PPP Over Asynchronous Transfer Mode Adaptation Layer 2 (AAL2)

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) provides a standard method for
   transporting multi-protocol datagrams over point-to-point links.

   This document describes the use of ATM Adaptation Layer 2 (AAL2) for
   framing PPP encapsulated packets.

Applicability

   This specification is intended for those implementations which desire
   to use the facilities which are defined for PPP, such as the Link
   Control Protocol, Network-layer Control Protocols, authentication,
   and compression. These capabilities require a point-to-point
   relationship between the peers, and are not designed for the multi-
   point relationships which are available in ATM and other multi-access
   environments.

1. Introduction

   PPP over AAL5 [2] describes the encapsulation format and operation of
   PPP when used with the ATM AAL5 adaptation layer. While this
   encapsulation format is well suited to PPP transport of IP, it is
   bandwidth inefficient when used for transporting small payloads such
   as voice. PPP over AAL5 is especially bandwidth inefficient when
   used with RTP header compression [3].

   PPP over AAL2 addresses the bandwidth efficiency issues of PPP over
   AAL5 for small packet transport by making use of the AAL2 Common Part
   Sublayer (CPS) [4] to allow multiple PPP payloads to be multiplexed
   into a set of ATM cells.

2. Conventions

   The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD,
   SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this
   document, are to be interpreted as described in [6].

3. AAL2 Layer Service Interface

   The PPP layer treats the underlying ATM AAL2 layer service as a bit-
   synchronous point-to-point link. In this context, the PPP link
   corresponds to an ATM AAL2 virtual connection. The virtual
   connection MUST be full-duplex, point to point, and it MAY be either
   dedicated (i.e., permanent, set up by provisioning) or switched (set
   up on demand). In addition, the PPP/AAL2 service interface boundary
   MUST meet the following requirements.

      Interface Format - The PPP/AAL2 layer boundary presents an octet
      service interface to the AAL2 layer. There is no provision for
      sub-octets to be supplied or accepted.

      Transmission Rate - The PPP layer does not impose any
      restrictions regarding transmission rate on the underlying ATM
      layer traffic descriptor parameters.

      Control Signals - The AAL2 layer MUST provide control signals to
      the PPP layer which indicate when the virtual connection link has
      become connected or disconnected. These provide the "Up" and
      "Down" events to the LCP state machine [1] within the PPP layer.

      In the case of PPP over AAL2, the state of the link can be derived
      from the type 3 fault management packets carried in-band within a
      given AAL2 CID flow.

4. PPP Operation with AAL2

   PPP over AAL2 defines an encapsulation that uses the Service Specific
   Segmentation and Reassembly Sublayer (SSSAR) [5] for AAL type 2. The
   SSSAR sub-layer is used to segment PPP packets into frames that can
   be transported using the AAL2 CPS. The SSSAR sub-layer uses
   different AAL2 UUI code-points to indicate whether a segment is the
   last segment of a packet or not.

   The encapsulation of PPP over AAL2 provides a 16-bit CRC for PPP
   payloads. There are 2 UUI code points assigned from SSSAR to
   indicate intermediate fragments of a packet and the last fragment of
   a packet. Code point 27 indicates intermediate frames of a
   fragmented packet and code point 26 indicates the last frame of a
   packet. The encapsulation format is more fully described in section
   6.2.1.

   An implementation of PPP over AAL2 MAY use one or more AAL2 Channel
   Identifiers (CIDs) for transport of PPP packets associated with each
   PPP session. Multiple CIDs could be used to implement a multiple
   class real time transport service for PPP using the AAL2 layer for
   link fragmentation and interleaving. A companion document [10]
   describes class extensions for PPP over AAL2 using multiple AAL2
   CIDs.

5. Comparison of PPP Over AAL2 with Existing Encapsulations

   This document proposes the substitution of AAL2 transport for PPP in
   scenarios where small packets are being transported over an ATM
   network. This is most critical in applications such as voice
   transport using RTP [9] where RTP Header compression [3] is used. In
   applications such as voice transport, both bandwidth efficiency and
   low delay are very important.

   This section provides justification for the PPP over AAL2 service for
   ATM transport by comparing it to existing PPP encapsulation formats
   used for transport over ATM. Two encapsulation formats will be
   examined here: PPP over AAL5 [2], and PPP with PPPMUX [8] over AAL5.

5.1 Comparison With PPP Over AAL5

   This proposal uses ATM AAL2 [4] rather than AAL5 as the transport for
   PPP. SSSAR along with the AAL2 CPS generates less ATM encapsulation
   overhead per PPP payload. The payload encapsulation consists of a 2
   byte CRC. The AAL2 CPS header consists of 3 bytes, and the AAL2
   Start Field (STF) is 1 byte. This is a total encapsulation overhead
   of 6 bytes. This compares to 8 bytes of overhead for the AAL5
   trailer used for PPP over AAL5.

The multiplexing function of the AAL2 CPS layer allows more bandwidth
efficient transport of PPP frames by multiplexing multiple PPP frames
into one or more ATM cells using the AAL2 CPS function. This removes
the pad overhead of AAL5 when used to transport short frames.

5.2 Comparison with PPPMUX over AAL5

PPP Multiplexing (PPPMUX) [8] is a new method for doing multiplexing
in the PPP layer. PPPMUX provides functionality similar to the CPS
based multiplexing function of AAL2. Using PPP multiplexing, a PPP
stack would look like PPP/PPPMUX/AAL5.

Both PPP/PPPMUX/AAL5 and PPP/AAL2 use multiplexing to reduce the
overhead of cell padding when frames are sent over an ATM virtual
circuit. However, the bandwidth utilization of PPP/AAL2 will
typically be better than the bandwidth used by PPP/PPPMUX/AAL5. This
is because multiplexed frames in PPP/PPPMUX/AAL5 must always be
encapsulated within an AAL5 frame before being sent. This
encapsulation causes an additional 8 bytes of AAL5 trailer to be
added to the PPPMUX encapsulation. In addition to the 8 bytes of
AAL5 trailer, PPPMUX will incur an average of 24 additional bytes of
AAL5 PAD. These 2 factors will end up reducing the effective
efficiency of PPPMUX when it is used over AAL5.

With PPP/AAL2, the AAL2 CPS layer treats individual PPP frames as a
series of CPS payloads that can be multiplexed. As long as PPP
frames arrive at the CPS layer before the CPS TIMER_CU expires, all
ATM cells coming from the CPS layer will be filled. Under these
conditions, PPP/AAL2 will have no PAD associated with it. When the
AAL2 CPS function causes no PAD to be generated, PPP/AAL2 will be
more bandwidth efficient than PPP/PPPMUX/AAL5.

In PPP/PPPMUX/AAL5, the AAL5 SAR and the PPP MUX/DEMUX are performed
in two different layers. Thus, the PPPMUX/AAL5 receiver must
reassemble a full AAL5 frame from the ATM layer before the PPPMUX
layer can extract the PPP payloads. To derive maximum PPP
Multiplexing efficiency, many PPP payloads may be multiplexed
together. This increases the size of the multiplexed frame to many
ATM cells. If one of these ATM cells is lost, the whole PPPMUX
packet will be discarded. Also, there may be a significant delay
incurred while the AAL5 layer waits for many ATM cell arrival times
until a full frame has been assembled before the full frame is passed
up to the PPP Multiplexing layer where the inverse PPP demux then
occurs. This same issue also applies to PPPMUX/AAL5 frames
progressing down the stack.

With AAL2, both the segmentation and reassembly and multiplexing
functions are performed in the AAL2 CPS layer. Because of the
definition of the AAL2 CPS function, a multiplexed payload will be
extracted as soon as it is received. The CPS receiver does not wait
until the many payloads of an AAL2 multiplexed frame are received
before removing payloads from the multiplexed stream. The same
benefit also applies to AAL2 CPS sender implementations. Also, the
loss of an ATM cell causes the loss of the packets that are included
in that cell only.

The AAL2 CPS function provides multiplexing in AAL2. This function
often needs to be implemented in hardware for performance reasons.
Because of this, a PPP/AAL2 implementation that takes advantage of an
AAL2 SAR implemented in hardware will have significant performance
benefits over a PPP/PPPMUX/AAL5 implementation where PPPMUX is
implemented in software. Also, the AAL2 specification has been
available significantly longer than the PPP Multiplexing
specification and because of this, optimized software and hardware
implementations of the AAL2 CPS function are further in development
than PPP Multiplexing implementations.

6. Detailed Protocol Operation Description

6.1 Background

6.1.1 AAL2 Multiplexing

   ITU-T I.363.2 specifies ATM Adaptation Layer Type 2. This AAL type
   provides for bandwidth efficient transmission of low-rate, short and
   variable length packets in delay sensitive applications. More than
   one AAL type 2 user information stream can be supported on a single
   ATM connection. There is only one definition for the sub-layer
   because it implements the interface to the ATM layer and is shared by
   more than one SSCS layer.

6.1.2 AAL2 Service Specific Convergence Sub-layers

   ITU-T I.366.1 and I.366.2 define Service Specific Convergence Sub-
   layers (SSCS) that operate above the Common Part Sub-layer defined in
   I.363.2. This layer specifies packet formats and procedures to
   encode the different information streams in bandwidth efficient
   transport. As the name implies, this sub-layer implements those
   elements of service specific transport. While there is only one
   definition of the Common Part Sublayer for AAL2, there can be
   multiple SSCS functions defined to run over this CPS layer.
   Different CIDs within an AAL2 virtual circuit may run different
   SSCSs.

6.1.3 AAL2 CPS Packet (CPS-PKT) Format

   The CPS-PKT format over AAL2 as defined in I.363.2:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         +         +         +         +                       |
|   CID   +   LI    +   UUI   +   HEC   +        CPS-INFO        |
|         +         +         +         +                       |
|         +         +         +         +                       |
|   (8)   +   (6)   +   (5)   +   (5)   +       (45/64 * 8)      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
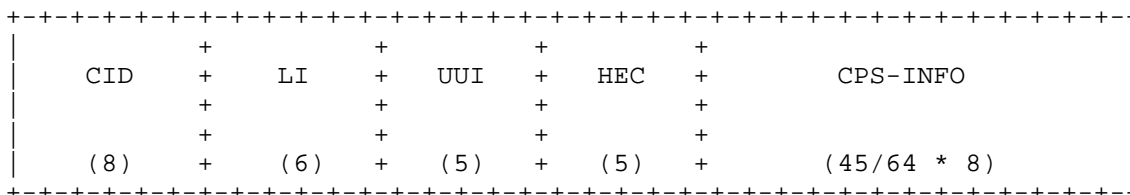
                 Note: The size of the fields denote bit-width

   The Channel ID (CID) identifies the sub-stream within the AAL2
   connection. The Length indication (LI) indicates the length of the
   CPS-INFO payload. The User-to-User Indication (UUI) carries
   information between the SSCS/Application running above the CPS. The
   SSSAR sub-layer as defined in I.366.1 uses the following code points:

      UUI Code-point   Packet Content
      +++++++++++++   +++++++++++++

      0-26   Framed mode data, final packet.

      27   Framed mode data, more to come.

   This proposal uses two UUI code-points as follows:

      UUI Code-point   Packet Content
      +++++++++++++   +++++++++++++

      27   non-final packet.

      26   final packet.

6.1.4 AAL2 CPS-PDU Format

   The CPS-PDU format over AAL2 as defined in I.363.2:

```
                        +-+-+-+~+~+-+-+
                        |CPS| CPS-INFO|
                        |PKT|         |
                        |HDR|         |
                        +-+-+-+~+~+-+-+
                        |  CPS-PKT    |

                        |               +-+-+-+~+~+-+-+
                        |               |CPS| CPS-INFO|
                        |               |PKT|         |
                        |               |HDR|         |
                        |               +-+-+-+~+~+-+-+
                                           CPS-PKT
                        |               |   +-+-+-+~+~+-+-+
                        |               |   |CPS| CPS-INFO|
                        |               |   |PKT|         |
                        |               |   |HDR|         |
                        |               |   +-+-+-+~+~+-+-+
                                               CPS-PKT
                        V               V       V           V
 +-+-+-+-+-+-+-+~+~+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   Cell   |         |                                   |     |   |
   Header |   STF   |       CPS-PDU Payload             | PAD |
          |         |                                   |     |   |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+~+~+
```
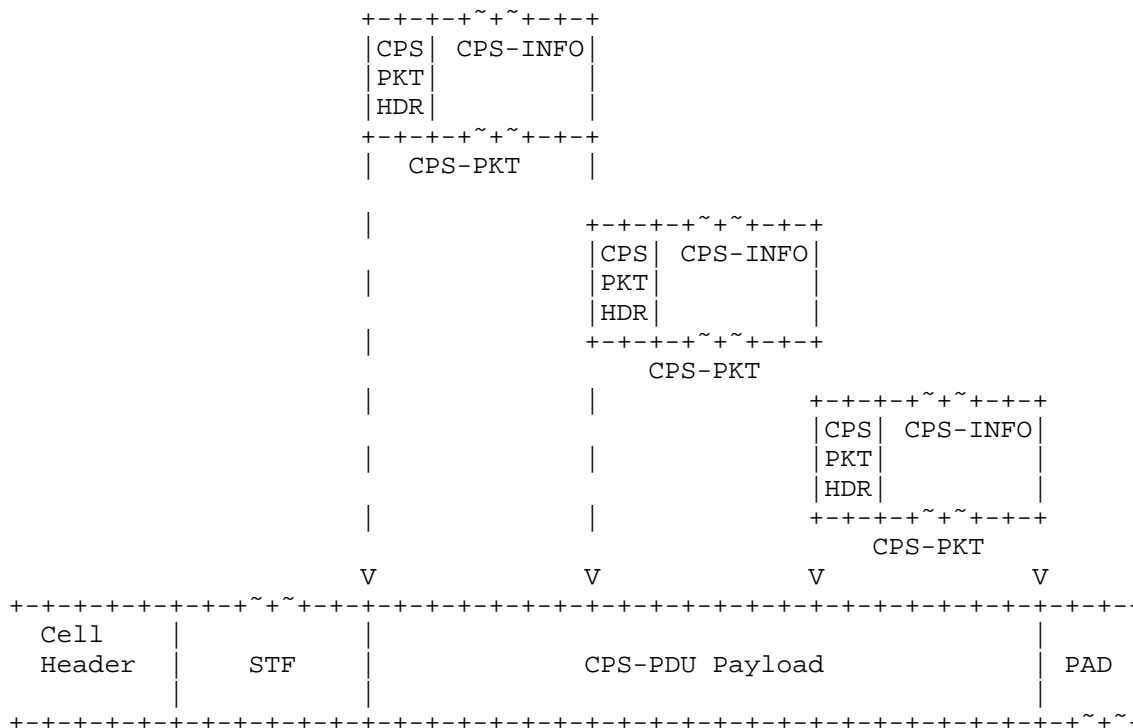
            Note: The size of the fields denote bitwidth

   The CPS-PDU format is used to carry one or more CPS-PKT's multiplexed
   on a single CPS-PDU. The CPS header contains enough information to
   identify the CPS packets within a CPS-PDU. In the event of cell loss,
   the STF field is used to find the first CPS-PKT in the current cell.

6.2 PPP Over AAL2 Encapsulation

    PPP encapsulation over AAL2 uses the AAL2 CPS with no change.

    Some PPP encapsulated protocols such as RTP header compression
    require that the link layer provide packet error detection. Because
    of this, PPP over AAL2 defines a 16-bit CRC that is used along with
    the SSSAR sub-layer of I.366.1 to provide packet error detection.
    The encapsulation format is described below.

6.2.1 PPP Payload Encapsulation Over AAL2 with 16-bit CRC (CRC-16).

    The payload encapsulation of PPP appends a two byte CRC to each PPP
    frame before using the SSSAR layer to send the PPP packet as a series
    of AAL2 frames.

    The format of a PPP over AAL2 packet is shown in the diagram below.
    Note that the diagram below shows the payload encapsulation when the
    packet is not segmented (UUI=26). When the PPP packet is segmented,
    the PPP Protocol ID, Information field, and CRC-16 fields will be
    split across multiple SSSAR frames. In this case, the UUI field will
    be set to 27 for all frames except the last frame. In the last frame,
    the UUI field will be set to 26.

Payload Encapsulation
```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         +       +       +       +          +          +       |
|   CID   +  LI   + UUI   + HEC   + Protocol +          +       |
|         +       +       +       +   ID     + Information + CRC-16  |
|         +       +       +       +          +          +       |
|   (8)   +  (6)  +  (5)  +  (5)  +  (8/16)  +          +  (16) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

            Note: The size of the fields denote bit-width

    The algorithms used for computing and verifying the CRC-16 field are
    identical to the algorithms specified for the Frame Check Sequence
    (FCS) field in Q.921 [13]. The algorithms from Q.921 are included in
    this section for ease of access.

    The CRC-16 field is filled with the value of a CRC calculation which
    is performed over the contents of the PPP packet, including the PPP
    Protocol ID and the information field. The CRC field shall contain
    the ones complement of the sum (modulo 2) of:

    1) the remainder of $x^k (x^{15} + x^{14} + \ldots + x + 1)$ divided (modulo
       2) by the generator polynomial, where k is the number of bits of
       the information over which the CRC is calculated; and

2) the remainder of the division (modulo 2) by the generator
   polynomial of the product of x^16 by the information over which
   the CRC is calculated.

The CRC-16 generator polynomial is:

   $G(x) = x^{16} + x^{12} + x^5 + 1$

The result of the CRC calculation is placed with the least
significant bit right justified in the CRC field.

As a typical implementation at the transmitter, the initial content
of the register of the device computing the remainder of the division
is preset to all "1"s and is then modified by division by the
generator polynomial (as described above) on the information over
which the CRC is to be calculated; the ones complement of the
resulting remainder is put into the CRC field.

As a typical implementation at the receiver, the initial content of
the register of the device computing the remainder of the division is
preset to all "1"s. The final remainder, after multiplication by
x^16 and then division (modulo 2) by the generator polynomial of the
serial incoming PPP packet (including the Protocol ID, the
information and the CRC fields), will be 0001110100001111 (x^15
through x^0, respectively) in the absence of transmission errors.

6.3 Use of AAL2 CPS-PKT CIDs

An implementation of PPP over AAL2 MAY use a single AAL2 Channel
Identifier (CID) or multiple CIDs for transport of all PPP packets.
In order for the endpoints of a PPP session to work with AAL2, they
MUST both agree on the number, SSCS mapping, and values of AAL2 CIDs
that will be used for a PPP session. The values of AAL2 CIDs to be
used for a PPP session MAY be obtained from either static
provisioning in the case of a dedicated AAL2 connection (PVC) or from
Q.2630.2 [7] signaling in the case of an AAL2 switched virtual
circuit (SPVC or SVC).

Using this proposal it is possible to support the use of conventional
AAL2 in CIDs that are not used to support PPP over AAL2. This
proposal allows the co-existence of multiple types of SSCS function
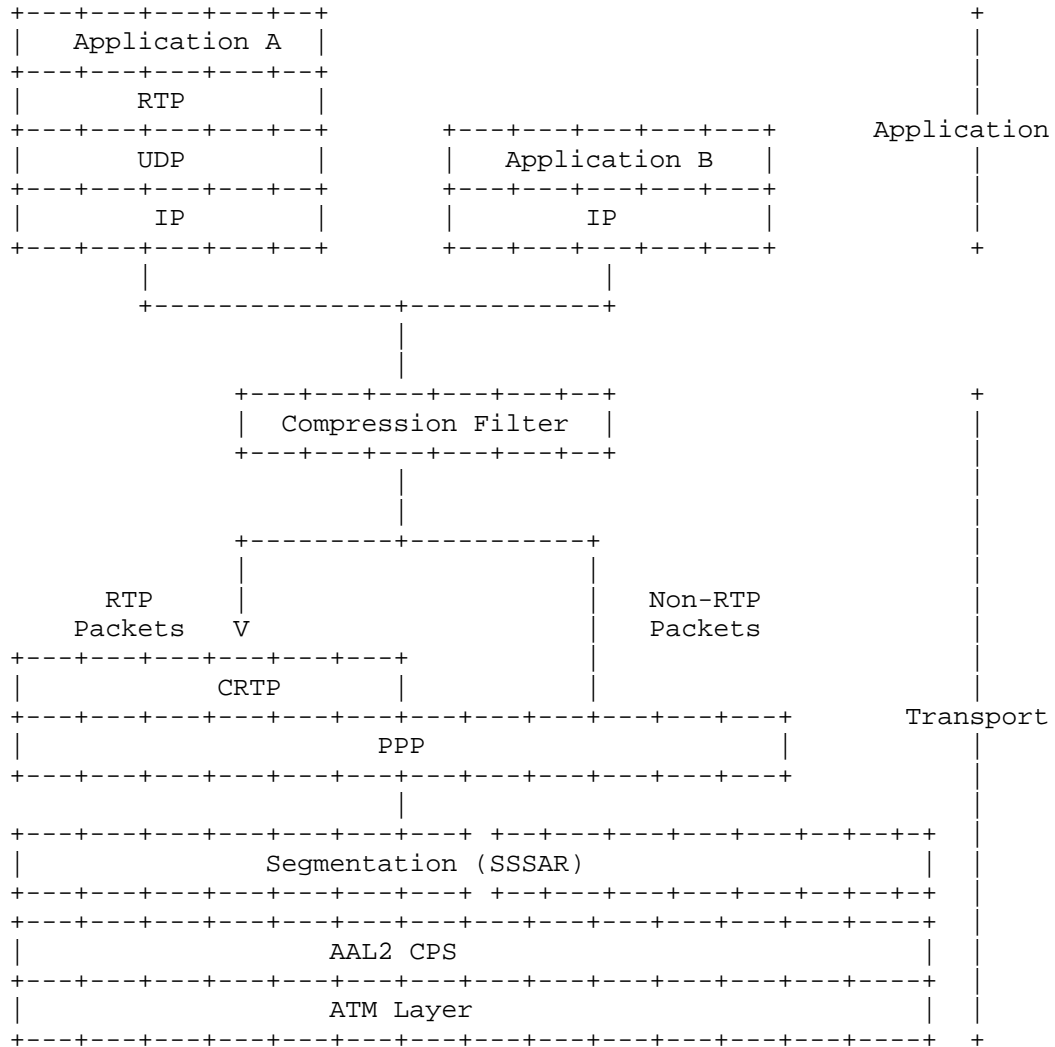within the same AAL2 VCC.

6.4 PPP over AAL2 Operation

   PPP operation with AAL2 will perform basic PPP encapsulation with the
   PPP protocol ID. A 16-bit CRC is calculated as described above and
   appended to the payload. The SSSAR sub-layer of AAL2 is used for
   transport.

   Applications implementing PPP over AAL2 MUST meet all the
   requirements of PPP [1].

7. Example implementation of PPP/AAL2

   This section describes an example implementation of how PPP can be
   encapsulated over AAL2. The example shows two application stacks
   generating IP packets that are sent to the same interface running
   PPP/AAL2. One Application stack is generating RTP packets and
   another application is generating IP Datagrams. The PPP/AAL2
   interface shown in this example is running an RFC 2508 compliant
   version of RTP header compression.

Here are the paths an Application packet can take in this
implementation:

```
    +---+---+---+---+--+                                    +
    |   Application A  |                                    |
    +---+---+---+---+--+                                    |
    |      RTP         |                                    |
    +---+---+---+---+--+        +---+---+---+---+---+    Application
    |      UDP         |        |   Application B   |        |
    +---+---+---+---+--+        +---+---+---+---+---+        |
    |      IP          |        |        IP         |        |
    +---+---+---+---+--+        +---+---+---+---+---+        +
            |                            |
            +--------------+------------+
                           |
                           |
                +---+---+---+---+---+--+                     +
                |  Compression Filter  |                     |
                +---+---+---+---+---+--+                     |
                           |                                 |
                           |                                 |
                +---------+----------+                       |
                |                    |                       |
        RTP     |                    |  Non-RTP              |
        Packets    V                 |  Packets              |
    +---+---+---+---+---+---+         |                       |
    |        CRTP         |          |                       |
    +---+---+---+---+---+---+---+---+---+---+---+---+---+  Transport
    |                     PPP                         |      |
    +---+---+---+---+---+---+---+---+---+---+---+---+---+     |
                           |                                 |
    +---+---+---+---+---+---+---+ +--+---+---+---+--+--+--+-+ |
    |          Segmentation (SSSAR)                     | |
    +---+---+---+---+---+---+---+ +--+---+---+---+--+--+--+-+ |
    +---+---+---+---+---+---+---+---+---+---+---+---+---+----+ |
    |                     AAL2 CPS                      | |
    +---+---+---+---+---+---+---+---+---+---+---+---+---+----+ |
    |                     ATM Layer                     | |
    +---+---+---+---+---+---+---+---+---+---+---+---+---+----+ +
```

In the picture above, application A is an RTP application generating
RTP packets. Application B is an IP application generating IP
datagrams. Application A gathers the RTP data and formats an RTP
packet. Lower level layers of application A add UDP and IP headers
to form a complete IP packet. Application B is generating datagrams
to the IP layer. These datagrams may not have UDP or RTP headers.

In the above picture, a protocol stack is configured to apply
CRTP/PPP/AAL2 compression on an interface to a destination host. All
packets that are sent to this interface will be tested to see if they
can be compressed using RTP header compression. As packets appear at
the interface, they will be tested by a compression filter to
determine if they are candidates for header compression. If the
compression filter determines that the packet is a candidate for
compression, the packet will be sent to the CRTP compressor. If the
packet is not a candidate for compression, it will be sent directly
to the PPP layer for encapsulation as an IP packet encapsulated in
PPP.

The destination UDP port number and packet length are examples of
criteria that may be used by the compression filter to select the
interface.

In this example, packets from application A will be passed to the
CRTP compressor which then hands the compressed packet to the PPP
layer for encapsulation as one of the compressed header types of
CRTP. The PPP layer will add the appropriate CRTP payload type for
the compressed packet.

Packets from application B will be sent directly to the PPP layer for
encapsulation as an IP/PPP packet. The PPP layer will add the PPP
payload type for an IP packet encapsulated in PPP.

PPP packets are then segmented using I.366.1 segmentation with SSSAR.

The resulting AAL2 frame mode PDU is passed down as a CPS SDU to the
CPS Layer for multiplexing accompanied by the CPS-UUI and the CPS-
CID. The CPS Layer multiplexes the CPS-PKT onto a CPS-PDU. CPS-PDUs
are passed to the ATM layer as ATM SDUs to be carried end-to-end
across the ATM network.

At the receiving end, the ATM SDU's arrive and are passed up to the
AAL2 CPS. As the AAL2 CPS PDU is accumulated, complete CPS-PKT's are
reassembled by the SSSAR SSCS. Reassembled packets are checked for
errors using the CRC algorithm.

At this point, the PPP layer on the receiving side uses the PPP
payload type to deliver the packet to either the CRTP decompressor or
the IP layer depending on the value of the PPP payload type.

8. LCP Configuration Options

   By default, PPP over AAL2 will use the 16 bit CRC encapsulation for
   all packets.

   The default Maximum-Receive-Unit (MRU) is 1500 bytes.

9. Security Considerations

   This memo defines mechanisms for PPP encapsulation over ATM. There
   is an element of trust in any encapsulation protocol: a receiver
   should be able to trust that the sender has correctly identified the
   protocol being encapsulated and that the sender has not been spoofed
   or compromised. A receiver should also be able to trust that the
   transport network between sender and receiver has not been
   compromised.

   A PPP session that runs over an ATM Virtual Circuit must follow the
   PPP link operation state machine described in RFC 1661 [1]. This
   state machine includes the ability to enforce the use of an
   authentication phase using the PAP/CHAP authentication protocols
   before any network layer packets are exchanged. Using PPP level
   authentication, a PPP receiver can authenticate a PPP sender.

   System security may also be compromised by the attacks of the ATM
   transport network itself. The ATM Forum has published a security
   framework [11] and a security specification [12] that define
   procedures to guard against common threats to an ATM transport
   network.

   PPP level authentication does not guard against man in the middle
   attacks. These attacks could occur if an attacker was able to
   compromise the security infrastructure of an ATM switching network.
   Applications that require protection against threats to an ATM
   switching network are encouraged to use authentication headers, or
   encrypted payloads, and/or the ATM-layer security services described
   in [12].

   When PPP over AAL2 is used on a set of CIDs in a virtual connection,
   there may be other non PPP encapsulated AAL2 CIDs running on the same
   virtual connection. Because of this, an end point cannot assume that
   the PPP session authentication and related security mechanisms also
   secure the non PPP encapsulated CIDs on that same virtual connection.

10. Acknowledgements

   The authors would like to thank Rajesh Kumar, Mike Mclaughlin, Pietro
   Schicker, James Carlson and John O'Neil for their contributions to
   this proposal.

11. References

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

   [2]   Gross, G., Kaycee, M., Li, A., Malis, A. and J. Stephens, "PPP
         over AAL5", STD 51, RFC 2364, July 1998.

   [3]   Casner, S. and V. Jacobson, "Compressing IP/UDP/RTP Headers for
         Low-Speed Serial Links", RFC 2508, February 1999.

   [4]   International Telecommunications Union, "BISDN ATM Adaptation
         layer specification: Type 2 AAL(AAL2)", ITU-T Recommendation
         I.363.2, September 1997.

   [5]   International Telecommunications Union, "Segmentation and
         Reassembly Service Specific Convergence Sublayer for the AAL
         type 2", ITU-T Recommendation I.366.1, June 1998.

   [6]   Bradner, S., "Key words for use in RFCs to Indicate Requirement
         Levels", BCP 14, RFC 2119, March 1997.

   [7]   ITU-T, "ITU-T RECOMMENDATION Q.2630.2", December 2000.

   [8]   Pazhyannur, R, Ali, I. and C. Fox, "PPP Multiplexing", RFC 3153,
         August 2001.

   [9]   Schulzrinne, H., Casner, S., Frederick, R. and V. Jacobson,
         "RTP: A Transport Protocol for Real-Time Applications", RFC
         1889, January 1996.

   [10] Thompson, B., Koren, T. and B. Buffam, "Class Extensions for PPP
         over Asynchronous Transfer Mode Adaptation Layer 2", RFC 3337,
         December 2002.

   [11] The ATM Forum, "ATM Security Framework Version 1.0", af-sec-
         0096.000, February 1998.

   [12] The ATM Forum, "ATM Security Specification v1.1", af-sec-
         0100.002, March 2001.

   [13] International Telecommunications Union, ISDN User-Network
        Interface-Data Link Layer Specification, ITU-T Recommendation
        Q.921, March 1993.

12. Authors' Addresses

   Bruce Thompson
   Cisco Systems, Inc.
   170 West Tasman Drive
   San Jose, CA 95134
   USA

   Phone: +1 408 527-0446
   EMail: brucet@cisco.com

   Tmima Koren
   Cisco Systems, Inc.
   170 West Tasman Drive
   San Jose, CA 95134
   USA

   Phone: +1 408 527-6169
   EMail: tmima@cisco.com

   Bruce Buffam
   Seaway Networks
   One Chrysalis Way,
   Suite 300,
   Ottawa, Canada
   K2G-6P9

   Phone: +1 613 723-9161
   EMail: bruce@seawaynetworks.com

13. Full Copyright Statement

     Copyright (C) The Internet Society (2002). All Rights Reserved.

     This document and translations of it may be copied and furnished to
     others, and derivative works that comment on or otherwise explain it
     or assist in its implementation may be prepared, copied, published
     and distributed, in whole or in part, without restriction of any
     kind, provided that the above copyright notice and this paragraph are
     included on all such copies and derivative works. However, this
     document itself may not be modified in any way, such as by removing
     the copyright notice or references to the Internet Society or other
     Internet organizations, except as needed for the purpose of
     developing Internet standards in which case the procedures for
     copyrights defined in the Internet Standards process must be
     followed, or as required to translate it into languages other than
     English.

     The limited permissions granted above are perpetual and will not be
     revoked by the Internet Society or its successors or assigns.

     This document and the information contained herein is provided on an
     "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING
     TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
     BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION
     HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF
     MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

     Funding for the RFC Editor function is currently provided by the
     Internet Society.