

# Standards for the Encapsulation of LAN and WAN over ATM

A number of standards have been developed which describe the encapsulation of LAN and WAN protocols over ATM. These standards allow users to integrate ATM into existing WANs and/or LANs, thus providing various cost-effective upgrade routes. This paper describes the various encapsulation methods in general.

## VC-Based Encapsulation

The following methods are used for encapsulation or transporting LAN or WAN protocols via ATM:

### VC-Based (null) Multiplexing

VC-based multiplexing uses one Virtual Channel (VCI/VPI pair) for each protocol. In this way, the protocol is carried directly over the AAL5 PDU. Since no additional information is added to the protocol, this is sometimes called null encapsulation.

For routed protocols e.g., TCP/IP, IPX, the PDU is carried directly in the payload of the AAL5 CPCS PDU. For bridged protocols e.g., Ethernet, Token Ring, FDDI, all fields following the PID field are carried in the payload of the AAL5 CPCS PDU.

### Multiprotocol Encapsulation over ATM

One method of transmitting LAN protocols over ATM is by using LLC encapsulation with the IETF RFC1483 standard. In this instance, the LLC and SNAP headers within the AAL5 PDU are used to identify the encapsulated protocol. This protocol stack is shown in the following illustration.

Upper-layer applications Upper-layer protocols
Ethernet or TCP/IP
SNAP
802.2 LLC
AAL5
ATM
Physical
<i>LAN protocols encapsulated over ATM</i>

In multiprotocol encapsulation, the PDUs are carried in the Payload field of the Common Part Convergence Sublayer (CPCS) PDU of ATM Adaptation Layer type 5 (AAL5). In this encapsulation method, the LLC header is always 0xAA-AA-03, which indicates that the LLC header is followed by a SNAP header.

DSAP	SSAP	Ctrl
AA	AA	03
<i>LLC header</i>		

The format of the SNAP header is as follows:

<b>Organisationally Unique Identifier (OUI)</b>	<b>Protocol Identifier (PID)</b>
3 bytes	2 bytes
<i>SNAP header</i>	

Non-ISO PDUs e.g., TCP/IP, IPX, are identified by an OUI value of 0x00-00-00, followed by the PID, which represents the 2-byte EtherType field. For example, an EtherType value of 0x08-00 identifies an IP PDU.

LLC 0xAA-AA-03
OUI 0x00-00-00
EtherType (2 bytes)
Non-ISO PDU (up to 65527 bytes)
<i>Payload format for routed non-ISO PDUs</i>

Non-routed bridged protocols are identified by an OUI value of 0x00-80-C2 followed by the 2-byte PID which specifies the protocol which follows. The following table lists some possible values for the PID field in this instance.

PID Value...	Protocol...
0x00-01/0x00-07	Ethernet/802.3
0x00-03/0x00-09	Token Ring/802.5
0x00-04/0x00-0A	FDDI

The following diagrams represent the payload format for the Ethernet, Token Ring, FDDI and SMDS PDUs.

LLC 0xAA-AA-03	LLC 0xAA-AA-03	
OUI 0x00-80-C2	OUI 0x00-80-C2	
PID 0x00-04 or 0x00-0A	PID 0x00-0B	
PAD 0x00-00-00	Reserved	BETag
Frame Control (1 byte)	Common PDUHeader	
MAC destination address	BAsize	
remainder of MAC frame	MAC destination address	
LAN FCS (if PID is 0x00-01)	remainder of MAC frame	
<i>Payload format for bridged FDDI PDUs</i>	LAN FCS (if PID is 0x00-01)	
	<i>Payload format for bridged SMDS PDUs</i>	

## Classical IP and ARP over ATM (RFC 1577)

The standard IETF RFC1577 describes a version of ARP and InARP which is appropriate for ATM. According to this standard, the LLC and SNAP headers identify the encapsulated protocol as described above in Multiprotocol Encapsulation over ATM. The LLC header has the value 0xAA-AA-03, which indicates that the LLC header is followed by a SNAP header. The OUI value for ARP is 0x00-00-00, followed by the EtherType field of value 0x08-06.

Internet addresses are assigned independently of ATM addresses. Each host implementation must know its own IP and ATM address(es) and must respond to address resolution requests appropriately. IP members must also use ATMARP and InATMARP to resolve IP addresses to ATM addresses when needed.

The ATMARP PDU is shown in the following illustration.

8	16	24	32
Hardware type		Protocol type	
Type and length of source ATM number	Type and length of source ATM subaddress	Operation	
Length of source protocol address	Type and length of target ATM number	Type and length of target ATM sub-address	Length of target protocol address
Source ATM number (q bytes)		Source ATM subaddress (r bytes)	
Source protocol address (s bytes) (IP=4 bytes)			
Target ATM number (x bytes)		Target ATM subaddress (y bytes)	
Target protocol address (z bytes) (IP=4 bytes)			
<i>Payload format for ATMARP PDU</i>			

ATMARP and InATMARP use the same hardware type, protocol type and operation code data formats as ARP and InARP. The location of these fields within the ATMARP packet are in the same byte position as those in ARP and InARP packets. A unique hardware type value has been assigned for ATMARP.

## Frame Relay/ATM Network Internetworking (FRF.5)

FRF.5 describes network internetworking. This is applicable when 2 distant Frame Relay networks are connected over an ATM backbone. This function encapsulates the Frame Relay frames inside the ATM cells. It requires software that can be added to the ATM switch (or anywhere else between or in the ATM/Frame Relay networks).

## Frame Relay/ATM Internetworking (FRF.8)

Mapping between ATM PDUs and Frame Relay PDUs requires examination of the incoming ATM AAL5 CPCS PDU payload header or Frame Relay Q.922 PDU payload header to determine the type, and then to overwrite the incoming header with the outgoing header.

The following diagram shows the translation between the Frame Relay Q.922 PDU payload header and the ATM AAL5 CPCS PDU payload header.

Control 0x03	Pad 0x0	LLC 0xAA-AA	
NLPID 0x80	OUI 0x00	0x03	OUI 0x00
0x80-C2		0x80-C2	
PID		PID	
Remainder of MAC frame	Pad 0x00-00		
	Remainder of MAC Frame		

Refer to Frame Relay Forum FRF.8 for detailed information concerning the provisions for traffic management between the two protocols.

## LAN Emulation

The LAN emulation protocol uses control messages to set up the LAN. LAN Emulation provides for two possible data packet formats: Ethernet and Token Ring. The LAN emulation data frames preserve all the information contained in the original 802.3 or 802.5 frames, but add a 2-byte LEC ID (the source ID), which is unique to each LEC.