Encapsulation Overhead(s) in ADSL Access Networks

Author: Dirk Van Aken, Sascha Peckelbeen

Date: June 2003

Edition: v1.0

Abstract: The focus of this white paper is to derive the efficiency of protocol stacks and encapsulations embedded in ADSL Bridge/Routers. First of all a quick review is given of the various header and trailer structures involved. Secondly the line efficiency is calculated for various protocol combinations that are in use today. Finally some attention is given to effects like IP fragmentation.

Updates: Due to the continuous evolution of DSL technology, existing products are regularly improved. For more information on the latest technological innovations, software updates, and documents, please visit the SpeedTouch™ web site at:

http://www.speedtouch.com

1 INTRODUCTION

Internet Access via traditional voiceband modems is based on a rather simple protocol stack. Basically only three layers are involved: IP, PPP (or SLIP) and Voiceband modem technology for the physical layer.

In sharp contrast, ADSL Bridge/Routers combine various protocol layers, traditionally found in LAN and WAN environments, for interfacing and to realize the frame/packet forwarding function. Although each layer is required to perform services that cannot be offered by the other constituting layers, they all impose overhead to the “net payload”.

Overhead calculations are not a trivial task although it should be as it boils down to calculating the N-PDU /(N-1)PDU ratio. This because of the multitude of protocol layers involved, each of which containing protocol options described in several documents issued by different standardization bodies.

Depending on the combination of specific layers and their particular properties unexpected side effects can occur. One of these side effects is already explained in [1]. i.e. The possibility to multiplex multiple users into a single Virtual Channel.

IP fragmentation is another example of such a side effect and is explained via overhead calculations.

2 TOTAL ENCAPSULATION OVERHEAD

In calculating the overhead of the access network, the segment between the CPE ADSL modem and the ISP's Broadband Remote Access Server is considered. The overhead of the ADSL link and the SDH/SONET transport network are not taken into account because they are considered as "link local" that is, they have no end-to-end significance.

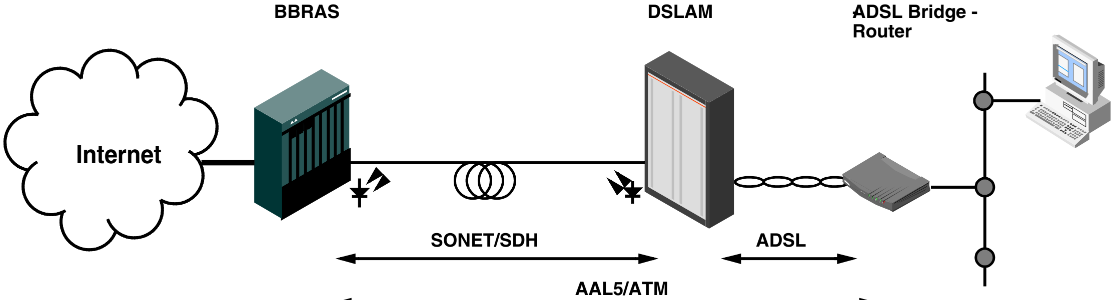


Figure 1 *Typical ADSL Overall Architecture*

The IP layer will be the "reference" layer in this paper in the sense that overheads will be calculated relative to IP. Although the added value of IP is in providing network wide connectivity, it is still at east two layers below the application layer. e.g. Most Internet applications run on top of TCP or UDP.

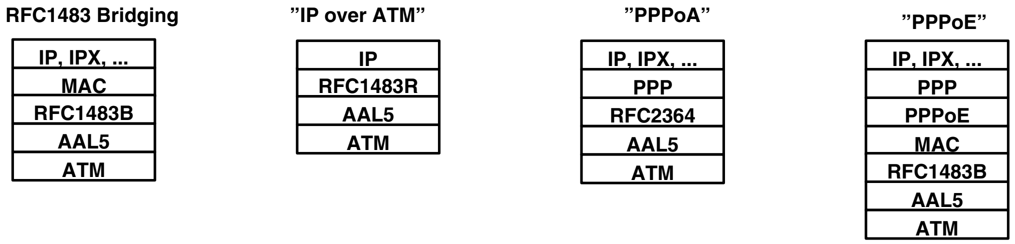
3 OVERHEAD INDIVIDUAL LAYERS

A few popular ADSL protocol stacks are illustrated in Figure 2. All of these stacks are currently in use by ADSL Bridge-Routers.

More specific:

- **RFC1483 Bridging:**
  was embedded from the very beginning in THOMSON's stand-alone DSL network terminators. It continues to survive because of its simplicity, multiprotocol nature and "zero-config" features
- **IP over ATM:**
  is applied either for RFC1483 Routing or for Classical IP [2]
- **PPPoA:**
  these layers were introduced as RFC1483 Bridging lacked features like session delineation, user authentication and link encryption to name a few
- **PPPoE:**
  is a rather recent protocol stack and allowed to reuse the installed base of ADSL Bridges in an PPP based environment.

The list is far from complete as individual layers can be combined resulting in specific configurations e.g. MAC Encapsulated Routing.



RFC1483 Bridging	"IP over ATM"	"PPPoA"	"PPPoE"
IP, IPX, ...	IP	IP, IPX, ...	IP, IPX, ...
MAC	RFC1483R	PPP	PPP
RFC1483B	AAL5	RFC2364	PPPoE
AAL5	ATM	AAL5	MAC
ATM		ATM	RFC1483B
			AAL5
			ATM

Figure 2 *Popular "ADSL-protocol" stacks.*

For a "walk" over the protocol layers, the PPPoE [3] protocol stack/solution was selected because it is the most complicated of the series. This does not compromise the general applicability of this paper though.

3.1 IP Layer

IP has a variable packet size which can range from 20 (typical IP header) to 65 535 octets. In most cases though the underlying layers greatly reduce the maximum IP packet size, impacting the overall transport efficiency.

The capabilities of the underlying layers are communicated towards the IP layer via the concept of MTU: Maximum Transfer Unit. The MTU value of an IP interface is either a fixed value or can be negotiated (Path MTU Discovery or PPP-MRU negotiation).

3.2 PPP Layer

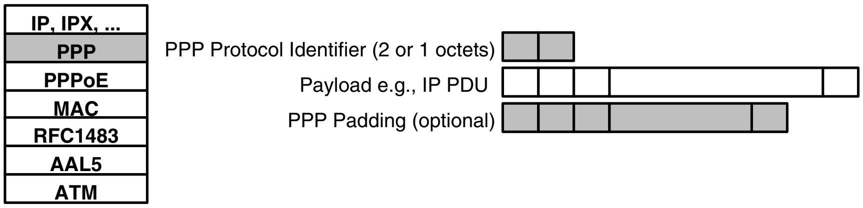


Figure 3 *PPP frame layout.*

Total PPP overhead: 2 octets.

In the generic PPP frame format [4] the possibility exists to include PPP padding octets. However most if not all applications using PPP never include padding octets. Therefore the PPP overhead is contained in the PPP protocol identifier, PPPid.

Via the Protocol Field Compression LCP Option, the PPP Protocol Id can be negotiated to a single octet. However [5] recommends not to use this option

3.3 PPPoE Layer

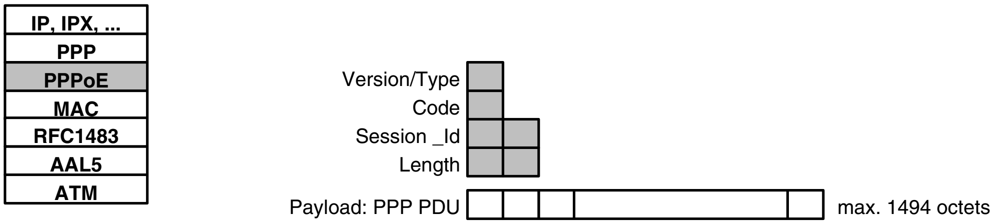


Figure 4 *PPPoE frame layout.*

Total PPPoE overhead: 6 octets.

PPPoEHeader= Version/Type + Code + PPPoE SessionId + PPPoE Length.

Note PPPoE is not a “standalone layer”, it is always accompanied with a PPP layer and MAC layer.

3.4 MAC Layer

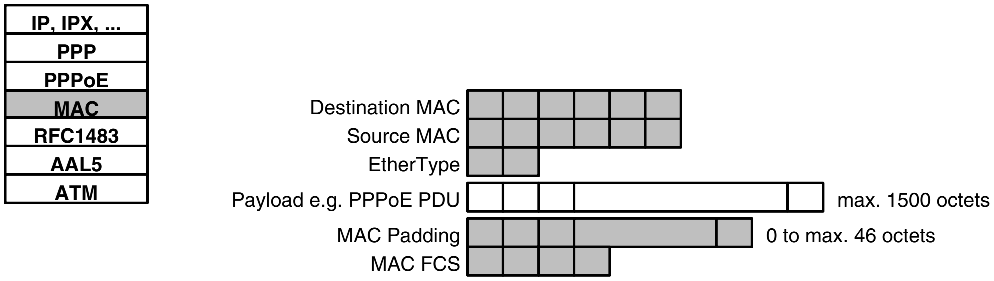


Figure 5 *Ethernet V2.0 / IEEE 802.3 frame layout.*

Total MAC overhead: 18 octets (provided that the payload is large enough to meet the minimum CSMA/CD requirements).

If the payload is less than 46 octets, then the MAC or upper layer adds padding octets to fulfil the minimum length requirement of Ethernet, see [6], [7]. i.e. An Ethernet frame must contain at least 64 octets (payload + overhead). As a result the 18 octet MAC overhead must be increased with an amount of MAC padding octets. This amount is easily calculated via the formula:

$$\#\text{PaddingOctets(PDU)} = \quad \text{Max } [(0, \text{minMACFrSize} - (\text{PDU} + \text{DstMAC} + \text{SrcMAC} + \text{Encap} + \text{FCS})]$$

$$\#\text{PaddingOctets(PDU)} = \quad \text{Max } [(0, 64 - (\text{PDU} + 6 + 6 + 2 + 4)]$$

Applied to a 40 octet TCP/IP Ack, this yields 6 octets of padding.

Note [8] Extends the traditional MAC frame format. In this paper the standard IEEE 802.3 format is used though.

In case MAC frames are transported via AAL5, then RFC1483/2684 options can be applied to reduce the MAC overhead; more specific FCS preservation and Tinygram Compression.

FCS Preservation

For MAC frames to be encapsulated in AAL5, RFC1483/2684 [9] provides the option to drop the MAC FCS field. This field contains a 32 bit CRC value calculated over the complete frame. AAL5 on itself exhibits a field containing a 32 bit CRC value calculated over the whole AAL5-CPCS frame. This renders the MAC FCS field redundant in most situations.

If the MAC FCS is not preserved, then the minimum MAC overhead is reduced to 14 octets.

Tinygram Compression

Another option that can be enabled in combination with the “No_FCS_Preservation” option of RFC1483/RFC2684 is Tinygram Compression.

Tinygram compression is described in [10] and refers to the principle of discarding/restoring MAC layer padding octets that are present in “tiny” MAC frames. i.e. Frames that are to short to fulfil the CSMA/CD requirements. Doing so further reduces the MAC overhead to the absolute minimum.

The difficulty with Tinygram compression is that a layer 3 unaware transmitter can apply it to IEEE 802.3 frames only, as there is no payload length indicator in Ethernet V2.0 frames. To simplify implementations, this form of compression can be applied in combination with "No FCS preservation" as RFC2684 specifies. This because padding octets are part of the CRC-32 calculation.

Note From the very beginning THOMSON ADSL Bridge/Routers supported Tinygram compression. In this way these devices are not only compliant with RFC1483 but also with its successor RFC2684.

3.5 RFC1483/2684 - RFC2364 Layer

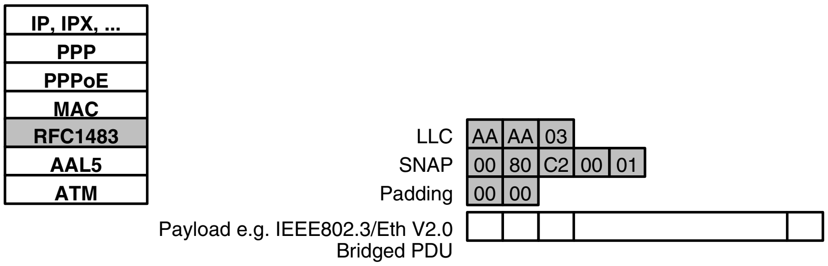


Figure 6 *RFC1483/2684 frame layout.*

Maximum RFC1483/RFC2684 overhead: 10 octets (Bridged PDU's, FCS preserved).

RFC1483/RFC2684 exhibits two encapsulation methods and its encapsulation headers vary according to the type of PDU to be transported. The table below gives an overview of the overhead for the most popular PDU's.

Encapsulated Protocol	LLC Encapsulation	VC Based Multiplexing
Bridged IEEE 802.3 PDU's FCS not preserved	6	-2[1]
Bridged IEEE 802.3 PDU's FCS preserved	10	2
Routed non-ISO PDU's (e.g. IP)	8	0
Routed ISO PDU's	4	0
PPP PDU's [2]	4	0
LLC PDU's (e.g. XID and TEST)	0	n.a.

Table 1 *FC1483/RFC2684 overhead.*

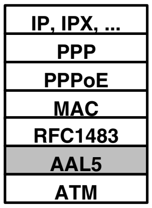
The de-facto default encapsulation for MAC frames, called Bridged PDU's in RFC1483/RFC2684, is the LLC/SNAP method MAC FCS not preserved. This encapsulation method imposes 10 octets overhead to the MAC frame (actually 10 - 4 (FCS)= 6).

---

[1] 2 RFC1483 Bridging padding octets - 4 MAC FCS octets

[2] This encapsulation is identical to Routed ISO PDU's. Although PPP over ATM encapsulation could have been integrated into RFC1483/2684 very easily, still it became a separate RFC i.e. RFC2364

3.6 AAL5 Layer



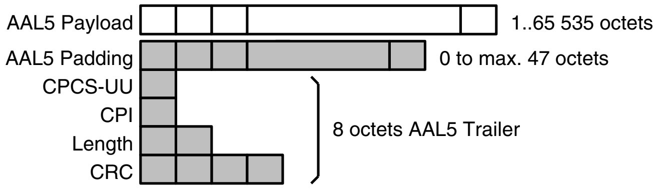


Figure 7 *AAL5 frame layout.*

Minimum AAL5 overhead: 8 octets (VC-Mux encapsulation, no AAL5 padding)
Maximum AAL5 overhead: 47 + 8= 55 octets.

AAL5 [11] is a trailer protocol i.e. it adds a trailer to the end of the encapsulated payload. Prior to transfer AAL5 frames to the ATM layer, they are segmented in 48 octet chunks. In order for the AAL5 frame to fit in an integral number of ATM cells, the AAL5 layer add padding octets until a multiple of 48 octets is reached. Via the formula below the number of AAL5 padding octets can be obtained.

$$\#AAL5Padding(Encap,PDU) = roundup[\frac{(Encap+PDU+AAL5Trailer)}{48}, 0] * 48$$

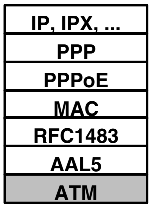
$$- (Encap + PDU + AAL5Trailer)$$

To calculate the AAL5 PDU size given a PDU and its encapsulation, the formula below was constructed. The arguments LLcVCMux, SNAPon, NLPIDon, BriOn and FCSpres are set to either 0 or 1 depending on encapsulation option. e.g. To calculate the AAL5 PDU size for Bridged IEEE 802.3/Ethernet frames that are LLC/SNAP encapsulated FCS not preserved set LLcVCMux= 1, SNAPon= 1, NLPIDon= 0, BriOn=1 and FCSpres= 0.

$$AAL5PDU(PDU,LLcVCMux,SNAPon,NLPIDon,BriOnFC,Spres) = \begin{aligned}&PDU + \\ &LLcVCMux * (LLCHeader + \\ &SNAPon * SNAPheader + \\ &NLPIDon * NLPIDheader) + \\ &BriOn * (BriPadding - FCSpres * FCS) + \\ &AAL5Trialer + \\ &AAL5Padding[LLcVCMux * (LLCHeader + \\ &SNAPon * SNAPheader + \\ &NLPIDon * NLPIDheader) + \\ &BriOn * (BriPadding - FCSpres * FCS)]\end{aligned}$$

Examples of the overhead induced by the selected encapsulation option and the AAL5 padding function are given in Table 2.

3.7 ATM Layer



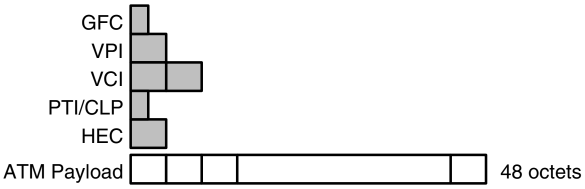


Figure 8 *ATM frame layout at the UNI*

ATM overhead: 5 octets.

The size of an ATM cell [12] and its accompanying header are fixed; therefore the ATM efficiency is easily calculated i.e. 48/53*100= 90.6 % or 9.4% of the cell is used for the ATM header.

In contrast to all previous sections, there is no 1-to-1 mapping of PDU's in ATM cells. As a consequence the overhead increases stepwise as a function of the AAL5 payload size. Following simple formulas allow to derive the total ATM layer octets for a given AAL5 PDU.

$$\#C\_ATMLayerOctets(AAL5PDU) = \frac{AAL5PDU}{48} * 53$$

$$\#C\_ATMHeaderOctets(AAL5PDU) = \frac{AAL5PDU}{48} * (53 - 48)$$

If the concatenated number of ATM layer octets are given or the equivalent number of ATM cells, then the output towards the AAL5 layer can be calculated as follows:

$$\#AAL5LayerOctets(C\_ATMLayerOctets) = \frac{C\_ATMLayerOctets}{53} *48$$

$$\#C\_ATMHeaderOctets(C\_ATMLayerOctets\ ) = \frac{C\_ATMLayerOctets}{53} *(53-48)$$

3.8 Summary

Assuming the PPPoE protocol stack, the total overhead can be calculated as follows:

- PPP: 2 octets
- PPPoE: 6 octets
- MAC: 18 octets provided the payload is at least 46 octets
- the RFC1483/RFC2684/RFC2364 layer and AAL5 layer are more difficult to asses in terms of overhead. RFC1483/RFC2684/RFC2364 because the different options it supports and AAL5 because the padding function. In Table 2. figures are given for a 40 octed IP packet.
- the amount of ATM layer octets can be obtained by taking 10.42% of the AAL5 PDU. Note 10.42 is from (53-48)/48.

Protocol Stack (payload: 40 octet IP packet)	MAC, PPP, PPPoE Overh.	RFC1483+ AAL5 CPCS	Unpadded AAL5 frame	AAL5 Padding	Total
Bridging/LLC-FCS	24	18	82	14	96
Bridging/LLC-nFCS		14	78	18	
Bridging/VC_Mux-FCS		10	74	22	
Bridging/VC_Mux-nFCS		6	70	26	
Routed non-ISO (e.g.IP)VC_Mux	0	8	48	0	48
Routed non-ISO (e.g.IP)/LLC		16	56	40	96
Routed ISO & PPPoA/LLC	2	12	54	42	
Routed ISO & PPPoA/VC_Mux		8	50	46	
PPPoE/LLC-FCS	26	18	84	12	
PPPoE/LLC-nFCS		14	80	16	
PPPoE/VC-Mux - FCS		10	76	20	
PPPoE/VC-Mux - nFCS		6	72	24	
LLC e.g. XID (=6 octets)	/	8	14	34	48

Table 2 *Overview of possible overheads given a 40 octet IP packet.*

4 ANALYSIS

The above summary tells little about the overall efficiency of a protocol encapsulation. First of all Ethernet packets are variable in size and secondly the padding function of AAL5 is capable of "absorbing" headers or trailers that are added by the upper layers.

Therefore the overall efficiency is calculated over a set of IP packet sizes (see Table 3.). This set has either an uniform distribution (1) or a distribution (2) that is regularly encountered on Internet backbones [14].

Packet Size ((1) Uniform Distribution)	Packet Size ((2) Internet Distribution)	Relative Occurrence (%), applies to (2) Internet Distribution only
40	40	52
100	100	1
200	300	5
300	500	4
400	550	1
500	552	2
600	575	6
700	576	7
800	700	2
900	900	1.5
1000	1100	1
1100	1300	2
1200	1400	1.5
1300	1500	14
1400	1492*	14*
1500	28*	14*

Table 3 *IP Packet Distributions*

*: in case of PPPoE 1500 octets have to be replaced by two fragments: 1492 and 28 octets.

1500 Octets was chosen as the maximum IP packet size for this set because it is a common value for Ethernet based networks and is the default MRU for PPP. Going beyond this size (e.g. 9180 octets for Classical IP) would make comparisons difficult because fragmentation would sure happen somewhere along the path between source and destination.

The “line Efficiency” of a protocol stack/encapsulation is defined as follows:

$$LineEfficiency = \frac{Volume\_of\_IPoctets}{53 * Volume\_of\_corresponding\_ATMCells} *100$$

Given the two distributions following results are obtained.

Protocol Stack - Encapsulation	Uniform Distribution (%)	Relative Difference, IP_R = ref	Internet Distribution (%)	Relative Difference, IP_R = ref
Bridging/LLC-FCS	84.4	-1.6	81.0	-0.7
Bridging/LLC-nFCS	84.8	-1.2	81.0	-0.7
Bridging/VC_Mux-FCS	85.1	-0.9	81.4	-0.3
Bridging/VC_Mux-nFCS	85.4	-0.6	81.6	-0.1
Routed non-ISO (e.g.IP)/LLC	**86.0**	**0**	**81.7**	**0**
Routed non-ISO (e.g.IP)VC_Mux	87.0	+1.0	86.3	+4.6
Routed ISO & PPPoA/LLC	86.0	0	81.7	0
Routed ISO & PPPoA/VC_Mux	86.4	+0.4	81.9	+0.2
PPPoE/LLC-FCS	82.9	-3.1	78.9	-2.8
PPPoE/LLC-nFCS	83.5	-2.5	79.0	-2.7
PPPoE/VC-Mux-nFCS	83.8	-2.2	79.4	-2.3
PPPoE/VC-Mux	84.1	-1.9	79.4	-2.3

Table 4 *Encapsulation efficiencies.*

Notice that the efficiency, given the typical Internet distribution, degrades by 3 to 4% w.r.t. the uniform distribution. An exception is the Routed non-ISO PDU’s employing VC_Mux encapsulation (IP directly into ATM). This because TCP Ack’s occurring some 50% within 100 given IP packets require only a single ATM cell.

5 SIDE EFFECT: IP FRAGMENTATION

In the above listed encapsulations, PPPoE requires special attention. Because of its location between the IP and MAC layer it induces fragmentation. On a PPPoE-IP interface, IP packets are encapsulated trice: first in a PPP frame, subsequently in a PPPoE frame and finally in a MAC frame. As the payload of standard MAC frames cannot grow beyond 1500 octets (IEEE 802.3-1998 Standard), the IP payload must be limited to allow for the PPP and PPPoE headers. To do so the maximum IP MTU size of a PPPoE-IP interface is set to 1492 octets (1500 - 2 (PPP) - 6 (PPPoE)). For external ADSL routers this results in the side effect described below.

Assume an ADSL router with two IP interfaces (see Figure 9):

- an ATM/MAC/PPPoE/PPP interface
- a standard Ethernet interface.

IP packets are submitted by the end-systems attached to the Ethernet interface of the ADSL router. Each 1500 octet IP packet arriving on the Ethernet interface and to be forwarded over the PPPoE-IP interface must be fragmented into a 1492 octet fragment and a 28 octet fragment (28= 20 octet IP header + 8 octet payload).

Note The Internet Protocol [13] states: "If an Internet datagram is fragmented, its data portion must be broken on 8 octet boundaries."

A consequence is that the "Line Efficiency" is no longer equal to the "Transfer efficiency" and this should be taken into account if comparisons are made.

To circumvent fragmentation, one might say that conceptually the ATM connection is a *virtual* Ethernet segment. More specific it uses the MAC frame format but does not rely on the CSMA/CD access mechanism as it is a point-to-point connection. In contrast a *real* Ethernet implementation must comply with the minimum and maximum MAC frame sizes to guaranty fair access to the medium. To avoid fragmentation, virtual Ethernet segments might deviate from the standard e.g. 1508 octets instead of 1500 octets. The last row in the table below reveals the efficiency for such a case.

Protocol Stack - Encapsulation	Uniform Distribution	Internet Distribution
PPPoE/LLC-nFCS (Line Efficiency)	83.7	79.0
PPPoE/LLC-nFCS (Transfer Efficiency)	83.5	78.5
PPPoE/LLC-nFCS (Line Efficiency, max MTU= 1508 octets)	83.8	79.6

Table 5

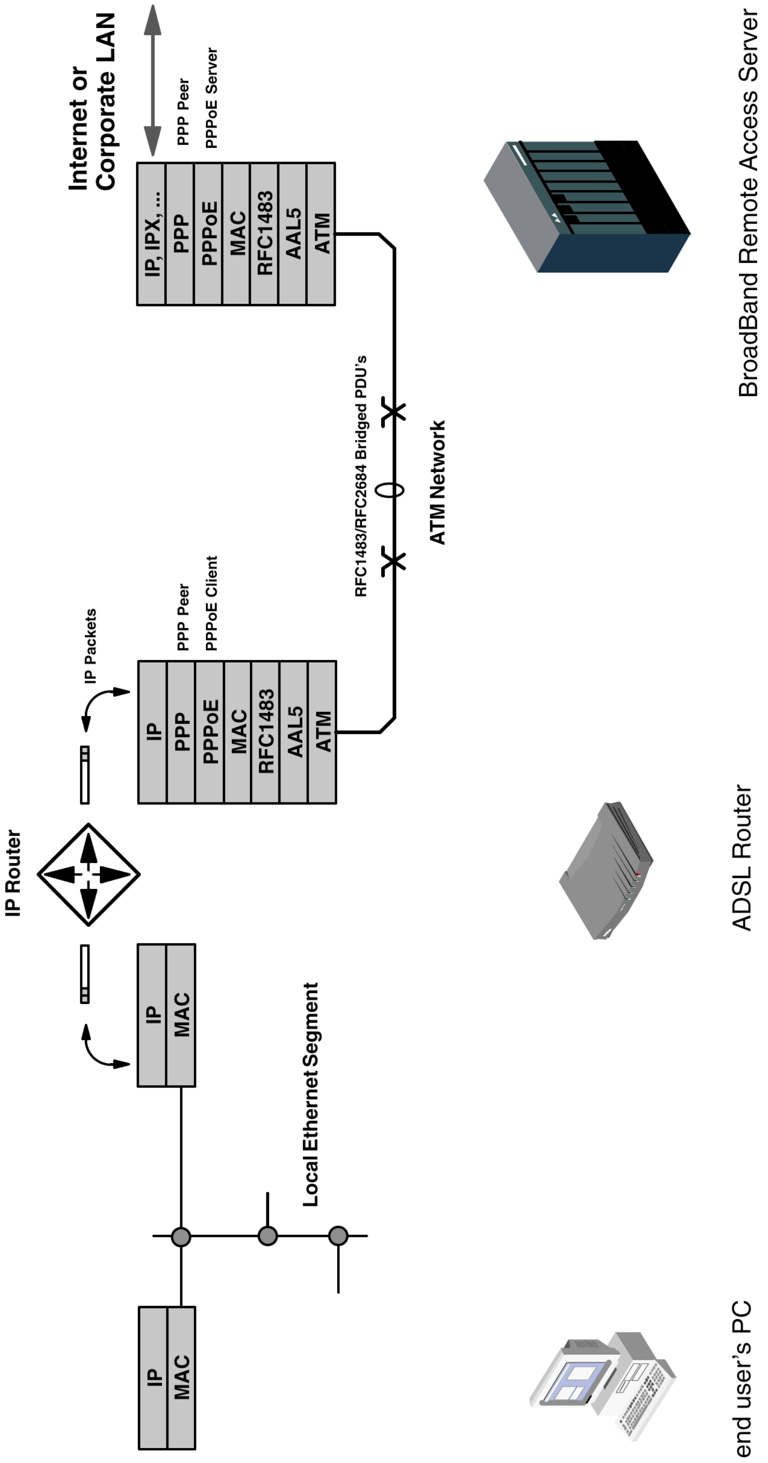


Figure 9 *ADSL Router with PPPoE interface.*

6 CONCLUSION

Generally speaking the “line efficiency” of a protocol stack/encapsulation is roughly 80% given the Internet packet distribution. The VC-Mux encapsulation of IP PDU’s generates the least overhead and the PPPoE encapsulation the most. The influence of fragmentation due to PPPoE is around 1% however the communication processor embedded in ADSL Bridge - Routers performing the fragmentation/reassembly might be a more important factor in overall performance degradation.

APPENDIX A REFERENCES

[1] User Multiplexing in a single PPP/PPPoE session vs. PPPoE session multiplexing. THOMSON Internal Technical Note - Dirk Van Aken, Gert Marynissen, Stan Claes.

[2] RFC1577/RFC2225 Classical IP and ARP over ATM.

[3] RFC2516 A Method for Transmitting PPP over Ethernet (PPPoE).

[4] RFC1661: The Point-to-Point Protocol (PPP).

[5] RFC 2364: PPP over AAL5.

[6] Ethernet V2.0 The Ethernet, a Local Area Network Data Link Layer and Physical Layer Specification. Digital Equipment Corporation, Intel Corporation and Xerox Corporation. Blue Book Version 2.0 November 1982.

[7] IEEE802.3 (IEEE Std 802.3, 1998 Edition).
Information technology--Telecommunications and information exchange between systems--Local and metropolitan area networks--Specific requirements.
Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) access method and physical layer specifications.

[8] IEEE802.3ac IEEE Std 802.3ac 1998.
Frame Extensions for Virtual Bridged Local Area (VLAN) Tagging on 802.3 Networks.

[9] RFC1483/RFC 2684 Multiprotocol Encapsulation over ATM Adaptation Layer 5.

[10] RFC1638 PPP Bridging Control Protocol.

[11] ITU-T I.363.5 B-ISDN ATM Adaptation Layer Specification: Type 5 AAL.

[12] ITU-T I.361 B-ISDN ATM Layer Specification

[13] RFC 791 The Internet Protocol

[14] Trends in Wide Area IP Traffic Patterns (www.caida.org)..

APPENDIX B ABBREVIATIONS & CLARIFICATION OF TERMS

Abbreviation	Clarification
AAL5	ATM Adaptation Layer 5
ADSL	Asymmetric Digital Subscriber Line
ATM	Asynchronous Transfer Mode
BBRAS	BroadBand Remote Access Server
CPE	Customer Premises Equipment
CSMA/CD	Carrier Sense Multiple Access/Collision Detect
MTU	Maximum Transfer Unit
MRU	Maximum Receive Unit
PDU	Protocol Data Unit
PPP	Point to Point Protocol
PPPoA	PPP over ATM
PPPoE	PPP over Ethernet
PPTP	Point to Point Tunnelling Protocol
IP	Internet Protocol
ISP	Internet Service Provider
LAN	Local Area Network
LCP	Link Control Protocol
LLC	Logical Link Control
MAC	Medium Access Control
NCP	Network Control Protocol
RFC	Request For Comments
SLIP	Serial Line Internet Protocol
SNAP	SubNetwork Access Protocol
TCP	Transmission Control Protocol
UDP	Unreliable Data Protocol
WAN	Wide Area Network

Visit us at:

www.speedtouch.com

Acknowledgements

All Colleagues for sharing their knowledge.

Coordinates

THOMSON

Prins Boudewijnlaan 47
B-2650 Edegem
Belgium

Email: documentation.speedtouch@thomson.net

Copyright