the RIF.

Given that source routing is configured on a line or set of lines, the specifics of the link state with respect to STE frames are defined by the Spanning Tree Protocol in use. Choice of the split-bridge or independent-bridge model does not affect spanning tree operation. In both cases, the spanning tree protocol is executed on the two systems independently.

2.5. SR-TB Translational Bridging

IEEE 802 is not currently addressing bridges that translate between Transparent Bridging and Source Routing. For the purposes of this standard, such a device is either a Transparent or a Source Routing bridge, and will act on the line in one of these two ways, just as it does on the LAN.

3. Traffic Services

Several services are provided for the benefit of different system types and user configurations. These include LAN Frame Checksum Preservation, LAN Frame Checksum Generation, Tinygram Compression, and the identification of closed sets of LANs.

3.1. LAN Frame Checksum Preservation

IEEE 802.1 stipulates that the Extended LAN must enjoy the same probability of undetected error that an individual LAN enjoys. Although there has been considerable debate concerning the algorithm, no other algorithm has been proposed than having the LAN Frame Checksum received by the ultimate receiver be the same value calculated by the original transmitter. Achieving this requires, of course, that the line protocols preserve the LAN Frame Checksum from end to end. The protocol is optimized towards this approach.

3.2. Traffic having no LAN Frame Checksum

The fact that the protocol is optimized towards LAN Frame Checksum preservation raises twin questions: "What is the approach to be used by systems which, for whatever reason, cannot easily support Frame Checksum preservation?" and "What is the approach to be used when the system originates a message, which therefore has no Frame Checksum precalculated?".

Surely, one approach would be to require stations to calculate the Frame Checksum in software if hardware support were unavailable; this would meet with profound dismay, and would raise serious questions of interpretation in a Bridge/Router.

However, stations which implement LAN Frame Checksum preservation must already solve this problem, as they do originate traffic. Therefore, the solution adopted is that messages which have no Frame Checksum are tagged and carried across the line.

When a system which does not implement LAN Frame Checksum preservation receives a frame having an embedded FCS, it converts it for its own use by removing the trailing four octets. When any system forwards a frame which contains no embedded FCS to a LAN, it forwards it in a way which causes the FCS to be calculated.

3.3. Tinygram Compression

An issue in remote Ethernet bridging is that the protocols that are most attractive to bridge are prone to problems on low speed (64 KBPS and below) lines. This can be partially alleviated by observing that the vendors defining these protocols often fill the PDU with octets of ZERO. Thus, an Ethernet or IEEE 802.3 PDU received from a line that is (1) smaller than the minimum PDU size, and (2) has a LAN Frame Checksum present, must be padded by inserting zeroes between the last four octets and the rest of the PDU before transmitting it on a LAN. These protocols are frequently used for interactive sessions, and therefore are frequently this small.

To prevent ambiguity, PDUs requiring padding are explicitly tagged. Compression is at the option of the transmitting station, and is probably performed only on low speed lines, perhaps under configuration control.

The pseudo-code in Appendix 1 describes the algorithms.

3.4. LAN Identification

In some applications, it is useful to tag traffic by the user community it is a part of, and guarantee that it will be only emitted onto a LAN which is of the same community. The user community is defined by a LAN ID. Systems which choose to not implement this feature must assume that any frame received having a LAN ID is from a different community than theirs, and discard it.

It should be noted that the enabling of the LAN Identification option requires behavior consistent with the following additions to the standard bridging algorithm.

Each bridge port may be considered to have two additional variables associated with it: "domain" and "checkDomain".

The variable "domain" (a 32-bit unsigned integer) is assigned a value that uniquely labels a set of bridge ports in an extended network, with a default value of 1, and the values of 0 and 0xffffffff being reserved.

The variable "checkDomain" (a boolean) controls whether this value is used to filter output to a bridge port. The variable "checkDomain" is generally set to the boolean value True for LAN bridge ports, and set to the boolean value False for WAN bridge ports.

The action of the bridge is then as modified as expressed in the following C code fragments:

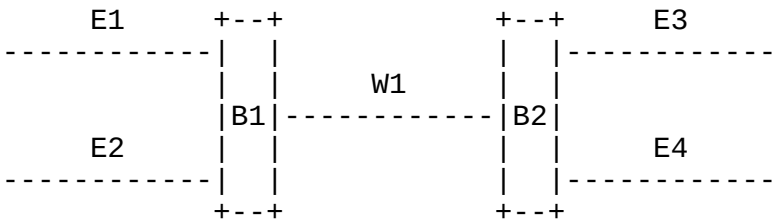
On a packet being received from a bridge port:

```
if (domainNotPresentWithPacket) {
    packetInformation.domain = portInformation[inputPort].domain;
} else {
    packetInformation.domain = domainPresentWithPacket;
}
```

On a packet being transmitted from a bridge port:

```
if (portInformation[outputPort].checkDomain &&
    portInformation[outputPort] != packetInformation.domain) {
    discardPacket();
    return;
}
```

For example, suppose you have the following configuration:

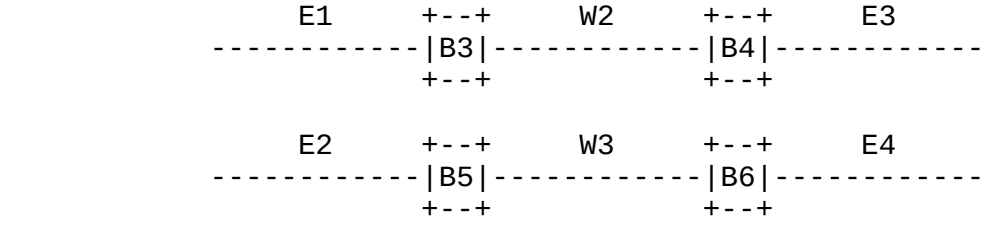


E1, E2, E3, and E4 are Ethernet LANs (or Token Ring, FDDI, etc.). W1 is a WAN (PPP over T1). B1 and B2 are MAC level bridges.

You want End Stations on E1 and E3 to communicate, and you want End Stations on E2 and E4 to communicate, but you do not want End Stations on E1 and E3 to communicate with End Stations on E2 and E4.

This is true for Unicast, Multicast, and Broadcast traffic. If a broadcast datagram originates on E1, you want it only to be propagated to E3, and not on E2 or E4.

Another way of looking at it is that E1 and E3 form a Virtual LAN, and E2 and E4 form a Virtual LAN, as if the following configuration were actually being used:



To accomplish this (using the LAN Identification option), B1 and B2 negotiate this option on, and send datagrams with bit 6 set to 1, with the LAN ID field inserted in the frame. Traffic on E1 and E3 would be assigned LAN ID 1, and traffic on E2 and E4 would be assigned LAN ID 2. Thus B1 and B2 can separate traffic going over W1.

Note that execution of the spanning tree algorithm may result in the subdivision of a domain. The administrator of LAN domains must ensure, through spanning tree configuration and topology design, that such subdivision does not occur.

4. A PPP Network Control Protocol for Bridging

The Bridging Control Protocol (BCP) is responsible for configuring, enabling and disabling the bridge protocol modules on both ends of the point-to-point link. BCP uses the same packet exchange mechanism as the Link Control Protocol. BCP packets may not be exchanged until PPP has reached the Network-Layer Protocol phase. BCP packets received before this phase is reached SHOULD be silently discarded.

The Bridging Control Protocol is exactly the same as the Link Control Protocol [6] 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.

Implementations SHOULD NOT negotiate Address-and-Control-Field-Compression or Protocol-Field-Compression on other than low speed links.

Data Link Layer Protocol Field

Exactly one BCP packet is encapsulated in the PPP Information field, where the PPP Protocol field indicates type hex 8031 (BCP).

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

BCP packets may 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.

Configuration Option Types

BCP has a distinct set of Configuration Options, which are defined in this document.

4.1. Sending Bridge Frames

Before any Bridged LAN Traffic or BPDUs may be communicated, PPP MUST reach the Network-Layer Protocol phase, and the Bridging Control Protocol MUST reach the Opened state.

Exactly one Bridged LAN Traffic or BPDU is encapsulated in the PPP Information field, where the PPP Protocol field indicates type hex 0031 (Bridged PDU).

4.1.1. Maximum Receive Unit Considerations

The maximum length of a Bridged datagram transmitted over a PPP link is the same as the maximum length of the Information field of a PPP encapsulated packet. Since there is no standard method for fragmenting and reassembling Bridged PDUs, PPP links supporting Bridging MUST negotiate an MRU large enough to support the MAC Types that are later negotiated for Bridging support. Because they include the MAC headers, even bridged Ethernet frames are larger than the default PPP MRU of 1500 octets.

4.1.2. Loopback and Link Quality Monitoring

It is strongly recommended that PPP Bridge Protocol implementations utilize Magic Number Loopback Detection and Link-Quality-Monitoring. The 802.1 Spanning Tree protocol, which is integral to both Transparent Bridging and Source Routing (as standardized), is unidirectional during normal operation. Configuration BPDUs emanate from the Root system in the general direction of the leaves, without any reverse traffic except in response to network events.

4.1.3. Message Sequence

The multiple link case requires consideration of message sequentiality. The transmitting system may determine either that the protocol being bridged requires transmissions to arrive in the order of their original transmission, and enqueue all transmissions on a given conversation onto the same link to force order preservation, or that the protocol does NOT require transmissions to arrive in the order of their original transmission, and use that knowledge to optimize the utilization of several links, enqueuing traffic to multiple links to minimize delay.

In the absence of such a determination, the transmitting system MUST act as though all protocols require order preservation. Many protocols designed primarily for use on a single LAN require order preservation.

Work is currently in progress on a protocol to allow use of multiple PPP links [7]. If approved, this protocol will allow use of multiple links while maintaining message sequentiality for Bridged LAN Traffic and BPDU frames.

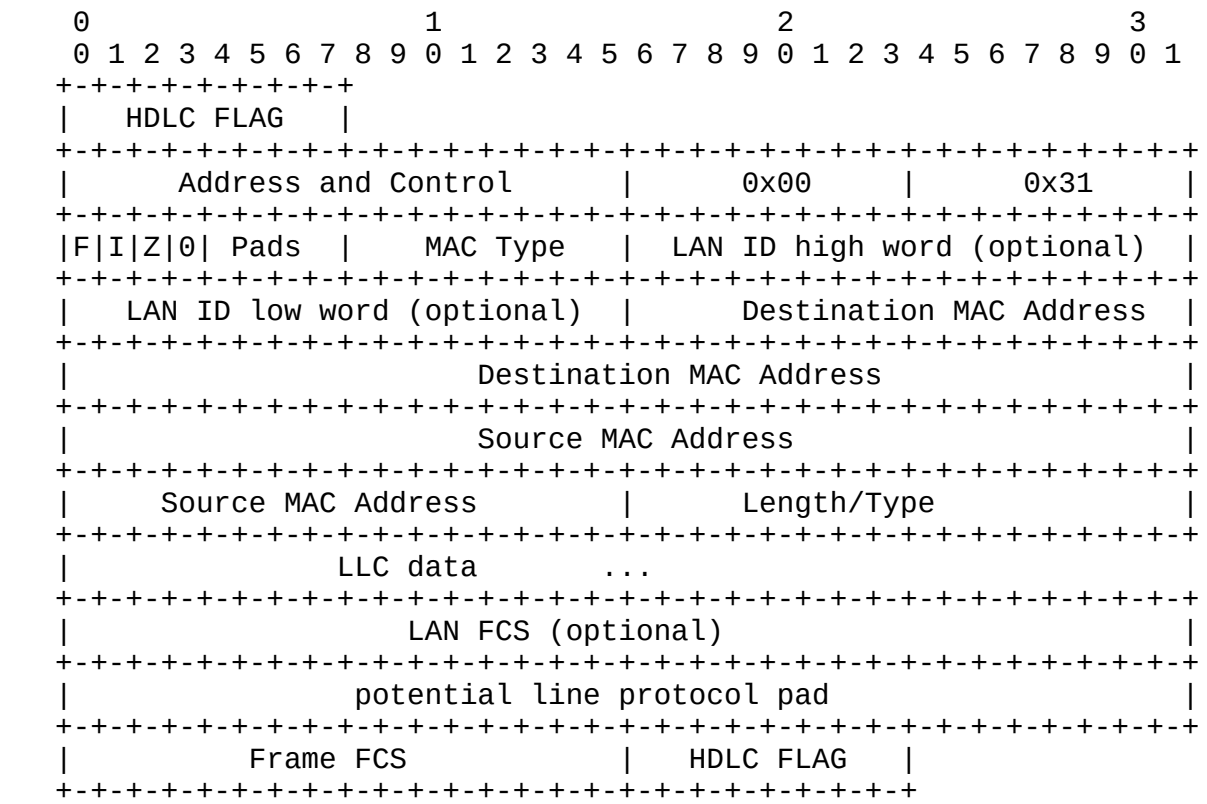
4.1.4. Separation of Spanning Tree Domains

It is conceivable that a network manager might wish to inhibit the exchange of BPDUs on a link in order to logically divide two regions into separate Spanning Trees with different Roots (and potentially different Spanning Tree implementations or algorithms). In order to do that, he should configure both ends to not exchange BPDUs on a link. An implementation that does not support any spanning tree protocol MUST silently discard any received IEEE 802.1D BPDU packets, and MUST either silently discard or respond to other received BPDU packets with an LCP Protocol-Reject packet.

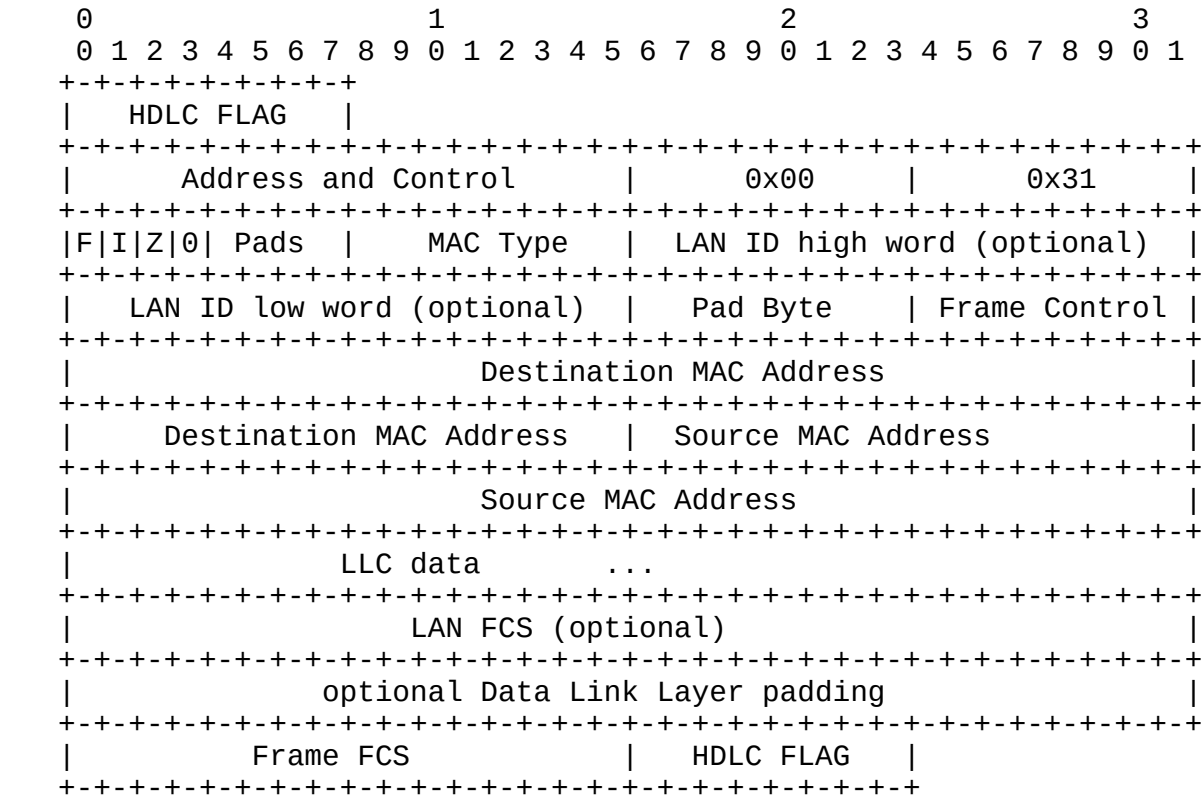
4.2. Bridged LAN Traffic

For Bridging LAN traffic, the format of the frame on the line is shown below. The fields are transmitted from left to right.

802.3 Frame format



802.4/802.5/FDDI Frame format



Address and Control

As defined by the framing in use.

PPP Protocol

0x0031 for PPP Bridging

Flags

bit F: Set if the LAN FCS Field is present
bit I: Set if the LAN ID Field is present
bit Z: Set if IEEE 802.3 Pad must be zero filled to minimum size
bit 0: reserved, must be zero

Pads

Any PPP frame may have padding inserted in the "Optional Data Link Layer Padding" field. This number tells the receiving system how many pad octets to strip off.

MAC Type

Up-to-date values of the MAC Type field are specified in the most recent "Assigned Numbers" RFC [4]. Current values are assigned as follows:

```
    0: reserved
    1: IEEE 802.3/Ethernet  with canonical addresses
    2: IEEE 802.4           with canonical addresses
    3: IEEE 802.5           with non-canonical addresses
    4: FDDI                 with non-canonical addresses
 5-10: reserved
   11: IEEE 802.5           with canonical addresses
   12: FDDI                 with canonical addresses
```

"Canonical" is the address format defined as standard address representation by the IEEE. In this format, the bit within each byte that is to be transmitted first on a LAN is represented as the least significant bit. In contrast, in non-canonical form, the bit within each byte that is to be transmitted first is represented as the most-significant bit. Many LAN interface implementations use non-canonical form. In both formats, bytes are represented in the order of transmission.

If an implementation supports a MAC Type that is the higher-numbered format of that MAC Type, then it MUST also support the lower-numbered format of that MAC Type. For example, if an implementation supports FDDI with canonical address format, then it MUST also support FDDI with non-canonical address format. The purpose of this requirement is to provide backward compatibility with earlier versions of this specification.

A system MUST NOT transmit a MAC Type numbered higher than 4 unless it has received from its peer a MAC-Support Configuration Option indicating that the peer is willing to receive frames of that MAC Type.

LAN ID

This optional 32-bit field identifies the Community of LANs which may be interested to receive this frame. If the LAN ID flag is not set, then this field is not present, and the PDU is four octets shorter.

Frame Control

On 802.4, 802.5, and FDDI LANs, there are a few octets preceding the Destination MAC Address, one of which is protected by the FCS.

The MAC Type of the frame determines the contents of the Frame Control field. A pad octet is present to provide 32-bit packet alignment.

Destination MAC Address

As defined by the IEEE. The MAC Type field defines the bit ordering.

Source MAC Address

As defined by the IEEE. The MAC Type field defines the bit ordering.

LLC data

This is the remainder of the MAC frame which is (or would be were it present) protected by the LAN FCS.

For example, the 802.5 Access Control field, and Status Trailer are not meaningful to transmit to another ring, and are omitted.

LAN FCS

If present, this is the LAN FCS which was calculated by (or which appears to have been calculated by) the originating station. If the LAN FCS flag is not set, then this field is not present, and the PDU is four octets shorter.

Optional Data Link Layer Padding

Any PPP frame may have padding inserted between the Information field and the Frame FCS. The Pads field contains the length of this padding, which may not exceed 15 octets.

The PPP LCP Extensions [5] specify a self-describing pad. Implementations are encouraged to set the Pads field to zero, and use the self-describing pad instead.

Frame FCS

Mentioned primarily for clarity. The FCS used on the PPP link is separate from and unrelated to the LAN FCS.

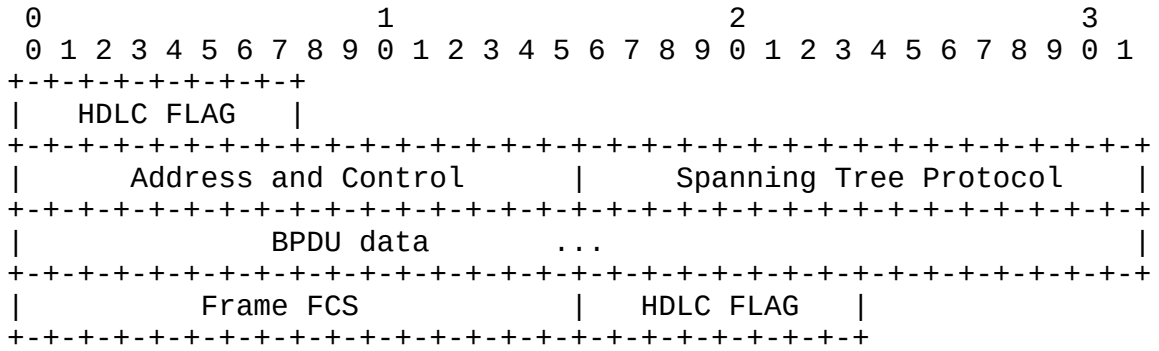
4.3. Spanning Tree Bridge PDU

This is the Spanning Tree BPDU, without any MAC or 802.2 LLC header (these being functionally equivalent to the Address, Control, and PPP Protocol Fields). The LAN Pad and Frame Checksum fields are likewise superfluous and absent.

The Address and Control Fields are subject to LCP Address-and-Control-Field-Compression negotiation.

A PPP system which is configured to participate in a particular spanning tree protocol and receives a BPDU of a different spanning tree protocol SHOULD reject it with the LCP Protocol-Reject. A system which is configured not to participate in any spanning tree protocol MUST silently discard all BPDUs.

Spanning Tree Bridge PDU



Address and Control

As defined by the framing in use.

Spanning Tree Protocol

Up-to-date values of the Spanning-Tree-Protocol field are specified in the most recent "Assigned Numbers" RFC [4]. Current values are assigned as follows:

Value (in hex)	Protocol
0201	IEEE 802.1 (either 802.1D or 802.1G)
0203	IBM Source Route Bridge
0205	DEC LANbridge 100

The two versions of the IEEE 802.1 spanning tree protocol frames can be distinguished by fields within the BPDU data.

BPDU data

As defined by the specified Spanning Tree Protocol.

5. BCP Configuration Options

BCP 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.

BCP uses the same Configuration Option format defined for LCP [6], with a separate set of Options.

Up-to-date values of the BCP Option Type field are specified in the most recent "Assigned Numbers" RFC [4]. Current values are assigned as follows:

1	Bridge-Identification
2	Line-Identification
3	MAC-Support
4	Tinygram-Compression
5	LAN-Identification
6	MAC-Address
7	Spanning-Tree-Protocol

5.1. Bridge-Identification

Description

The Bridge-Identification Configuration Option is designed for use when the line is an interface between half bridges connecting virtual or physical LAN segments. Since these remote bridges are modeled as a single bridge with a strange internal interface, each remote bridge needs to know the LAN segment and bridge numbers of the adjacent remote bridge. This option MUST NOT be included in the same Configure-Request as the Line-Identification option.

The Source Routing Route Descriptor and its use are specified by the IEEE 802.1D Appendix on Source Routing. It identifies the segment to which the interface is attached by its configured segment number, and itself by bridge number on the segment.

The two half bridges MUST agree on the bridge number. If a bridge number is not agreed upon, the Bridging Control Protocol MUST NOT enter the Opened state.

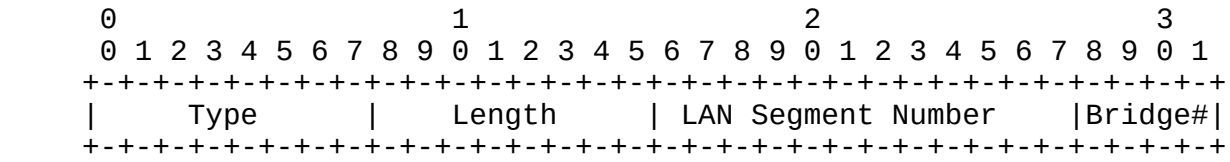
Since mismatched bridge numbers are indicative of a configuration error, it is strongly recommended that a system not change its bridge number for the purpose of resolving a mismatch. However, to allow two systems to proceed to the Opened state despite a mismatch, a system MAY change its bridge number to the higher of the two numbers. A higher-numbered system MUST NOT change its bridge number to a lower number.

By default, a system that does not negotiate this option is assumed to be configured not to use the model of the two systems as two halves of a single source-route bridge. It is instead assumed to be configured to use the model of the two systems as two independent bridges.

Example

If System A announces LAN Segment AAA, Bridge #1, and System B announces LAN Segment BBB, Bridge #1, then the resulting Source Routing configuration (read in the appropriate direction) is then AAA,1,BBB.

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



Type

1

Length

4

LAN Segment Number

A 12-bit number identifying the LAN segment, as defined in the IEEE 802.1D Source Routing Specification.

Bridge Number

A 4-bit number identifying the bridge on the LAN segment, as defined in the IEEE 802.1D Source Routing Specification.

5.2. Line-Identification

Description

The Line-Identification Configuration Option is designed for use when the line is assigned a LAN segment number as though it were a two system LAN segment in accordance with the Source Routing algorithm. This option MUST NOT be included in the same Configure-Request as the Bridge-Identification option.

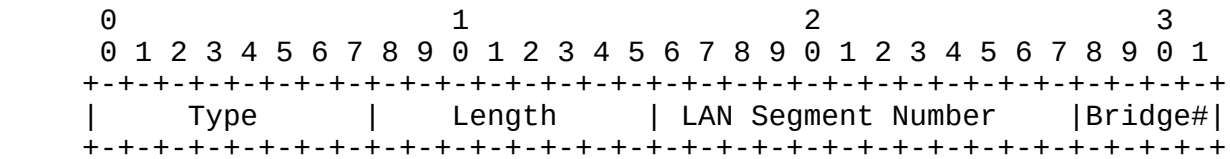
The Source Routing Route Descriptor and its use are specified by the IEEE 802.1D Appendix on Source Routing. It identifies the segment to which the interface is attached by its configured segment number, and itself by bridge number on the segment.

The two bridges MUST agree on the LAN segment number. If a LAN segment number is not agreed upon, the Bridging Control Protocol MUST NOT enter the Opened state.

Since mismatched LAN segment numbers are indicative of a configuration error, it is strongly recommended that a system not change its LAN segment number for the purpose of resolving a mismatch. However, to allow two systems to proceed to the Opened state despite a mismatch, a system MAY change its LAN segment number to the higher of the two numbers. A higher-numbered system MUST NOT change its LAN segment number to a lower number.

By default, a system that does not negotiate this option is assumed to have its LAN segment number correctly configured by the user.

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



Type

2

Length

4

LAN Segment Number

A 12-bit number identifying the LAN segment, as defined in the IEEE 802.1D Source Routing Specification.

Bridge Number

A 4-bit number identifying the bridge on the LAN segment, as defined in the IEEE 802.1D Source Routing Specification.

5.3. MAC-Support

Description

The MAC-Support Configuration Option is provided to permit implementations to indicate the sort of traffic they are prepared to receive. Negotiation of this option is strongly recommended.

By default, when an implementation does not announce the MAC Types that it supports, all MAC Types are sent by the peer which are capable of being transported given other configuration parameters. The receiver will discard those MAC Types that it does not support.

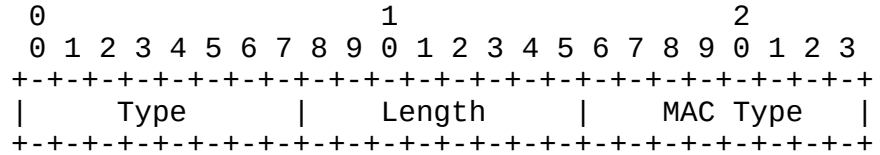
A device supporting a 1600 octet MRU might not be willing to support 802.5, 802.4 or FDDI, which each support frames larger than 1600 octets.

By announcing the MAC Types it will support, an implementation is advising its peer that all unspecified MAC Types will be discarded. The peer MAY then reduce bandwidth usage by not sending the unsupported MAC Types.

Announcement of support for multiple MAC Types is accomplished by placing multiple options in the Configure-Request.

The nature of this option is advisory only. This option MUST NOT be included in a Configure-Nak.

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



Type

3

Length

3

MAC Type

One of the values of the PDU MAC Type field (previously described in the "Bridged LAN Traffic" section) that this system is prepared to receive and service.

5.4. Tinygram-Compression

Description

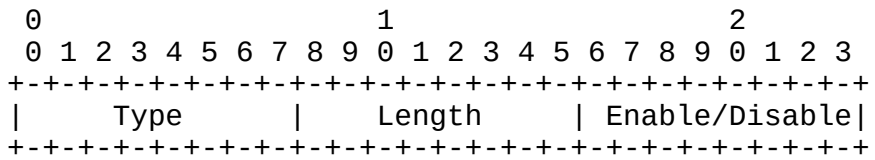
This Configuration Option permits the implementation to indicate support for Tinygram compression.

Not all systems are prepared to make modifications to messages in transit. On high speed lines, it is probably not worth the effort.

This option MUST NOT be included in a Configure-Nak if it has been received in a Configure-Request. This option MAY be included in a Configure-Nak in order to prompt the peer to send the option in its next Configure-Request.

By default, no compression is allowed. A system which does not negotiate, or negotiates this option to be disabled, should never receive a compressed packet.

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



Type

4

Length

3

Enable/Disable

If the value is 1, Tinygram-Compression is enabled. If the value is 2, Tinygram-Compression is disabled, and no decompression will occur.

The implementations need not agree on the setting of this parameter. One may be willing to decompress and the other not.

5.5. LAN-Identification

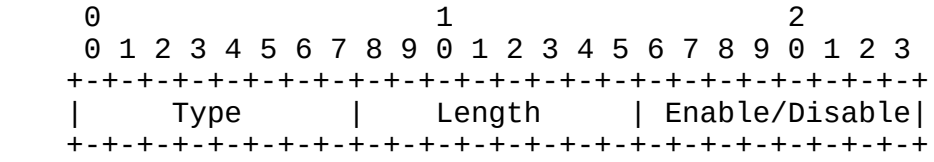
Description

This Configuration Option permits the implementation to indicate support for the LAN Identification field, and that the system is prepared to service traffic to any labeled LANs beyond the system.

A Configure-NAK MUST NOT be sent in response to a Configure-Request that includes this option.

By default, LAN-Identification is disabled. All Bridge LAN Traffic and BPDUs that contain the LAN ID field will be discarded. The peer may then reduce bandwidth usage by not sending the unsupported traffic.

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



Type

5

Length

3

Enable/Disable

If the value is 1, LAN Identification is enabled. If the value is 2, LAN Identification is disabled.

The implementations need not agree on the setting of this parameter. One may be willing to accept LAN Identification and the other not.

5.6. MAC-Address

Description

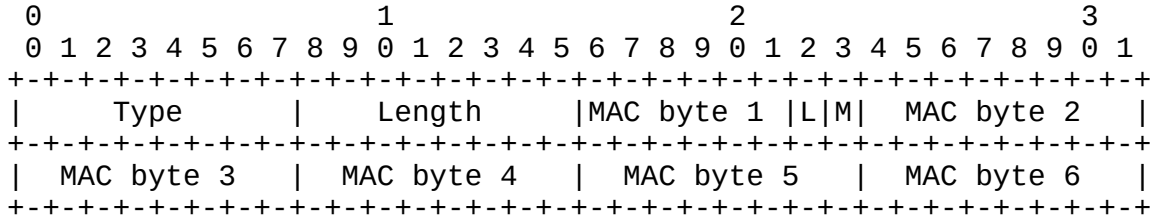
The MAC-Address Configuration Option enables the implementation to announce its MAC address or have one assigned. The MAC address is represented in IEEE 802.1 Canonical format, which is to say that the multicast bit is the least significant bit of the first octet of the address.

If the system wishes to announce its MAC address, it sends the option with its MAC address specified. When specifying a non-zero MAC address in a Configure-Request, any inclusion of this option in a Configure-Nak MUST be ignored.

If the implementation wishes to have a MAC address assigned, it sends the option with a MAC address of 00-00-00-00-00-00. Systems that have no mechanism for address assignment will Configure-Reject the option.

A Configure-Nak MUST specify a valid IEEE 802.1 format physical address; the multicast bit MUST be zero. It is strongly recommended (although not mandatory) that the "locally assigned address" bit (the second least significant bit in the first octet) be set, indicating a locally assigned address.

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



Type

6

Length

8

MAC Byte

Six octets of MAC address in 802.1 Canonical order. For clarity, the position of the Local Assignment (L) and Multicast (M) bits are shown in the diagram.

5.7. Spanning-Tree-Protocol

Description

The Spanning-Tree-Protocol Configuration Option enables the Bridges to negotiate the version of the spanning tree protocol in which they will participate.

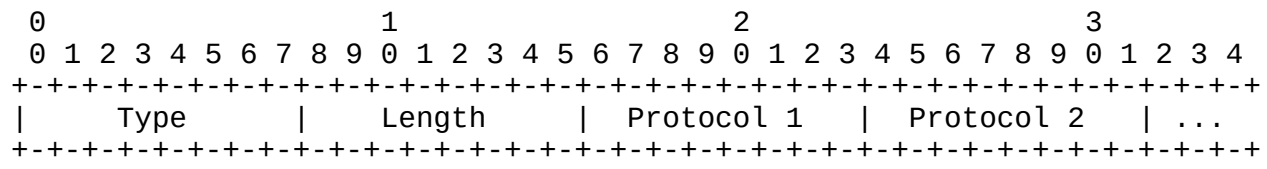
If both bridges support a spanning tree protocol, they MUST agree on the protocol to be supported. When the two disagree, the lower-numbered of the two spanning tree protocols should be used. To resolve the conflict, the system with the lower-numbered protocol SHOULD Configure-Nak the option, suggesting its own protocol for use. If a spanning tree protocol is not agreed upon, except for the case in which one system does not support any spanning tree protocol, the Bridging Control Protocol MUST NOT enter the Opened state.

Most systems will only participate in a single spanning tree protocol. If a system wishes to participate simultaneously in more than one spanning tree protocol, it MAY include all of the appropriate protocol types in a single Spanning-Tree-Protocol Configuration Option. The protocol types MUST be specified in increasing numerical order. For the purpose of comparison during negotiation, the protocol numbers MUST be considered to be a single number. For instance, if System A includes protocols 01 and 03 and System B indicates protocol 03, System B should Configure-Nak and indicate a protocol type of 03 since 0103 is greater than 03.

By default, an implementation MUST either support the IEEE 802.1D spanning tree or support no spanning tree protocol. An implementation that does not support any spanning tree protocol MUST silently discard any received IEEE 802.1D BPDU packets, and

MUST either silently discard or respond to other received BPDU packets with an LCP Protocol-Reject packet.

A summary of the Spanning-Tree-Protocol Option format is shown below. The fields are transmitted from left to right.



Type

7

Length

2 octets plus 1 additional octet for each protocol that will be actively supported. Most systems will only support a single spanning tree protocol, resulting in a length of 3.

Protocol n

Each Protocol field is one octet and indicates a desired spanning tree protocol. Up-to-date values of the Protocol field are specified in the most recent "Assigned Numbers" RFC [4]. Current values are assigned as follows:

Value	Protocol
0	Null (no Spanning Tree protocol supported)
1	IEEE 802.1D spanning tree
2	IEEE 802.1G extended spanning tree protocol
3	IBM Source Route Spanning tree protocol
4	DEC LANbridge 100 Spanning tree protocol

A. Tinygram-Compression Pseudo-Code

PPP Transmitter:

```
if (ZeroPadCompressionEnabled &&
    BridgedProtocolHeaderFormat == IEEE8023 &&
    PacketLength == Minimum8023PacketLength) {
 /*
  * Remove any continuous run of zero octets preceding,
  * but not including, the LAN FCS, but not extending
  * into the MAC header.
  */
    Set (ZeroCompressionFlag);            /* Signal receiver */
    if (is_Set (LAN_FCS_Present)) {
        FCS = TrailingOctets (PDU, 4);    /* Store FCS */
        RemoveTrailingOctets (PDU, 4);    /* Remove FCS */
        while (PacketLength > 14 &&       /* Stop at MAC header or */
               TrailingOctet (PDU) == 0)  /*  last non-zero octet */
            RemoveTrailingOctets (PDU, 1);/* Remove zero octet */
        Appendbuf (PDU, 4, FCS);          /* Restore FCS */
    }
    else {
        while (PacketLength > 14 &&       /* Stop at MAC header */
               TrailingOctet (PDU) == 0)  /*  or last zero octet */
            RemoveTrailingOctets (PDU, 1);/* Remove zero octet */
    }
}
```

PPP Receiver:

```
if (ZeroCompressionFlag) {                /* Flag set in header? */
 /* Restoring packet to minimum 802.3 length */
    Clear (ZeroCompressionFlag);
    if (is_Set (LAN_FCS_Present)) {
        FCS = TrailingOctets (PDU, 4);   /* Store FCS */
        RemoveTrailingOctets (PDU, 4);   /* Remove FCS */
        Appendbuf (PDU, 60 - PacketLength, zeroes);/* Add zeroes */
        Appendbuf (PDU, 4, FCS);         /* Restore FCS */
    }
    else {
        Appendbuf (PDU, 60 - PacketLength, zeroes);/* Add zeroes */
    }
}
```

Security Considerations

Security issues are not discussed in this memo.

References

[1] IBM, "Token-Ring Network Architecture Reference", 3rd edition, September 1989.

[2] IEEE 802.1, "Draft Standard 802.1G: Remote MAC Bridging", P802.1G/D7, December 30, 1992.

[3] IEEE 802.1, "Media Access Control (MAC) Bridges", ISO/IEC 15802-3:1993 ANSI/IEEE Std 802.1D, 1993 edition., July 1993.

[4] Reynolds, J., and J. Postel, "Assigned Numbers", STD 2, RFC 1340, USC/Information Sciences Institute, July 1992.

[5] Simpson, W., "PPP LCP Extensions", RFC 1570, Daydreamer, January 1994.

[6] Simpson, W., "The Point-to-Point Protocol (PPP)", RFC 1548, Daydreamer, December 1993.

[7] Sklower, K., "A Multilink Protocol for Synchronizing the Transmission of Multi-protocol Datagrams", Work in Progress, August 1993.

Acknowledgments

This document is a product of the Point-to-Point Protocol Extensions Working Group.

Special thanks go to Steve Senum of Network Systems, Dino Farinacci of 3COM, Rick Szmauz of Digital Equipment Corporation, and Andrew Fuqua of IBM.

Chair's Address

The working group can be contacted via the current chair:

Fred Baker
Advanced Computer Communications
315 Bollay Drive
Santa Barbara, California 93117

EMail: fbaker@acc.com

Author's Address

Questions about this memo can also be directed to:

Rich Bowen
International Business Machines Corporation
P. O. Box 12195
Research Triangle Park, NC 27709

Phone: (919) 543-9851
EMail: Rich_Bowen@vnet.ibm.com