                      The PPP Multilink Protocol (MP)

Status of This Memo

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

Abstract

   This document proposes a method for splitting, recombining and
   sequencing datagrams across multiple logical data links. This work
   was originally motivated by the desire to exploit multiple bearer
   channels in ISDN, but is equally applicable to any situation in which
   multiple PPP links connect two systems, including async links. This
   is accomplished by means of new PPP [2] options and protocols.

Acknowledgements

Table of Contents

1. Introduction

1.1. Motivation

   Basic Rate and Primary Rate ISDN both offer the possibility of
   opening multiple simultaneous channels between systems, giving users
   additional bandwidth on demand (for additional cost). Previous
   proposals for the transmission of internet protocols over ISDN have
   stated as a goal the ability to make use of this capability, (e.g.,
   Leifer et al., [1]).

   There are proposals being advanced for providing synchronization
   between multiple streams at the bit level (the BONDING proposals);
   such features are not as yet widely deployed, and may require
   additional hardware for end system. Thus, it may be useful to have a
   purely software solution, or at least an interim measure.

   There are other instances where bandwidth on demand can be exploited,
   such as using a dialup async line at 28,800 baud to back up a leased
   synchronous line, or opening additional X.25 SVCs where the window
   size is limited to two by international agreement.

   The simplest possible algorithms of alternating packets between
   channels on a space available basis (which might be called the Bank
   Teller's algorithm) may have undesirable side effects due to
   reordering of packets.

   By means of a four-byte sequencing header, and simple synchronization
   rules, one can split packets among parallel virtual circuits between
   systems in such a way that packets do not become reordered, or at
   least the likelihood of this is greatly reduced.

1.2. Functional Description

   The method discussed here is similar to the multilink protocol
   described in ISO 7776 [4], but offers the additional ability to split
   and recombine packets, thereby reducing latency, and potentially
   increase the effective maximum receive unit (MRU). Furthermore,
   there is no requirement here for acknowledged-mode operation on the
   link layer, although that is optionally permitted.

   Multilink is based on an LCP option negotiation that permits a system
   to indicate to its peer that it is capable of combining multiple
   physical links into a "bundle". Only under exceptional conditions
   would a given pair of systems require the operation of more than one
   bundle connecting them.

   Multilink is negotiated during the initial LCP option negotiation. A
   system indicates to its peer that it is willing to do multilink by
   sending the multilink option as part of the initial LCP option
   negotiation. This negotiation indicates three things:

   1.   The system offering the option is capable of combining
         multiple physical links into one logical link;

   2.   The system is capable of receiving upper layer protocol data
         units (PDU) fragmented using the multilink header (described
         later) and reassembling the fragments back into the original
         PDU for processing;

   3.   The system is capable of receiving PDUs of size N octets
         where N is specified as part of the option even if N is larger
         than the maximum receive unit (MRU) for a single physical
         link.

   Once multilink has been successfully negotiated, the sending system
   is free to send PDUs encapsulated and/or fragmented with the
   multilink header.

1.3. Conventions

   The following language conventions are used in the items of
   specification in this document:

   o   MUST, SHALL or MANDATORY -- the item is an absolute requirement
         of the specification.

   o   SHOULD or RECOMMENDED -- the item should generally be followed
         for all but exceptional circumstances.

     o   MAY or OPTIONAL -- the item is truly optional and may be
            followed or ignored according to the needs of the implementor.

2. General Overview

     In order to establish communications over a point-to-point link, each
     end of the PPP link must first send LCP packets to configure the data
     link during Link Establishment phase. After the link has been
     established, PPP provides for an Authentication phase in which the
     authentication protocols can be used to determine identifiers
     associated with each system connected by the link.

     The goal of multilink operation is to coordinate multiple independent
     links between a fixed pair of systems, providing a virtual link with
     greater bandwidth than any of the constituent members. The aggregate
     link, or bundle, is named by the pair of identifiers for two systems
     connected by the multiple links. A system identifier may include
     information provided by PPP Authentication [3] and information
     provided by LCP negotiation. The bundled links can be different
     physical links, as in multiple async lines, but may also be instances
     of multiplexed links, such as ISDN, X.25 or Frame Relay. The links
     may also be of different kinds, such as pairing dialup async links
     with leased synchronous links.

     We suggest that multilink operation can be modeled as a virtual PPP
     link-layer entity wherein packets received over different physical
     link-layer entities are identified as belonging to a separate PPP
     network protocol (the Multilink Protocol, or MP) and recombined and
     sequenced according to information present in a multilink
     fragmentation header. All packets received over links identified as
     belonging to the multilink arrangement are presented to the same
     network-layer protocol processing machine, whether they have
     multilink headers or not.

     The packets to be transmitted using the multilink procedure are
     encapsulated according to the rules for PPP where the following
     options would have been manually configured:

          o No async control character Map
          o No Magic Number
          o No Link Quality Monitoring
          o Address and Control Field Compression
          o Protocol Field Compression
          o No Compound Frames
          o No Self-Describing-Padding

Of course, individual links are permitted to have different settings
for these options. As described below, member links SHOULD negotiate
Self-Describing-Padding, even though pre-fragmented packets MUST NOT
be padded.

LCP negotiations are not permitted on the bundle itself. An
implementation MUST NOT transmit LCP Configure-Request, -Reject,
-Ack, -Nak, Terminate-Request or -Ack packets via the multilink
procedure, and an implementation receiving them MUST silently discard
them. (By "silently discard" we mean to not generate any PPP packets
in response; an implementation is free to generate a log entry
registering the reception of the unexpected packet). By contrast,
other LCP packets having control functions not associated with
changing the defaults for the bundle itself are permitted. An
implementation MAY transmit LCP Code-Reject, Protocol-Reject, Echo-
Request, Echo-Reply and Discard-Request Packets.

The effective MRU for the logical-link entity is negotiated via an
LCP option. It is irrelevant whether Network Control Protocol
packets are encapsulated in multilink headers or not, or even over
which link they are sent, once that link identifies itself as
belonging to a multilink arrangement.

Note that network protocols that are not sent using multilink headers
cannot be sequenced. (And consequently will be delivered in any
convenient way).

For example, consider the case in Figure 1. Link 1 has negotiated
network layers NL 1, NL 2, and MP between two systems. The two
systems then negotiate MP over Link 2.

Frames received on link 1 are demultiplexed at the data link layer
according the PPP network protocol identifier and can be sent to NL
1, NL 2, or MP. Link 2 will accept frames with all network protocol
identifiers that Link 1 does.

Frames received by MP are further demultiplexed at the network layer
according to the PPP network protocol identifier and sent to NL 1 or
NL 2. Any frames received by MP for any other network layer
protocols are rejected using the normal protocol reject mechanism.

Figure 1. Multilink Overview.

```
   Network Layer
   -------------
          _____              _____
         /      \            /      \
        |  NL 1  |          |  NL 2  |
         _____/            _____/
          | | |              | | |
          | | +-------------o-o-o-+
          | +------+  +-----+ | | |
          |        |  |       | | |
          | +------o--o-------+ + |
          | |      |__|_       | |
          | |     /      \     | |
          | |    |  MLCP  | <--- Link Layer
          | |     _____/     Demultiplexing
          | |        |         | |
          | |        |         | |
          | |        | <--- Virtual Link
          | |        |         | |
          | |        |         | |
          | |        |         | |
          | |        |         | |
          | |        +         | |
         _|_|       |         _|_|
        /    \      |        /    \
       |  LCP |-----+-----|  LCP  | <--- Link Layer
        \____/     |        \____/     Demultiplexing
          |        |          |
          |        |          |
       Link 1             Link 2
```
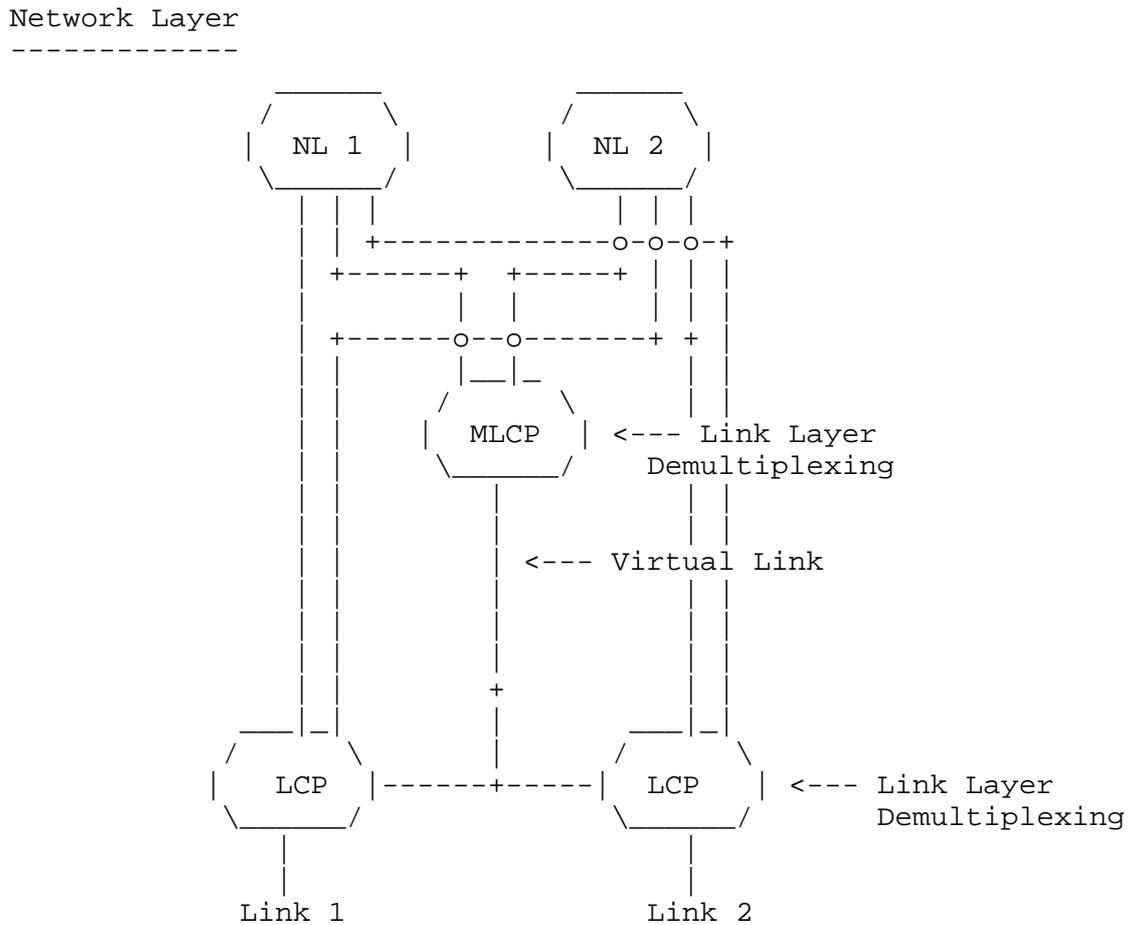
3. Packet Formats

   In this section we describe the layout of individual fragments, which
   are the "packets" in the Multilink Protocol. Network Protocol
   packets are first encapsulated (but not framed) according to normal
   PPP procedures, and large packets are broken up into multiple
   segments sized appropriately for the multiple physical links. A new
   PPP header consisting of the Multilink Protocol Identifier, and the
   Multilink header is inserted before each section. (Thus the first
   fragment of a multilink packet in PPP will have two headers, one for
   the fragment, followed by the header for the packet itself).

   Systems implementing the multilink procedure are not required to
   fragment small packets. There is also no requirement that the
   segments be of equal sizes, or that packets must be broken up at all.
   A possible strategy for contending with member links of differing
   transmission rates would be to divide the packets into segments
   proportion to the transmission rates. Another strategy might be to
   divide them into many equal fragments and distribute multiple
   fragments per link, the numbers being proportional to the relative
   speeds of the links.

   PPP multilink fragments are encapsulated using the protocol
   identifier 0x00-0x3d. Following the protocol identifier is a four
   byte header containing a sequence number, and two one bit fields
   indicating that the fragment begins a packet or terminates a packet.
   After negotiation of an additional PPP LCP option, the four byte
   header may be optionally replaced by a two byte header with only a 12
   bit sequence space. Address & Control and Protocol ID compression
   are assumed to be in effect. Individual fragments will, therefore,
   have the following format:

             Figure 2: Long Sequence Number Fragment Format.


                     +---------------+---------------+
   PPP Header: | Address 0xff | Control 0x03 |
                     +---------------+---------------+
                     | PID(H) 0x00 | PID(L) 0x3d |
                     +-+-+-+-+-+-+-+-+---------------+
   MP Header: |B|E|0|0|0|0|0|0|sequence number|
                     +-+-+-+-+-+-+-+-+---------------+
                     | sequence number (L) |
                     +---------------+---------------+
                     | fragment data |
                     | . |
                     | . |
                     | . |
                     +---------------+---------------+
   PPP FCS: | FCS |
                     +---------------+---------------+
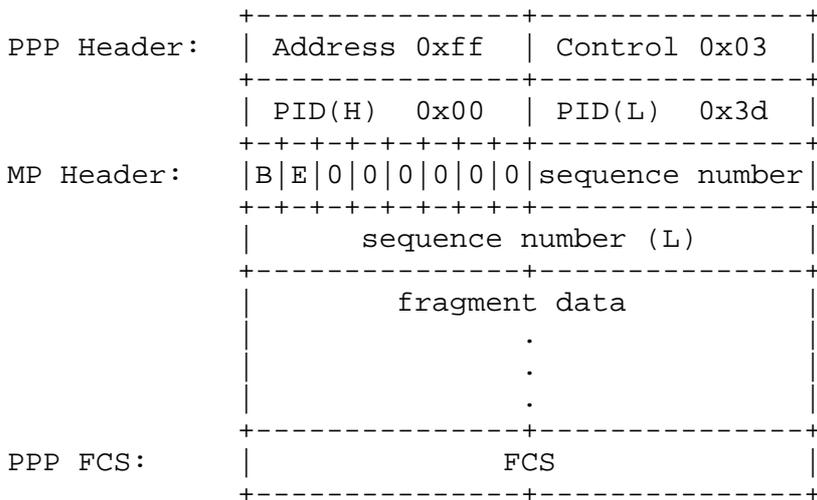
Figure 3: Short Sequence Number Fragment Format.

```
                        +---------------+---------------+
    PPP Header:         | Address 0xff  | Control 0x03  |
                        +---------------+---------------+
                        | PID(H)  0x00  | PID(L)  0x3d  |
                        +-+-+-+-+-------+---------------+
    MP Header:          |B|E|0|0|   sequence number     |
                        +-+-+-+-+-------+---------------+
                        |      fragment data            |
                        |                .              |
                        |                .              |
                        |                .              |
                        +---------------+---------------+
    PPP FCS:            |             FCS               |
                        +---------------+---------------+
```
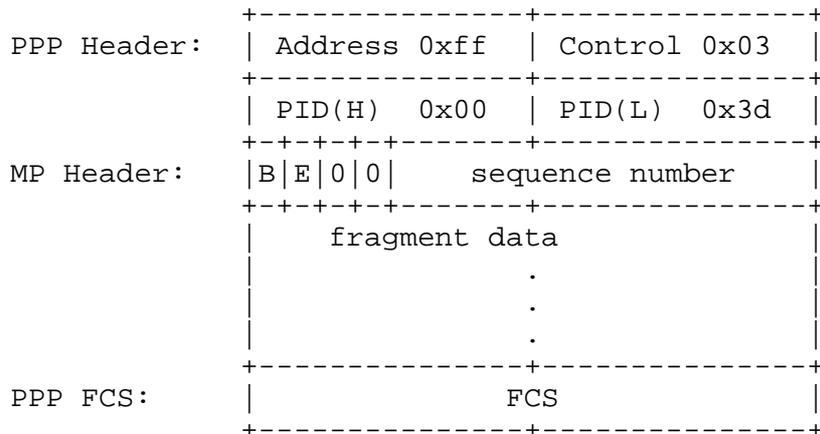
The (B)eginning fragment bit is a one bit field set to 1 on the first
fragment derived from a PPP packet and set to 0 for all other
fragments from the same PPP packet.

The (E)nding fragment bit is a one bit field set to 1 on the last
fragment and set to 0 for all other fragments. A fragment may have
both the (B)eginning and (E)nding fragment bits set to 1.

The sequence field is a 24 bit or 12 bit number that is incremented
for every fragment transmitted. By default, the sequence field is 24
bits long, but can be negotiated to be only 12 bits with an LCP
configuration option described below.

Between the (E)nding fragment bit and the sequence number is a
reserved field, whose use is not currently defined, which MUST be set
to zero. It is 2 bits long when the use of short sequence numbers
has been negotiated, 6 bits otherwise.

In this multilink protocol, a single reassembly structure is
associated with the bundle. The multilink headers are interpreted in
the context of this structure.

The FCS field shown in the diagram is inherited from the normal
framing mechanism from the member link on which the packet is
transmitted. There is no separate FCS applied to the reconstituted
packet as a whole if transmitted in more than one fragment.

3.1. Padding Considerations

   Systems that support the multilink protocol SHOULD implement Self-
   Describing-Padding. A system that implements self-describing-padding
   by definition will either include the padding option in its initial
   LCP Configure-Requests, or (to avoid the delay of a Configure-Reject)
   include the padding option after receiving a NAK containing the
   option.

   A system that must pad its own transmissions but does not use Self-
   Describing-Padding when not using multilink, MAY continue to not use
   Self-Describing-Padding if it ensures by careful choice of fragment
   lengths that only (E)nding fragments of packets are padded. A system
   MUST NOT add padding to any packet that cannot be recognized as
   padded by the peer. Non-terminal fragments MUST NOT be padded with
   trailing material by any other method than Self-Describing-Padding.

   A system MUST ensure that Self-Describing-Padding as described in RFC
   1570 [11] is negotiated on the individual link before transmitting
   any multilink data packets if it might pad non-terminal fragments or
   if it would use network or compression protocols that are vulnerable
   to padding, as described in RFC 1570. If necessary, the system that
   adds padding MUST use LCP Configure-NAK's to elicit a Configure-
   Request for Self-Describing-Padding from the peer.

   Note that LCP Configure-Requests can be sent at any time on any link,
   and that the peer will always respond with a Configure-Request of its
   own. A system that pads its transmissions but uses no protocols
   other than multilink that are vulnerable to padding MAY delay
   ensuring that the peer has Configure-Requested Self-Describing-
   Padding until it seems desireable to negotiate the use of Multilink
   itself. This permits the interoperability of a system that pads with
   older peers that support neither Multilink nor Self-Describing-
   Padding.

4. Trading Buffer Space Against Fragment Loss

   In a multilink procedure one channel may be delayed with respect to
   the other channels in the bundle. This can lead to fragments being
   received out of order, thus increasing the difficulty in detecting
   the loss of a fragment. The task of estimating the amount of space
   required for buffering on the receiver becomes more complex because
   of this. In this section we discuss a technique for declaring that a
   fragment is lost, with the intent of minimizing the buffer space
   required, yet minimizing the number of avoidable packet losses.

4.1. Detecting Fragment Loss

   On each member link in a bundle, the sender MUST transmit fragments
   with strictly increasing sequence numbers (modulo the size of the
   sequence space). This requirement supports a strategy for the
   receiver to detect lost fragments based on comparing sequence
   numbers. The sequence number is not reset upon each new PPP packet,
   and a sequence number is consumed even for those fragments which
   contain an entire PPP packet, i.e., one in which both the (B)eginning
   and (E)nding bits are set.

   An implementation MUST set the sequence number of the first fragment
   transmited on a newly-constructed bundle to zero. (Joining a
   secondary link to an exisiting bundle is invisible to the protocol,
   and an implementation MUST NOT reset the sequence number space in
   this situation).

   The receiver keeps track of the incoming sequence numbers on each
   link in a bundle and maintains the current minimum of the most
   recently received sequence number over all the member links in the
   bundle (call this M). The receiver detects the end of a packet when
   it receives a fragment bearing the (E)nding bit. Reassembly of the
   packet is complete if all sequence numbers up to that fragment have
   been received.

   A lost fragment is detected when M advances past the sequence number
   of a fragment bearing an (E)nding bit of a packet which has not been
   completely reassembled (i.e., not all the sequence numbers between
   the fragment bearing the (B)eginning bit and the fragment bearing the
   (E)nding bit have been received). This is because of the increasing
   sequence number rule over the bundle.

   An implementation MUST assume that if a fragment bears a (B)eginning
   bit, that the previously numbered fragment bore an (E)nding bit.
   Thus if a packet is lost bearing the (E)nding bit, and the packet
   whose fragment number is M contains a (B)eginning bit, the
   implementation MUST discard fragments for all unassembled packets
   through M-1, but SHOULD NOT discard the fragment bearing the new
   (B)eginning bit on this basis alone.

   The detection of a lost fragment causes the receiver to discard all
   fragments up to M. If the fragment with sequence number M has the
   (B)eginning bit set then the receiver starts reassembling the new
   packet, otherwise the receiver resynchronizes on the next fragment
   bearing the (B)eginning bit. All fragments received while the
   receiver is attempting to resynchronize not bearing the (B)eginning
   bit SHOULD be discarded.

Fragments may be lost due to corruption of individual packets or
catastrophic loss of the link (which may occur only in one
direction). This version of the multilink protocol mandates no
specific procedures for the detection of failed links. The PPP link
quality management facility, or the periodic issuance of LCP echo-
requests could be used to achieve this.

Senders SHOULD avoid keeping any member links idle to maximize early
detection of lost fragments by the receiver, since the value of M is
not incremented on idle links. Senders SHOULD rotate traffic among
the member links if there isn't sufficient traffic to overflow the
capacity of one link to avoid idle links.

Loss of the final fragment of a transmission can cause the receiver
to stall until new packets arrive. The likelihood of this may be
decreased by sending a null fragment on each member link in a bundle
that would otherwise become idle immediately after having transmitted
a fragment bearing the (E)nding bit, where a null fragment is one
consisting only of a multilink header bearing both the (B)egin and
(E)nding bits (i.e., having no payload). Implementations concerned
about either wasting bandwidth or per packet costs are not required
to send null fragments and may elect to defer sending them until a
timer expires, with the marginally increased possibility of lengthier
stalls in the receiver. The receiver SHOULD implement some type of
link idle timer to guard against indefinite stalls.

The increasing sequence per link rule prohibits the reallocation of
fragments queued up behind a failing link to a working one, a
practice which is not unusual for implementations of ISO multilink
over LAPB [4].

4.2. Buffer Space Requirements

There is no amount of buffering that will guarantee correct detection
of fragment loss, since an adversarial peer may withhold a fragment
on one channel and send arbitrary amounts on the others. For the
usual case where all channels are transmitting, you can show that
there is a minimum amount below which you could not correctly detect
packet loss. The amount depends on the relative delay between the
channels, (D[channel-i,channel-j]), the data rate of each channel,
R[c], the maximum fragment size permitted on each channel, F[c], and
the total amount of buffering the transmitter has allocated amongst
the channels.

When using PPP, the delay between channels could be estimated by
using LCP echo request and echo reply packets. (In the case of links
of different transmission rates, the round trip times should be
adjusted to take this into account.) The slippage for each channel

is defined as the bandwidth times the delay for that channel relative
to the channel with the longest delay, S[c] = R[c] * D[c,c-worst].
(S[c-worst] will be zero, of course!)

A situation which would exacerbate sequence number skew would be one
in which there is extremely bursty traffic (almost allowing all
channels to drain), and then where the transmitter would first queue
up as many consecutively numbered packets on one link as it could,
then queue up the next batch on a second link, and so on. Since
transmitters must be able to buffer at least a maximum- sized
fragment for each link (and will usually buffer up at least two) A
receiver that allocates any less than S[1] + S[2] + ... + S[N] + F[1]
+ ... + F[N], will be at risk for incorrectly assuming packet loss,
and therefore, SHOULD allocate at least twice that.

5. PPP Link Control Protocol Extensions

   If reliable multilink operation is desired, PPP Reliable Transmission
   [6] (essentially the use of ISO LAPB) MUST be negotiated prior to the
   use of the Multilink Protocol on each member link.

   Whether or not reliable delivery is employed over member links, an
   implementation MUST present a signal to the NCP's running over the
   multilink arrangement that a loss has occurred.

   Compression may be used separately on each member link, or run over
   the bundle (as a logical group link). The use of multiple
   compression streams under the bundle (i.e., on each link separately)
   is indicated by running the Compression Control Protocol [5] but with
   an alternative PPP protocol ID.

5.1. Configuration Option Types

   The Multilink Protocol introduces the use of additional LCP
   Configuration Options:

        o   Multilink Maximum Received Reconstructed Unit
        o   Multilink Short Sequence Number Header Format
        o   Endpoint Discriminator

5.1.1. Multilink MRRU LCP option

                   Figure 4: Multilink MRRU LCP option

       0   1   2   3
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
      +-+
      | Type = 17 | Length = 4 | Max-Receive-Reconstructed-Unit|
      +-+
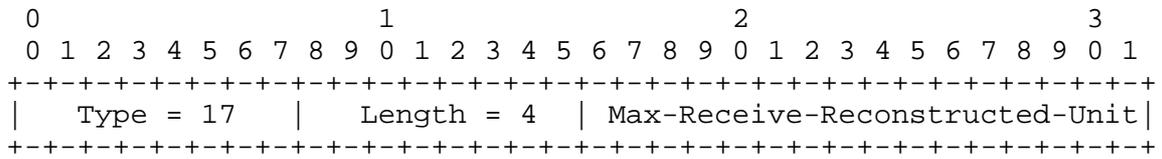
      The presence of this option indicates that the system sending it
      implements the PPP Multilink Protocol, and unless rejected, will
      construe all packets receive on this link as being able to be
      processed by a common protocol machine with any other packets
      received from the same peer on any other link on which this option
      has been accepted. A system MUST NOT accept the Multilink MRRU LCP
      Option if it is not willing to symmetrically have the packets it
      sends interpreted in the same fashion.

      This option also advises the peer that the implementation will be
      able to reconstruct a PPP packet whose payload will contain the
      number of bytes as Max-Receive-Reconstructed-Unit.

      A system MAY indicate the desire to conduct multilink operation
      solely by use of the Multilink Short Sequence Number Header Format
      LCP option (discussed next); the default value for MRRU option is
      1600 bytes if not otherwise explicitly negotiated.

      Note: this option corresponds to what would have been the MRU of the
      bundle when conceptualized as a PPP-like entity.

5.1.2. Short Sequence Number Header Format Option

              Figure 5: Short Sequence Number Header Format Option

       0   1
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      | Type = 18 | Length = 2 |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
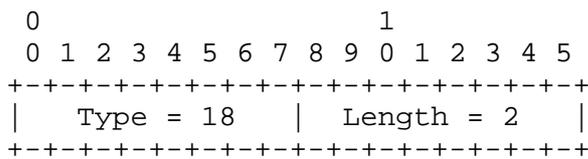
      This option advises the peer that the implementation wishes to
      receive fragments with short, 12 bit sequence numbers. By default
      sequence, numbers are 24 bits long. When this option is received, an
      implementation MUST either transmit all subsequent multilink packets
      on all links of the bundle with 12 bit sequence numbers or
      configure-NAK or configure-Reject the option.

An implementation wishing to transmit multilink fragments with short
sequence numbers MAY include the multilink short sequence number in a
configure-NAK to ask that the peer respond with a request to receive
short sequence numbers. The peer is not compelled to respond with
the option.

5.1.3. Endpoint Discriminator Option

                 Figure 7: Endpoint Discriminator Option

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type = 19   |     Length    |     Class     |  Address ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
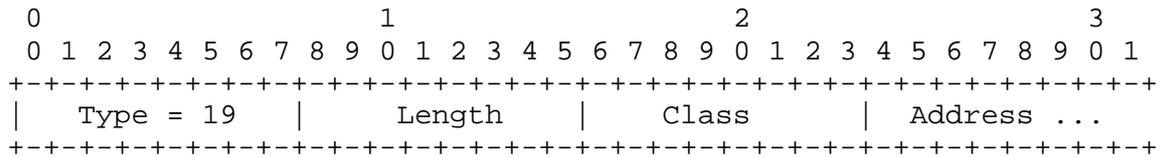
The Endpoint Discriminator Option represents identification of the
system transmitting the packet. This option advises a system that
the peer on this link could be the same as the peer on another
existing link. If the option distinguishes this peer from all
others, a new bundle MUST be established from the link being
negotiated. If this option matches the class and address of some
other peer of an existing link, the new link MUST be joined to the
bundle containing the link to the matching peer or MUST establish a
new bundle, depending on the decision tree shown in (1) through (4)
below.

To securely join an existing bundle, a PPP authentication protocol
[3] must be used to obtain authenticated information from the peer to
prevent a hostile peer from joining an existing bundle by presenting
a falsified discriminator option.

This option is not required for multilink operation. If a system
does not receive either of the Multilink MRRU or Short Sequence
options, but does receive the Endpoint Discriminator Option, and
there is no manual configuration providing outside information, the
implementation MUST NOT assume that multilink operation is being
requested on this basis alone.

As there is also no requirement for authentication, there are four
sets of scenarios:

(1) No authentication, no discriminator:
     All new links MUST be joined to one bundle.

(2) Discriminator, no authentication:
     Discriminator match -> MUST join matching bundle,
     discriminator mismatch -> MUST establish new bundle.

   (3)   No discriminator, authentication:
         Authenticated match -> MUST join matching bundle,
         authenticated mismatch -> MUST establish new bundle.

   (4)   Discriminator, authentication:
         Discriminator match and authenticated match -> MUST join bundle,
         discriminator mismatch -> MUST establish new bundle,
         authenticated mismatch -> MUST establish new bundle.

The option contains a Class which selects an identifier address space
and an Address which selects a unique identifier within the class
address space.

This identifier is expected to refer to the mechanical equipment
associated with the transmitting system. For some classes,
uniqueness of the identifier is global and is not bounded by the
scope of a particular administrative domain. Within each class,
uniqueness of address values is controlled by a class dependent
policy for assigning values.

Each endpoint may chose an identifier class without restriction.
Since the objective is to detect mismatches between endpoints
erroneously assumed to be alike, mismatch on class alone is
sufficient. Although no one class is recommended, classes which have
universally unique values are preferred.

This option is not required to be supported either by the system or
the peer. If the option is not present in a Configure-Request, the
system MUST NOT generate a Configure-Nak of this option, instead it
SHOULD behave as if it had received the option with Class = 0,
Address = 0. If a system receives a Configure-Nak or Configure-
Reject of this option, it MUST remove it from any additional
Configure-Request.

The size is determined from the Length field of the element. For
some classes, the length is fixed, for others the length is variable.
The option is invalid if the Length field indicates a size below the
minimum for the class.

An implementation MAY use the Endpoint Discriminator to locate
administration or authentication records in a local database. Such
use of this option is incidental to its purpose and is deprecated
when a PPP Authentication protocol [3] can be used instead. Since
some classes permit the peer to generate random or locally assigned
address values, use of this option as a database key requires prior
agreement between peer administrators.

The specification of the subfields are:

Type
     19 = for Endpoint Discriminator

Length
     3 + length of Address

Class
     The Class field is one octet and indicates the identifier
     address space. The most up-to-date values of the LCP Endpoint
     Discriminator Class field are specified in the most recent
     "Assigned Numbers" RFC [7]. Current values are assigned as
     follows:

     0   Null Class

     1   Locally Assigned Address

     2   Internet Protocol (IP) Address

     3   IEEE 802.1 Globally Assigned MAC Address

     4   PPP Magic-Number Block

     5   Public Switched Network Directory Number

Address
     The Address field is one or more octets and indicates the
     identifier address within the selected class. The length and
     content depend on the value of the Class as follows:

     Class 0 - Null Class

          Maximum Length: 0

          Content:
          This class is the default value if the option is not
          present in a received Configure-Request.

Class 1 - Locally Assigned Address

    Maximum Length: 20

    Content:

    This class is defined to permit a local assignment in the
    case where use of one of the globally unique classes is not
    possible. Use of a device serial number is suggested. The
    use of this class is deprecated since uniqueness is not
    guaranteed.

Class 2 - Internet Protocol (IP) Address

    Fixed Length: 4

    Content:

    An address in this class contains an IP host address as
    defined in [8].

Class 3 - IEEE 802.1 Globally Assigned MAC Address

    Fixed Length: 6

    Content:

    An address in this class contains an IEEE 802.1 MAC address
    in canonical (802.3) format [9]. The address MUST have the
    global/local assignment bit clear and MUST have the
    multicast/specific bit clear. Locally assigned MAC
    addresses should be represented using Class 1.

Class 4 - PPP Magic-Number Block

    Maximum Length: 20

    Content:

    This is not an address but a block of 1 to 5 concatenated
    32 bit PPP Magic-Numbers as defined in [2]. This class
    provides for automatic generation of a value likely but not
    guaranteed to be unique. The same block MUST be used by an
    endpoint continuously during any period in which at least
    one link is in the LCP Open state. The use of this class
    is deprecated.

Note that PPP Magic-Numbers are used in [2] to detect
unexpected loopbacks of a link from an endpoint to itself.
There is a small probability that two distinct endpoints
will generate matching magic-numbers. This probability is
geometrically reduced when the LCP negotiation is repeated
in search of the desired mismatch, if a peer can generate
uncorrelated magic-numbers.

As used here, magic-numbers are used to determine if two
links are in fact from the same peer endpoint or from two
distinct endpoints. The numbers always match when there is
one endpoint. There is a small probability that the
numbers will match even if there are two endpoints. To
achieve the same confidence that there is not a false match
as for LCP loopback detection, several uncorrelated magic-
numbers can be combined in one block.

Class 5 - Public Switched Network Directory Number

Maximum Length: 15

Content:

An address in this class contains an octet sequence as
defined by I.331 (E.164) representing an international
telephone directory number suitable for use to access the
endpoint via the public switched telephone network [10].

6. Closing Member links

   Member links may be terminated according to normal PPP LCP procedures
   using LCP terminate-request and terminate-ack packets on that member
   link. Since it is assumed that member links usually do not reorder
   packets, receipt of a terminate ack is sufficient to assume that any
   multilink protocol packets ahead of it are at no special risk of
   loss.

   Receipt of an LCP terminate-request on one link does not conclude the
   procedure on the remaining links.

   So long as any member links in the bundle are active, the PPP state
   for the bundle persists as a separate entity.

   If the multilink procedure is used in conjunction with PPP reliable
   transmission, and a member link is not closed gracefully, the
   implementation should expect to receive packets which violate the
   increasing sequence number rule.

7. Interaction with Other Protocols

   In the common case, LCP, and the Authentication Control Protocol
   would be negotiated over each member link. The Network Protocols
   themselves and associated control exchanges would normally have been
   conducted once, on the bundle.

   In some instances it may be desirable for some Network Protocols to
   be exempted from sequencing requirements, and if the MRU sizes of the
   link did not cause fragmentation, those protocols could be sent
   directly over the member links.

   Although explicitly discouraged above, if there were several member
   links connecting two implementations, and independent sequencing of
   two protocol sets were desired, but blocking of one by the other was
   not, one could describe two multilink procedures by assigning
   multiple endpoint identifiers to a given system. Each member link,
   however, would only belong to one bundle. One could think of a
   physical router as housing two logically separate implementations,
   each of which is independently configured.

   A simpler solution would be to have one link refuse to join the
   bundle, by sending a Configure-Reject in response to the Multilink
   LCP option.

8. Security Considerations

   Operation of this protocol is no more and no less secure than
   operation of the PPP authentication protocols [3]. The reader is
   directed there for further discussion.

9. References

   [1] Leifer, D., Sheldon, S., and B. Gorsline "A Subnetwork Control
       Protocol for ISDN Circuit-Switching", University of Michigan
       (unpublished), March 1991.

   [2] Simpson, W., Editor, "The Point-to-Point Protocol (PPP)", STD 51,
       RFC 1661, Daydreamer, July 1994.

   [3] Lloyd, B., and W. Simpson, "PPP Authentication Protocols", RFC
       1334, Lloyd Internetworking, Daydreamer, October 1992.

   [4] International Organisation for Standardization, "HDLC -
       Description of the X.25 LAPB-Compatible DTE Data Link
       Procedures", International Standard 7776, 1988

   [5] Rand, D., "The PPP Compression Control Protocol (CCP)", PPP
       Extensions Working Group, Work in Progress.

   [6] Rand, D., "PPP Reliable Transmission", PPP Extensions Working
       Group, Work in Progress.

   [7] Reynolds, J., and J. Postel, "Assigned Numbers", STD 2, RFC 1700,
       USC/Information Sciences Institute, October 1994.

   [8] Postel, J., Editor, "Internet Protocol - DARPA Internet Program
       Protocol Specification", STD 5, RFC 791, USC/Information Sciences
       Institute, September 1981.

   [9] Institute of Electrical and Electronics Engineers, Inc., "IEEE
       Local and Metropolitan Area Networks: Overview and Architecture",
       IEEE Std. 802-1990, 1990.

  [10] The International Telegraph and Telephone Consultative Committee
       (CCITT), "Numbering Plan for the ISDN Area", Recommendation I.331
       (E.164), 1988.

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

10. Authors' Addresses

Keith Sklower
Computer Science Department
384 Soda Hall, Mail Stop 1776
University of California
Berkeley, CA 94720-1776

Phone: (510) 642-9587
EMail: sklower@CS.Berkeley.EDU

Brian Lloyd
Lloyd Internetworking
3031 Alhambra Drive
Cameron Park, CA 95682

Phone: (916) 676-1147
EMail: brian@lloyd.com

Glenn McGregor
Lloyd Internetworking
3031 Alhambra Drive
Cameron Park, CA 95682

Phone: (916) 676-1147
EMail: glenn@lloyd.com

Dave Carr
Newbridge Networks Corporation
600 March Road
P.O. Box 13600
Kanata, Ontario,
Canada, K2K 2E6

Phone: (613) 591-3600
EMail: dcarr@Newbridge.COM