                   The PPP Compression Control Protocol (CCP)

Status of this Memo

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

Abstract

   The Point-to-Point Protocol (PPP) [1] provides a standard method for
   transporting multi-protocol datagrams over point-to-point links. PPP
   also defines an extensible Link Control Protocol.

   This document defines a method for negotiating data compression over
   PPP links.

Table of Contents

1. Introduction

   In order to establish communications over a PPP link, each end of the
   link must first send LCP packets to configure and test the data link
   during Link Establishment phase. After the link has been
   established, optional facilities may be negotiated as needed.

One such facility is data compression. A wide variety of compression
methods may be negotiated, although typically only one method is used
in each direction of the link.

A different compression algorithm may be negotiated in each
direction, for speed, cost, memory or other considerations, or only
one direction may be compressed.

2. Compression Control Protocol (CCP)

The Compression Control Protocol (CCP) is responsible for
configuring, enabling, and disabling data compression algorithms on
both ends of the point-to-point link. It is also used to signal a
failure of the compression/decompression mechanism in a reliable
manner.

CCP uses the same packet exchange mechanism as the Link Control
Protocol (LCP). CCP packets may not be exchanged until PPP has
reached the Network-Layer Protocol phase. CCP packets received
before this phase is reached should be silently discarded.

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

Frame Modifications

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

Data Link Layer Protocol Field

   Exactly one CCP packet is encapsulated in the PPP Information
   field, where the PPP Protocol field indicates type hex 80FD
   (Compression Control Protocol).

   When individual link data compression is used in a multiple link
   connection to a single destination, the PPP Protocol field
   indicates type hex 80FB (Individual link Compression Control
   Protocol).

Code field

   In addition to Codes 1 through 7 (Configure-Request, Configure-
   Ack, Configure-Nak, Configure-Reject, Terminate-Request,
   Terminate-Ack and Code-Reject), two additional Codes 14 and 15
   (Reset-Request and Reset-Ack) are defined for this protocol.
   Other Codes should be treated as unrecognized and should result in
   Code-Rejects.

Timeouts

CCP 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

CCP has a distinct set of Configuration Options.

2.1. Sending Compressed Datagrams

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

One or more compressed packets are encapsulated in the PPP
Information field, where the PPP Protocol field indicates type hex
00FD (Compressed datagram). Each of the compression algorithms may
use a different mechanism to indicate the inclusion of more than one
uncompressed packet in a single Data Link Layer frame.

When using multiple PPP links to a single destination, there are two
methods of employing data compression. The first method is to
compress the data prior to sending it out through the multiple links.
The second is to treat each link as a separate connection, that may
or may not have compression enabled. In the second case, the PPP
Protocol field MUST be type hex 00FB (Individual link compressed
datagram).

Only one primary algorithm in each direction is in use at a time, and
that is negotiated prior to sending the first compressed frame. The
PPP Protocol field of the compressed datagram indicates that the
frame is compressed, but not the algorithm with which it was
compressed.

The maximum length of a compressed packet transmitted over a PPP link
is the same as the maximum length of the Information field of a PPP
encapsulated packet. Larger datagrams (presumably the result of the
compression algorithm increasing the size of the message in some
cases) may be sent uncompressed, using its standard form, or may be
sent in multiple datagrams, if the compression algorithm supports it.

Each of the compression algorithms must supply a way of determining
if they are passing data reliably, or they must require the use of a

reliable transport such as LAPB [3]. Vendors are strongly encouraged
to employ a method of validating the compressed data, or recognizing
out-of-sync compressor/decompressor pairs.

3. Additional Packets

   The Packet format and basic facilities are already defined for LCP
   [1].

   Up-to-date values of the CCP Code field are specified in the most
   recent "Assigned Numbers" RFC [2]. This specification concerns the
   following values:

      14   Reset-Request
      15   Reset-Ack

3.1. Reset-Request and Reset-Ack

   Description

      CCP includes Reset-Request and Reset-Ack Codes in order to provide
      a mechanism for indicating a decompression failure in one
      direction of a compressed link without affecting traffic in the
      other direction. A decompression failure may be determined by
      periodically passing a hash value, performing a CRC check on the
      decompressed data, or other mechanism. It is strongly suggested
      that some mechanism be available in all compression algorithms to
      validate the decompressed data before passing the data on to the
      rest of the system.
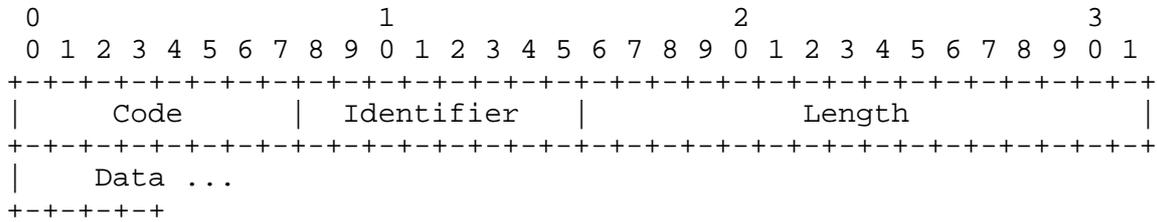
      A CCP implementation wishing to indicate a decompression failure
      SHOULD transmit a CCP packet with the Code field set to 14
      (Reset-Request), and the Data field filled with any desired data.
      Once a Reset-Request has been sent, any Compressed packets
      received are discarded, and another Reset-Request is sent with the
      same Identifier, until a valid Reset-Ack is received.

      Upon reception of a Reset-Request, the transmitting compressor is
      reset to an initial state. This may include clearing a
      dictionary, resetting hash codes, or other mechanisms. A CCP
      packet MUST be transmitted with the Code field set to 15 (Reset-
      Ack), the Identifier field copied from the Reset-Request packet,
      and the Data field filled with any desired data.

      On receipt of a Reset-Ack, the receiving decompressor is reset to
      an initial state. This may include clearing a dictionary,
      resetting hash codes, or other mechanisms. Since there may be
      several Reset-Acks in the pipe, the decompressor MUST be reset for

each Reset-Ack which matches the currently expected identifier.

A summary of the Reset-Request and Reset-Ack packet formats is shown
below. The fields are transmitted from left to right.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Data ...
+-+-+-+-+
```

Code

    14 for Reset-Request;

    15 for Reset-Ack.

Identifier

    On transmission, the Identifier field MUST be changed whenever the
    content of the Data field changes, and whenever a valid reply has
    been received for a previous request. For retransmissions, the
    Identifier MAY remain unchanged.

    On reception, the Identifier field of the Reset-Request is copied
    into the Identifier field of the Reset-Ack packet.

Data

    The Data field is zero or more octets and contains uninterpreted
    data for use by the sender. The data may consist of any binary
    value and may be of any length from zero to the peer's established
    MRU minus four.

4. CCP Configuration Options

   CCP Configuration Options allow negotiation of compression algorithms
   and their parameters. CCP uses the same Configuration Option format
   defined for LCP [1], with a separate set of Options.

   Configuration Options, in this protocol, indicate algorithms that the
   receiver is willing or able to use to decompress data sent by the
   sender. As a result, it is to be expected that systems will offer to
   accept several algorithms, and negotiate a single one that will be
   used.

There is the possibility of not being able to agree on a compression
algorithm. In that case, no compression will be used, and the link
will continue to operate without compression. If link reliability
has been separately negotiated, then it will continue to be used,
until the LCP is re-negotiated.

We expect that many vendors will want to use proprietary compression
algorithms, and have made a mechanism available to negotiate these
without encumbering the Internet Assigned Number Authority with
proprietary number requests.

The LCP option negotiation techniques are used. If an option is
unrecognized, a Configure-Reject MUST be sent. If all protocols the
sender implements are Configure-Rejected by the receiver, then no
compression is enabled in that direction of the link.

If an option is recognized, but not acceptable due to values in the
request (or optional parameters not in the request), a Configure-NAK
MUST be sent with the option modified appropriately. The Configure-
NAK MUST contain only those options that will be acceptable. A new
Configure-Request SHOULD be sent with only the single preferred
option, adjusted as specified in the Configure-Nak.

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

    CCP Option   Compression type
    0   OUI
    1   Predictor type 1
    2   Predictor type 2
    3   Puddle Jumper
    4-15   unassigned
    16   Hewlett-Packard PPC
    17   Stac Electronics LZS
    18   Microsoft PPC
    19   Gandalf FZA
    20   V.42bis compression
    21   BSD LZW Compress
    255   Reserved

The unassigned values 4-15 are intended to be assigned to other
freely available compression algorithms that have no license fees.
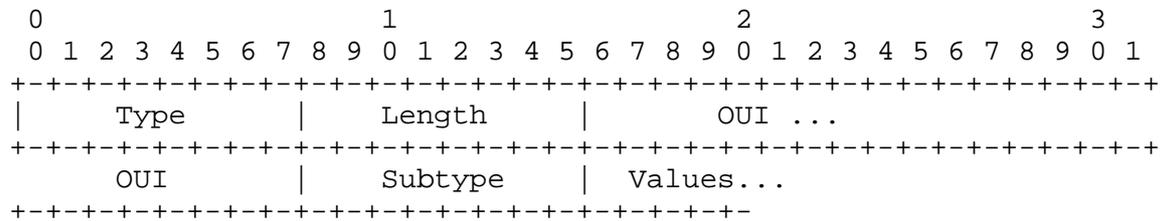
4.1. Proprietary Compression OUI

   Description

      This Configuration Option provides a way to negotiate the use of a
      proprietary compression protocol.

      Since the first matching compression will be used, it is
      recommended that any known OUI compression options be transmitted
      first, before the common options are used.

      Before accepting this option, the implementation must verify that
      the Organization Unique Identifier identifies a proprietary
      algorithm that the implementation can decompress, and that any
      vendor specific negotiation values are fully understood.

   A summary of the Proprietary Compression OUI Configuration Option
   format is shown below. The fields are transmitted from left to
   right.

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |          OUI ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     OUI        |    Subtype    |  Values...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

   Type

      0

   Length

      >= 6

   IEEE OUI

      The vendor's IEEE Organization Unique Identifier (OUI), which is
      the most significant three octets of an Ethernet Physical Address,
      assigned to the vendor by IEEE 802. This identifies the option as
      being proprietary to the indicated vendor. The bits within the
      octet are in canonical order, and the most significant octet is
      transmitted first.

   Subtype

      This field is specific to each OUI, and indicates a compression
      type for that OUI. There is no standardization for this field.
      Each OUI implements its own values.

   Values

      This field is zero or more octets, and contains additional data as
      determined by the vendor's compression protocol.
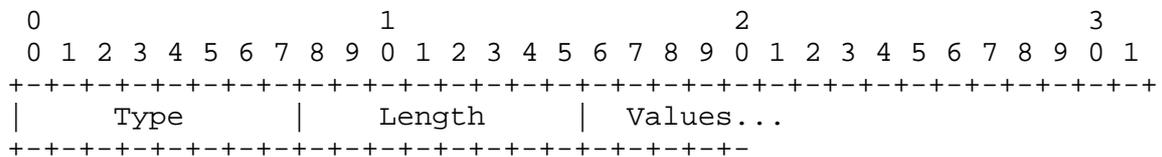
4.2. Other Compression Types

   Description

      These Configuration Options provide a way to negotiate the use of
      a publicly defined compression algorithm. Many compression
      algorithms are specified. No particular compression technique has
      arisen as an Internet Standard.

      These protocols will be made available to all interested parties,
      but may have certain licensing restrictions associated with them.
      For additional information, refer to the compression protocol
      documents that define each of the compression types.

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

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

   Type

      1 to 254

   Length

      >= 2

   Values

      This field is zero or more octets, and contains additional data as
      determined by the compression protocol.

Security Considerations

    Security issues are not discussed in this memo.

References

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

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

    [3]   Rand, D., "PPP Reliable Transmission", RFC 1663, July 1994.

Acknowledgments

    Bill Simpson helped with the document formatting.

Chair's Address

    The working group can be contacted via the current chair:

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

        EMail: karl@ascend.com

Author's Address

    Questions about this memo can also be directed to:

        Dave Rand
        Novell, Inc.
        2180 Fortune Drive
        San Jose, CA 95131

        +1 408 321-1259

        EMail: dlr@daver.bungi.com