Project Report
NASA/G-2

Internet over the VDL-2 Subnetwork— the VDL-2/IP Aviation Datalink System

R.D. Grappel

20 October 2000

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS

Prepared for the National Aeronautics and Space Administration
John H. Glenn Research Center, Lewis Field, Cleveland, OH 44135

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No.
NASA/G-2

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle

Internet over the VDL-2 Subnetwork—the VDL-2/IP Aviation Datalink System

5. Report Date
20 October 2000

6. Performing Organization Code

7. Author(s)
R.D. Grappel

8. Performing Organization Report No.
NASA/G-2

9. Performing Organization Name and Address
MIT Lincoln Laboratory
244 Wood Street
Lexington, MA 02420-9108

10. Work Unit No. (TRAIS)

11. Contract or Grant No.
NASA Glenn

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Glenn Research Center
Cleveland, OH 44135

13. Type of Report and Period Covered
Project Report

14. Sponsoring Agency Code

15. Supplementary Notes

This report is based on studies performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology, under Air Force Contract F19628-00-C-0002.

16. Abstract

This report describes a design to operate the standard Internet communications protocols (IP) over the VHF aviation Data Link Mode 2 (VDL-2) subnetwork. The VDL-2/IP system specified in this report can operate transparently with the current aviation users of VDL-2 (ACARS and ATN) and proposed users (FIS-B). The VDL-2/IP system provides a straightforward mechanism to utilize inexpensive, "commercial off-the-shelf" (COTS) communications packages developed for the Internet as part of the aviation datalink system.

Section 2 of this report gives background information about aviation datalink applications using the aviation VHF frequencies. Section 3 describes the VDL-2 system in greater depth. Section 4 describes the "ACARS Over AVLC" (AOA) system developed by ARINC to allow "legacy" ACARS applications to operate over VDL-2. AOA provides the basic communications protocol mechanism for VDL-2/IP. Section 5 describes the proposed FIS-B system, and how it can interoperate with AOA (and VDL-2/IP). Section 6 covers the Internet "Address Resolution Protocol" (ARP) that is used (with minor modifications) to support the VDL-2/IP interface. Section 7 describes the details of the proposed VDL-2/IP design. Section 8 describes the Internet Mobile Routing protocols to be used at the network level (VDL-2/IP as described in section 7 is a link-layer protocol) in order to provide fully-mobile IP routing support. And finally, section 9 summarizes the VDL-2/IP design and the open issues remaining in its specification.

17. Key Words

Internet	ACARS	IPv4
Datalink	ARP	IPv6
Mobile IP	VDL-2	TCP
ATN	IP	FIS-B

18. Distribution Statement

This document is available to the public through the
National Technical Information Service,
Springfield, VA 22161.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	78	

Reproduction of completed page authorized

ABSTRACT

This report describes a design to operate the standard Internet communications protocols (IP) over the VHF aviation Data Link Mode 2 (VDL-2) subnetwork. The VDL-2/IP system specified in this report can operate transparently with the current aviation users of VDL-2 (ACARS and ATN) and proposed users (FIS-B). The VDL-2/IP system provides a straightforward mechanism to utilize inexpensive, "commercial off-the-shelf" (COTS) communications packages developed for the Internet as part of the aviation datalink system.

Section 2 of this report gives background information about aviation datalink applications using the aviation VHF frequencies. Section 3 describes the VDL-2 system in greater depth. Section 4 describes the "ACARS Over AVLC" (AOA) system developed by ARINC to allow "legacy" ACARS applications to operate over VDL-2. AOA provides the basic communications protocol mechanism for VDL-2/IP. Section 5 describes the proposed FIS-B system, and how it can inter-operate with AOA (and VDL-2/IP). Section 6 covers the Internet "Address Resolution Protocol" (ARP) that is used (with minor modifications) to support the VDL-2/IP interface. Section 7 describes the details of the proposed VDL-2/IP design. Section 8 describes the Internet Mobile Routing protocols to be used at the network level (VDL-2/IP as described in section 7 is a link-layer protocol) in order to provide fully-mobile IP routing support. And finally, section 9 summarizes the VDL-2/IP design and the open issues remaining in its specification.

ACKNOWLEDGMENTS

The author wishes to acknowledge the considerable assistance received from Robert J. Nelsen of Aeronautical Datalink Consultants in the design of the VDL-2/IP system and in the preparation of this paper. Mr. Nelsen provided extensive background into the design of VDL systems and helped the author work his way through the various VDL and protocol standards documents. Mr. Nelsen was a "sounding board" for numerous initial concepts and design ideas that led up to VDL-2/IP.

In addition, the author wishes to thank Stephen Kolek of MIT Lincoln Laboratory, Group 61, for his assistance in explaining the function and impact of Reed Solomon error-correcting codes on the VDL-2 overall error rate.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (CONT.)

LIST OF TABLES

Table No. **Page**

1. Introduction

The meteoric growth of the Internet has caused implementations of the Internet communications protocols to become both ubiquitous and inexpensive. The TCP/IP (Internet) protocol suite is now a standard part of the system software/firmware on PCs, workstations, and embedded controllers. A single-chip hardware implementation of the protocol stack is commercially available (see Reference 1). The standards for TCP/IP are freely available from the Internet Engineering Task Force (IETF) web site and in many books (Reference 2 for example). The TCP/IP protocols are, by far, the most used and tested "open systems" communications methods in use today.

In parallel with the growth of the Internet, the increase in aviation applications of datalink has created a need for more capable, higher speed, air/ground networks. (Note: in this paper, the term "datalink" generally refers to air-ground communications systems, including HF, VHF, UHF, SATCOM, and Mode S.) The proliferation of aviation datalinks has been accompanied by the creation of multiple datalink communications protocols, each tailored to its specific link medium and/or application. It would seem compelling that the Internet protocols can provide an effective solution to the communications protocol needs of many aviation datalink applications. (The selection criteria for aviation datalink protocols are discussed in Reference 3). In addition, the Internet today provides a ubiquitous, worldwide ground infrastructure that supports a wide variety of applications. Utilizing the Internet internetworking protocols as the basis for aviation datalink applications would appear to enable inexpensive and simple ground infrastructures for aviation datalinks, as opposed to using aviation-specific datalink architectures running over aviation-specific communications networks.

One of the new air/ground datalinks coming into use is the VHF Data Link Mode 2 (VDL-2). VDL-2 provides a high-speed, bit-oriented link between aircraft and ground stations, replacing the much slower "Airline Communications and Reporting System" (ACARS) datalink in use today. Designed as a subnetwork for the ICAO Aeronautical Telecommunications Network (ATN), VDL-2 supports a connection-mode, addressable datalink with an ISO 8208 network interface. The ATN is based on a tailored and modified set of ISO protocols, not compatible with the Internet.

However, the VDL-2 specification also provides a means for aircraft systems to transition from ACARS applications to the new ATN applications. Utilizing a lower-level VDL-2 interface (separate from the ISO 8208 ATN interface) and an extension to the VDL-2 transitional mechanism, this paper will define a datalink system that allows the VDL-2 equipment (both avionics and ground stations) to serve as an Internet subnetwork capable of inter-operating with the ATN or standing alone. Requiring no new hardware, and very little added software, the new system (termed "VDL-2/IP") will allow users of VDL-2 equipment to attach their applications to standard Internet infrastructures.

Section 2 of this paper gives background information about existing and proposed datalink applications to be operated on the aviation VHF frequencies. These applications include the “Airline Communications and Reporting System” (ACARS), “VHF Data Link” (VDL), and “Flight Information Services via Broadcast” (FIS-B). Section 3 of this paper goes into greater depth describing the VDL-2 system. Section 4 of this paper describes the transitional system “ACARS Over AVLC” (AOA) that provides a means for “legacy” applications (such as ACARS) to utilize the VDL-2 network. The AOA technique that allows alternate application protocols to coexist on the VDL-2 network forms the basis of the new VDL-2/IP design. (It should be noted that an alternate system design supporting ACARS applications over VDL-2 via the complete ATN protocol stack has also been developed. Both approaches may end up being supported over VDL-2.) Section 5 of this paper describes how the FIS-B applications operate over VDL-2. Section 6 of this paper describes the Internet “Address Resolution Protocol” (ARP) that will be used (with minor modifications) to support the VDL-2/IP interface. Section 7 of this paper gives the details of the proposed VDL-2/IP design, drawn from the functions described in previous sections. Section 8 of this paper describes the Mobile Routing protocols to be used at the network layer (VDL-2/IP as defined in section 7 is a link-layer protocol) in order to provide fully-mobile IP routing support. Finally, section 9 of this paper summarizes the factors remaining to be specified for the VDL-2/IP design.

2. Background

This section will give brief overviews of the current state of aviation datalink systems utilizing the aviation VHF frequencies. It will cover both those systems that are presently operational and those undergoing specification and development. The operational system is the "Airline Communications and Reporting System" (ACARS). ACARS (see References 4-6) is supported by "Aeronautical Radio Incorporated" (ARINC) and "State-of-the-art Information Technologies in Aviation" (SITA) to supply Air Traffic Services (ATS), Airline Administrative Control (AAC), Airline Operational Control (AOC), and Airline Passenger Communications (APC) datalink world-wide. "VHF Data Link" (VDL) is a new system under development and specification intended to replace ACARS with a far faster and more capable datalink. VDL is specified to form one subnetwork of the Aeronautical Telecommunications Network (ATN). Several variations on VDL exist – this paper will deal exclusively with VDL Mode 2. VDL-2 is the immediate successor to ACARS with the least detrimental impact on the current applications of the aviation VHF frequencies. VDL-2 has the support of the airlines and avionics manufacturers. VDL-2 ground stations are scheduled to be deployed in the U.S. and Europe by 2001. Radio equipment capable of VDL-2 operation is also planned for introduction in 2001. One application of a subset of the VDL-2 functionality under current specification and development is the "Flight Information Service via Broadcast" (FIS-B) (see Reference 7). FIS-B systems will utilize the VDL-2 datalink for uplink broadcast transmission of weather data (and other non-critical ground-derived information) to aircraft.

2.1 ACARS

ACARS is a character-oriented VHF datalink system presently in use by airlines worldwide for a variety of ATS, AAC, AOC, and APC datalink applications. Developed and maintained by ARINC in the U.S. and by SITA in Europe, ACARS uses basic "Amplitude Modulated – Minimum Shift Keying" (AM-MSK) and a non-adaptive "Carrier-Sense Multiple Access" (CSMA) media control protocol over the RF link to achieve a raw throughput of 2,400 bits per second. ACARS provides an effective range (limited by line-of-sight) of around 160 nautical miles while utilizing 25 watts of transmitter power (airborne) or 50 watts (ground). ACARS uses a centralized topology where all aircraft downlinks received by ACARS ground stations anywhere in the network are relayed to a central processor (located at ARINC in Annapolis, MD, and SITA processors in France and Singapore). The central processor eliminates any duplicate downlinks received and relays its uplink response to the ground station having the best signal quality with the given aircraft. Further details of ACARS may be found in References 4-6.

As stated above, ACARS is a character-oriented datalink. In order to support modern bit-oriented applications over ACARS, ARINC has developed a special protocol called "622" (see Reference 8) that causes each 4-bit block of the user's binary data to be converted into a hexadecimal ACARS character. The ARINC 622 protocol also appends a 16-bit "cyclic redundancy check" (CRC) to each data block for purposes of error detection. The ARINC

622 protocol results in an inherent doubling of the message size (4 bits of binary data into one 8-bit ACARS character), but the added efficiency of binary message encoding can still result in an overall increase in datalink bandwidth efficiency. The ARINC 622 protocol provides a transition mechanism from existing ACARS systems to more modern binary aviation VHF datalinks, such as VDL.

2.2 VDL

VHF Datalink (VDL) is a replacement system for ACARS. VDL Mode 2 (VDL-2) provides high-speed, bit-oriented datalink functions over the same aviation VHF frequency spectrum now supporting ACARS. VDL-2 utilizes differential 8-phase shift keying (D8PSK) and a "p-persistent Carrier-Sense Multiple Access" (p-CSMA) protocol over the RF link to achieve a raw data rate of 31,500 bits per second (3 bits/symbol, and 10,500 symbols/second). The bit error rate (BER) of the VDL-2 RF link is specified at approximately 10^{-4} assuming a signal level at the receiver of –87 dBm. (The VDL-2 BER increases to approximately 10^{-3} for a signal level at the receiver of –98 dBm.) Reed-Solomon "Forward Error Correction" (FEC) is applied to the decoded RF data (RS(255,249) 2^8 coding is specified for VDL) to correct up to three erroneous bytes in any 255-byte data block. Figure 2-1 below illustrates the impact of the FEC algorithm on the VDL-2 BER at the "Media Access Control" (MAC) interface (assuming that RF bit errors are uniformly distributed in the message). Beyond the FEC, a 16-bit link-layer Cyclic Redundancy Check (CRC) is used to give the VDL-2 link an overall undetected bit error rate (BER) approaching 10^{-9}. VDL-2 can achieve a range (limited by line-of-sight) of over 160 nautical miles using a 15 Watt transmitter. (When an aircraft is on the ground, its transmitter power is limited to 4 watts, and its effective range is reduced to about 100 nautical miles.) VDL-2 defines a distributed topology with mechanisms for link establishment and handoff to set up and manage air-ground connections.

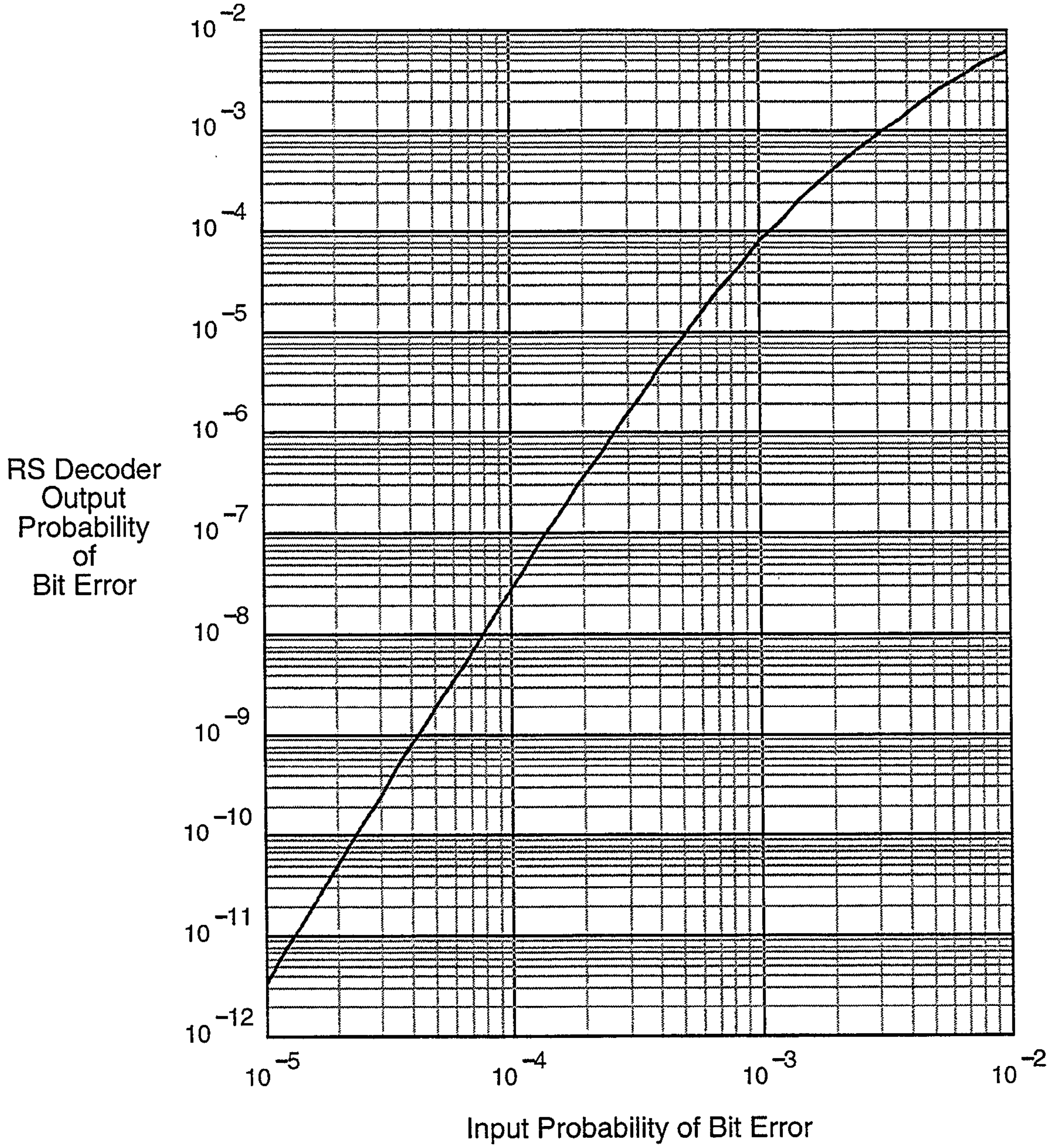


Figure 2-1 Impact of Reed Solomon FEC on VDL-2 BER

A second VDL mode (VDL-3) is also under current development. VDL-3 combines the functions of digital datalink (similar in data bandwidth to VDL-2) with the capability of providing concurrent digital voice radio communications. The requirements of digital voice add a great deal of complexity to the VDL-3 design. They also have a dramatic impact on the spectrum utilization of VDL-3, and they may cause difficulties in the fielding of VDL-3 during the transition from the current voice and datalink radios. This paper will deal entirely with VDL-2, as this will be the first VDL mode fielded, and it provides all the capabilities required for the weather information dissemination application that is the goal of this paper.

2.3 FIS-B

The "Flight Information Services via Broadcast" (FIS-B) system is currently being developed in the U.S. and specified by the RTCA (see Reference 7). FIS-B is intended as an inexpensive means to uplink ground-derived, near real-time weather data (both textual and graphic) to aircraft – both GA and commercial. FIS-B will use a subset of the VDL-2 subnetwork specification. As its name suggests, FIS-B will utilize only the uplink broadcast capability of the VDL-2 subnetwork. FIS-B avionics will require only a VDL receiver, and no support for the VDL-2 "management entities" will be necessary. It is intended that FIS-B applications will be able to inter-operate with full VDL-2 implementations (as would be expected in commercial aircraft supporting the ATN).

3. VHF Data Link Mode 2 (VDL-2)

This section describes in more detail the VHF Data Link Mode 2 system, over which the Internet protocols will be applied. The VDL-2/IP design builds upon the existing VDL-2 subnetwork architecture supporting ATN and FIS-B applications. In general, only those areas of the VDL-2 design that bear on the VDL-2/IP system will be elaborated here. The complete specifications for VDL-2 can be found in the cited references.

3.1 VDL-2 Physical Layer

As was described in section 2.2 above, VDL-2 utilizes D8PSK modulation and p-persistent CDMA for its media-access (MAC) processing. The net result is a 31,500 bits/second raw transmission rate on the RF channel. VDL-2 provides for fully-binary messages (as opposed to the character-oriented ACARS). VDL-2 is based on a distributed network topology with multiple routers selecting the best message path (unlike the centralized ACARS topology). The VDL-2 system utilizes a sophisticated method of link establishment and handoff to control the selection of a ground station to communicate with a given aircraft (while ACARS is a broadcast system).

The format of a VDL-2 transmission "burst" is shown in Figure 3-1 below. Each transmission burst starts with a ramp-up of transmitter power (a string of zero symbols), followed by a synchronization string used to identify the start of the data. The "reserved symbol" might be used in future versions of the system to identify alternate formats – it is set to zero for now. The next value in the VDL-2 transmission is the length (in bits) of the VDL-2 frame. The maximum VDL-2 burst length is 131,071 bits. The VDL-2 header block is preceded by Forward Error Correction (FEC) Reed Solomon code check value that is used to correct errors in the header. Following the header is the VDL-2 data, interleaved with FEC for the data. The VDL-2 FEC algorithms (combined with the link-layer frame-check described in section 3.2 below) will yield an expected overall undetected BER of approximately 10^{-9}. (See Figure 2-1 above for the impact of the FEC on the BER of VDL-2.) The VDL-2 transmission burst concludes with a transmitter ramp-down.

The VDL-2 data is further broken down into "frames". (VDL-2 frames are discussed in Section 3.2 below.) A VDL-2 transmission burst may contain one or more frames. The default maximum length for a VDL-2 frame is 8,312 bits. (Note: VDL-2 provides for negotiation of parameters between the sender and receiver. (See Section 3.6 for a further discussion of VDL-2 parameter negotiation.) The maximum frame size available in the VDL-2 system is 16,504 bits.

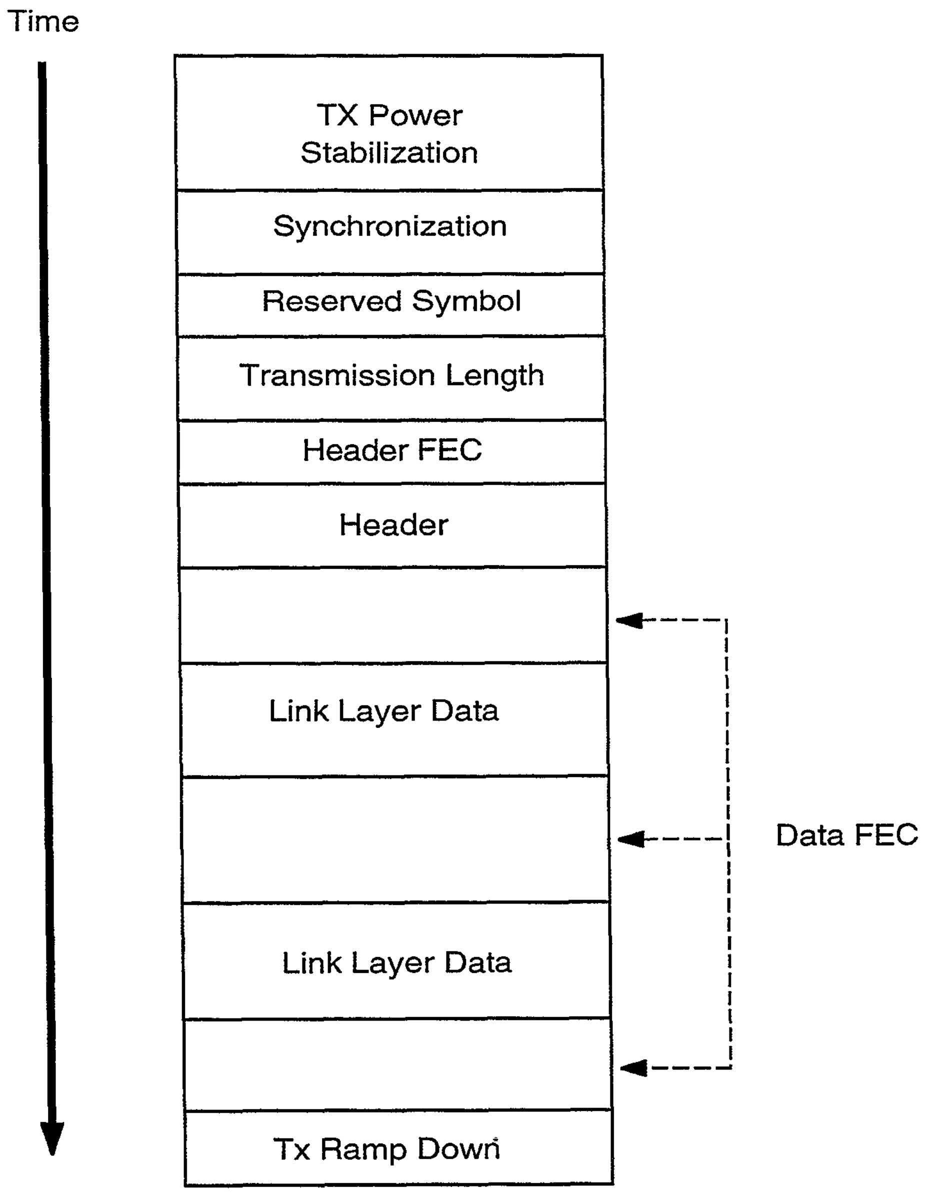


Figure 3-1 VDL-2 Transmission Burst Format.

3.2 VDL-2 Data Link Layer

The data link protocol used by the VDL-2 system is an extended version of the standard High Level Data Link Control (HDLC) protocol defined in References 9-12. The VDL-2 variant of the HDLC protocol is called "Aviation VHF Link Control" (AVLC). AVLC extends HDLC by providing mechanisms for VDL link establishment and handoff. Effectively, AVLC is able to maintain a consistent VDL-2 air-ground "connection" even though the aircraft moves from the coverage of a given VDL-2 ground station to another, and may need to retune its radio frequency from time to time. These AVLC extensions are implemented in software functions termed the "VDL Management Entity" (VME) and the "Link Management Entity" (LME) that are hosted in the VDL-2 "Communications Management Unit" (CMU). The VME (defined in Reference 13) decides when to establish a link and performs the necessary handoffs. The VME also notifies the other software entities about the status of the VDL connection. The LMEs (also defined in Reference 13) manage a given air-ground connection – there will be an LME instantiation for each connection that a given CMU is currently maintaining. The LME performs link establishment (at the command of the VME) and controls the flow of VDL-2 frames between the aircraft and ground station. The LME also informs the VME of changes in its link status.

Figure 3-2 below illustrates the structure of an AVLC Frame. Each frame is delimited by "flag" bytes that provide for synchronization of the VDL-2 bit stream. Note that the AVLC frame is of variable length – the ending flag byte marks the termination of a given frame (and, possibly, the start of another frame). The flag bytes have a unique value (0x7E) that cannot occur anywhere inside the AVLC frame (just as in HDLC). To prevent this, the AVLC protocol (just as the HDLC protocol) performs a function termed "bit stuffing". If a pattern of five consecutive 1-bits are encountered in the data, the AVLC protocol inserts an extra zero bit during transmission. The AVLC protocol replaces these extra zero bits with one bit during reception. Bit stuffing is transparent to the higher VDL-2 protocol levels.

AVLC Frame

Flag	Address	Link Control	Information Field	FCS	Flag
1	8	1	n	2	1

Figure 3-2 AVLC Frame Structure.

The AVLC frame starts with 8 bytes of addressing. (AVLC addressing will be described further in the next section.) There is a single byte of link control (also described further below). Next comes the variable-length data field of the frame. The AVLC frame concludes with a 16-bit Frame Check Sequence (FCS) that provides a high level of link error detection. (This AVLC FCS link error detection operates on top of the VDL-2 physical layer FEC error detection and correction algorithm.) An AVLC frame that fails its FCS test is immediately discarded. (Note: the link control and FCS functions of AVLC are identical to HDLC. Software and/or hardware COTS implementations of the standard HDLC protocols can be used with little or no change in AVLC applications.)

3.3 VDL-2 Addressing

An expanded illustration of the AVLC frame is given in Figure 3-3 below. Note that the 8 total bytes of AVLC addressing are divided into a 4-byte destination address and a 4-byte source address. The structure of the addresses adheres to the ISO 3309 protocol (Reference 9) for extensible HDLC addressing. Following ISO 3309, the low-order bit of each address byte except the last one is cleared to zero. The low-order bit of the last address byte is set to one. The overall structure of VDL-2 addresses is defined in Reference 18.

			BIT NUMBER						First Bit Transmitted	
	Description	BYTE	8	7	6	5	4	3	2	1
	FLAG	--	0	1	1	1	1	1	1	0
ISO 3309 Address Format	Destination Address Field	1	D22	D23	D24	D25	D26	D27	**A/G**	0
		2	D15	D16	D17	D18	D19	D20	D21	0
		3	D8	D9	D10	D11	D12	D13	D14	0
		4	D1	D2	D3	D4	D5	D6	D7	0
	Source Address Field	5	S22	S23	S24	S25	S26	S27	**C/R**	0
		6	S15	S16	S17	S18	S19	S20	S21	0
		7	S8	S9	S10	S11	S12	S13	S14	0
		8	S1	S2	S3	S4	S5	S6	S7	1
	Link Control	9				**P/F**				
	User Data	10 -- -- N-2								
	Frame Check	N-1								
		N								
	FLAG	--	0	1	1	1	1	1	1	0

Figure 3-3 AVLC Frame Structure (expanded).

The second bit of the AVLC destination address is reserved for the "air-ground" indicator. The A/G bit is cleared to zero if the source of the AVLC frame is an aircraft and the aircraft is not currently located on the ground (as determined by a "squat switch" in the aircraft's landing gear). An aircraft lacking such a switch always reports "in the air". A VDL-2 ground station sets the A/G bit to one (the same as would an aircraft that detects it is currently "on the ground"). This bit is used in the VDL-2 management protocols.

The second bit of the AVLC source address is reserved for the "command/response" indicator. The C/R bit is cleared to zero to indicate an AVLC command frame – it is set to one to indicate a response frame. This bit is used in the VDL-2 management protocols.

As a result of the use of ISO 3309 extensible-address protocol and the reservation of a bit in each address for a special VDL-2 system indicator, an AVLC address has only 27 usable bits. These 27 bits are further subdivided into a 3-bit address type (in the high-order bits) and a 24-bit station address, as shown in Table 3-1 below. (See Reference 15 for the ICAO assignment of 24-bit aircraft addresses.) Two sets of VDL-2 ground station addresses are provided for in this design – one set for ICAO global administration, and a second set for delegation to independent operators of VDL-2 networks (such as the FIS-B system in the U.S. specified by Reference 7).

TABLE 3-1 VDL-2 Basic Addressing Encoding

Upper 3 bits	Address Type	Comments
000	Reserved	Future use
001	Aircraft	24-bit ICAO address
010	Reserved	Future use
011	Reserved	Future use
100	Ground station	ICAO administered
101	Ground station	ICAO delegated
110	Reserved	Future use
111	All stations	Broadcast

AVLC broadcast addressing is further encoded as indicated in Table 3-2 below. Note that, in general, an AVLC 24-bit broadcast address has its 3-bit type field and all ones in the remaining 21 bits. The one exception is a broadcast directed to all ground stations of a particular ICAO-delegated service provider. In this case, a set of leading bits in the 21-bit "specific address field" denote the provider, while the remainder of the bits are set to one.

TABLE 3-2 VDL-2 Broadcast Addressing Encoding

Broadcast Destination	Type bits	Specific Address Field
All aircraft	001	All ones
All ICAO ground stations	100	All ones
All ground stations of a particular provider	101	Variable-length provider code, remainder all ones
All ground stations	101	All ones
All stations (air & ground)	111	All ones

3.4 VDL-2 Link Control

The structure of the AVLC link control field (derived from the HDLC protocol) is illustrated in Figure 3-4 below. (See Reference 10 for a complete definition of the link control field.) There are three types of AVLC frames. Information frames carry user data that may consist of multiple packets. Supervisory frames are used to carry system control data that also may span multiple frames. Unnumbered frames stand alone, and may carry either user data or control information. (Note: the FIS-B application will utilize unnumbered frames exclusively.) The 3-bit send and receive sequence numbers in the AVLC protocol provide for a flow-control window of up to 7 frames in each direction. The "poll/final" bit is cleared to zero when a response to the frame is not required. Setting the P/F bit to one indicates that a response is required by the protocol.

Bit Numbers

	1	2	3	4	5	6	7	8
Information	0	Send Sequence Number			P/F	Receive Sequence Number		
Supervisory	1	0	Supervisory Function Bits		P/F	Receive Sequence Number		
Unnumbered	1	1	Unnumbered Function Bits		P/F	Unnumbered Function Bits		

Figure 3-4 AVLC Link Control Field.

3.5 VDL-2 Equipment Architecture

Figure 3-5 below illustrates the typical architecture components of a VDL-2 installation. The "VHF Data Radio" (VDR) component provides the system physical layer functionality. A typical VDR in a commercial installation might support analog voice and ACARS as well as VDL-2. The hardware and software requirements for the VDR component are given in Reference 14. The physical input/output interface of the VDR is typically an ARINC 429 high-speed bus. The Williamsburg Version 3 binary file-transfer protocol operates over the ARINC 429 link-layer. A collection of software functions called the "VDL Management Entity" (VME) serves to interface the VDR with the "outside world" applications. The VME component contains a "Link Management Entity" (LME) for each current VDL-2 connection. The VME also includes the protocols necessary to manage the AVLC frames as described in sections 3.2-3.4 above. The ISO 8208 interface (described further in section 3.6 below) provides the ATN connection to network routers. The VME functionality is implemented in a hardware/software component called the "Communications Management Unit" (CMU). The CMU in a commercial installation might handle ACARS and SATCOM messaging as well as VDL-2. The hardware and software requirements for the CMU component are given in Reference 13.

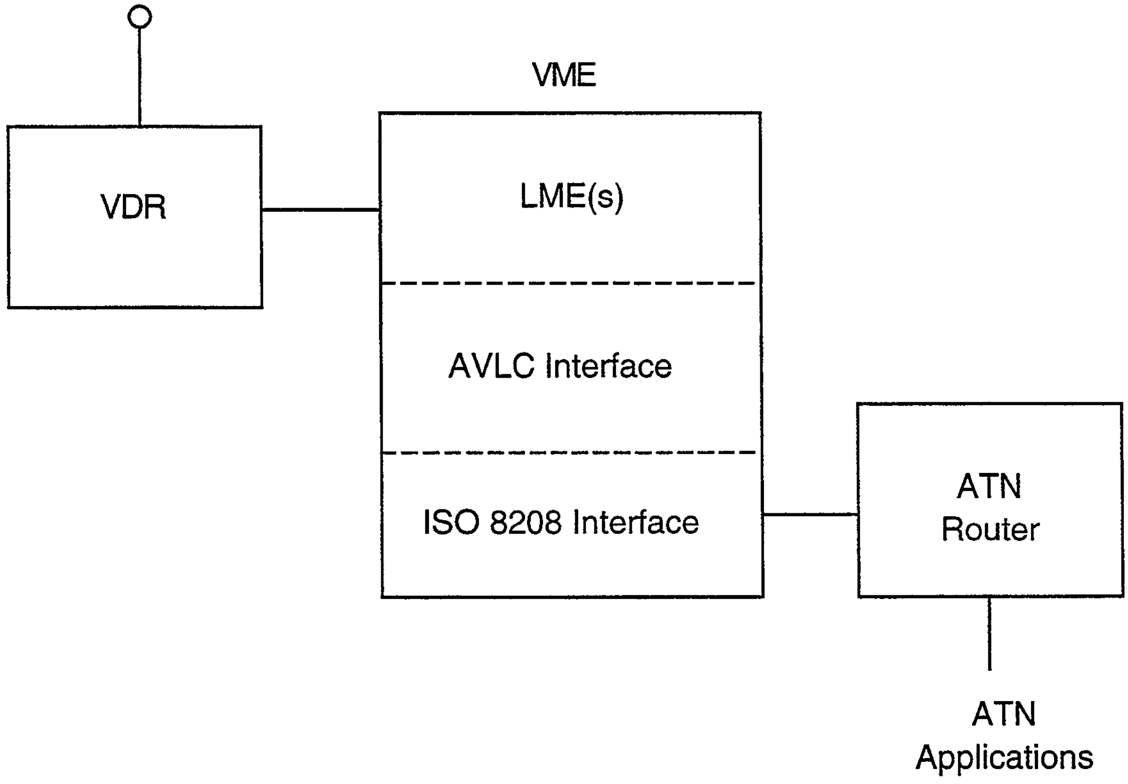


Figure 3-5 VDL-2 System Architecture.

3.6 ISO 8208 Interface

Each ATN subnetwork (including VDL-2) utilizes the ISO 8208 standard protocol (also known as X.25) as the system interface to the upper layers of the communications stack. As illustrated in Figure 3-6 below, the ISO 8208 packet format starts with a 4-bit format identifier that indicates the format of the rest of the packet header. Following the format identifier is a 4-bit group number and an 8-bit logical channel number. Together, these fields form a 12-bit identifier that specifies the particular network connection. (Note: ISO 8208 is a strictly connection-oriented protocol.) Next comes an 8-bit set of packet specification fields whose format depends on the particular packet type. (Further detail is unnecessary to the VDL-2/IP design here.) There are four types of ISO 8208 packets: (1) CALL REQUEST, (2) CALL ACCEPT, (3) CLEAR CONFIRMATION, and (4) DATA. A key point in the design of the VDL-2/IP interface is that the initial 4-bit format identifier field of an ISO 8208 packet can never be all ones (1111 binary) or have 3 initial ones (111x binary). (Reference 16 describes the ISO 8208 / X.25 packet formats in more detail.)

General Format Identifier	Group Number	Logical Channel Number	Packet Data Fields	Additional Data
4 bits	4 bits	8 bits	8 bits	

Figure 3-6 ISO 8208 Packet Format.

The ISO 8208 standard defines two types of network “entities”: “Data Terminating Equipment” (DTE), and “Data Communications Equipment” (DCE). VDL-2 aircraft systems are DTEs (since they are end-points of a message), while VDL-2 ground stations are DCEs (since they act as routers to relay the message on to its end application).

The ISO 8208 interface in VDL-2 provides a number of useful functions to the communications system (see Reference 16 for details of the ISO 8208 protocols). One such function is packet retransmission. This function is used to request the re-sending of ISO 8208 packets that were declared “invalid” at the receiver. This function is implemented in a non-standard way in VDL-2 so that both the DTE and DCE can request retransmission. The ISO 8208 interface provides a mechanism for the negotiation of system parameters (including the default packet size and flow-control window size) for each DCE-DTE connection. To speed up the connection process, the ISO 8208 interface provides a mechanism to “piggyback” up to 128 bytes of user data on the CALL REQUEST and CALL ACCEPT packets. The ISO 8208 interface also provides the “Called Address Line Modification Notification” function. This allows a DCE to indicate that it received a packet with a called address different from the one expected. A DTE can also use this facility when more than one address applies to the same DTE/DCE connection.

3.7 VDL-2 CMU Functions

The “Communications Management Unit” (CMU) component of the VDL-2 architecture is the hardware which hosts the software entities labeled VME in Figure 3-5 above. The primary function of the CMU is to create and maintain the air/ground VDL-2 link connections. The CMU is able to tune the VDR to selected frequencies for each link and to perform handoffs from one ground station to another as the aircraft moves about. Either the VDL-2 aircraft or ground system can initiate a handoff. Handoffs can be caused by changes in VDL-2 coverage (ex. loss of signal strength), or by policy (ex. selection of a particular airport). VDL-2 handoffs are “make-before-break”, with a typical handoff time of 20 seconds for the initiating LME and 60 seconds for the responding LME.

The VDL-2 architecture supports the airborne CMU functions by providing a default VDL-2 frequency called the “Common Signaling Channel” (CSC) that is available throughout the VDL-2 system coverage area. (The CSC is assigned to 136.975 MHz.) All

VDL-2 ground stations broadcast a special message called the "Ground Station Information Frame" (GSIF) every 90 seconds on the CSC. The GSIF contains the operating parameters of the ground station, including its operating frequencies, nearest airport, ATN network addresses supported, timer values, etc. Aircraft listen to the GSIF broadcasts in order to determine how to make a connection to the VDL-2 ground station. The GSIF provides a ground-based, distributed system database that can be updated (ex. to deal with new, failed, or moved VDL-2 ground-stations) without requiring any avionics changes.

The CMU component also hosts the VDL-2 outside system interfaces. The connection to the ATN is the 8208 interface (described in section 3.6 above). The secondary interface is at the AVLC level (described in section 3.2 above). The AVLC interface will be used to form the VDL-2/IP outside system connection.

4. ACARS Over AVLC (AOA)

ARINC and the airlines face a transitional problem as they seek to convert the ACARS network over into the faster and more-capable VDL-2. There will need to be some significant period of time (perhaps 15-20 years) during which both ACARS and VDL-2 systems must coexist. Aircraft equipped with VDL-2 radios will need to have both ACARS and VDL-2 functionality. VHF ground stations will need to continue the support of both ACARS and VDL-2 data links. Besides the subnetwork compatibility issues, "legacy" ACARS applications will need to have support over the VDL-2 subnetwork. In order to address this problem, ARINC has developed a means to operate ACARS applications over the VDL-2 subnetwork in a way that is totally transparent to the ATN applications that will eventually be supported over VDL-2. The "ACARS Over AVLC" (AOA) protocol is fully defined in reference 4. This section will describe the basic operation of the AOA protocol. Note that the AOA protocol forms the basis of the VDL-2/IP system that is the end goal of this paper (see section 7). A similar technique to AOA is used to deal with the FIS-B system (see section 5).

The basis of the AOA protocol is message "encapsulation" as illustrated in Figure 4-1 below. Following down the right-hand "stack" of the figure (assuming an airborne ATN message is being sent to the ground), the ATN message is first enclosed in a "Connectionless Network Protocol" (CLNP) datagram. (The details of CLNP are not relevant to this discussion.) The CLNP datagram is then enclosed in an ISO 8208 packet (as described in section 3.6 above). The ISO 8208 packet is further enclosed in an AVLC frame (as described in section 3.2 above). The process of stripping the various "encapsulations" is repeated in reverse sequence on the ground until only the ATN message remains. The left-hand "stack" in the figure illustrates an ACARS message following the same path as the ATN message. The ACARS message is encapsulated in an AOA packet which is then further encapsulated in an AVLC frame. The "trick" to the AOA protocol is how to differentiate AVLC frames containing AOA packets from AVLC frames containing ISO 8208 (ATN) packets.

The way that the AOA protocol differentiates the contents of AVLC frames is illustrated in Figure 4-2 below. Following ISO 9577 (reference 17), the first byte of the AVLC frame information field is defined to be the "Initial Protocol Identifier" (IPI) that defines the protocol of the data that follows. ISO 9577 defines IPI encodings for ISO 8208, as well as other ISO protocols. An IPI of all ones (0xff) is defined by ISO 9577 for extensions to the protocol set. Note that this value of IPI cannot be confused with the start of an ISO 8208 packet, since 1111 binary is not a valid ISO 8208 "General Protocol Identifier" (see section 3.6 above). Following the all-ones IPI, ISO 9577 specifies a second byte for the "Extended Protocol Identifier" (EPI). The AOA protocol assumes an all-ones EPI to identify an AOA packet (i.e. encapsulated ACARS message) within the AVLC frame. Hence, the AOA protocol entails a 2-byte additional overhead (the IPI and EPI bytes) to provide transparent interoperability of ATN and ACARS over the AVLC interface of VDL-2. A simple "Data Link Processor" (DLP) at the VDL-2 AVLC interface serves to direct ACARS-containing frames to their proper end-system. AVLC frames containing ISO 8208 datagrams are allowed to go on to their specific interface unchanged.

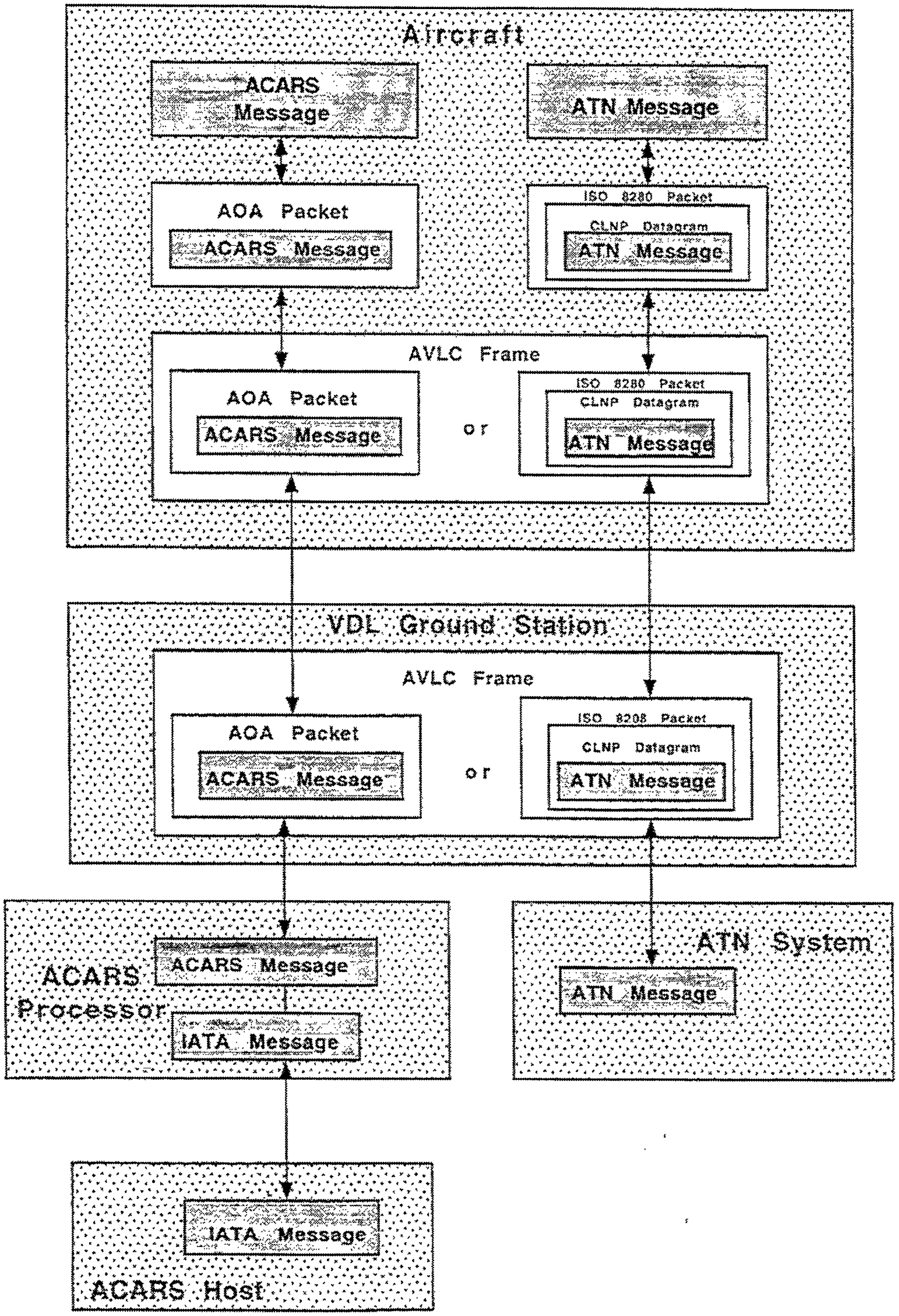


Figure 4-1 Structure of ATN and AOA Message Encapsulation.

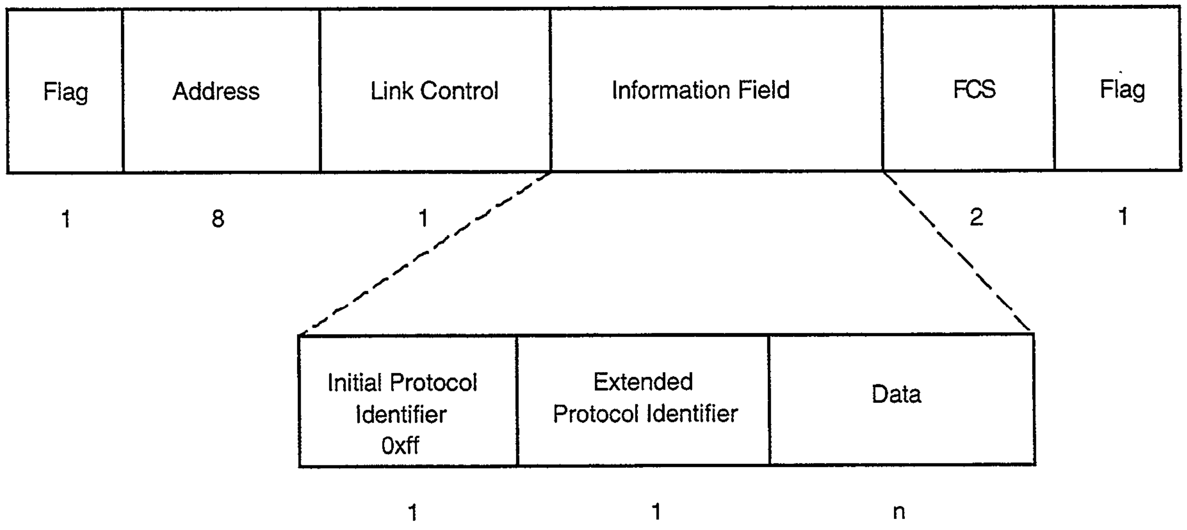

Flag
Address
Link Control
Information Field
FCS
Flag
1
8
1
2
1
Initial Protocol Identifier 0xff
Extended Protocol Identifier
Data
1
1
n

Figure 4-2 AOA Extended AVLC Frame Structure.

Note that the AOA protocol EPI value of all-ones is not currently recognized in ISO 9577 (reference 17). In fact, ISO 9577 says nothing at all about EPI encodings! IPI values from 11000000 though 11111111 binary are currently listed as "reserved". ARINC will need to get a recognized EPI encoding (possibly selected from the "reserved" IPI set) to complete the specification of the AOA protocol. Note that Appendix C of ISO 9577 sets aside several IPI encoding values for network-level interoperability with Internet applications. The encoding 11001100 binary specifies IPv4 (the current Internet version of IP). The encoding 10001100 binary specifies IPv6 (the newer Internet version of IP).

Figure 4-3 below describes the architecture components of a VDL-2 system supporting both AOA and ATN functionality over the VDL-2 subnetwork. The shaded portions of the figure would be omitted if ATN capability was not desired. The AOA "DLP" functions as a very simple router, directing AVLC frames containing ACARS messages to the ACARS router. Note that this architecture is an extension of Figure 3-5 above.

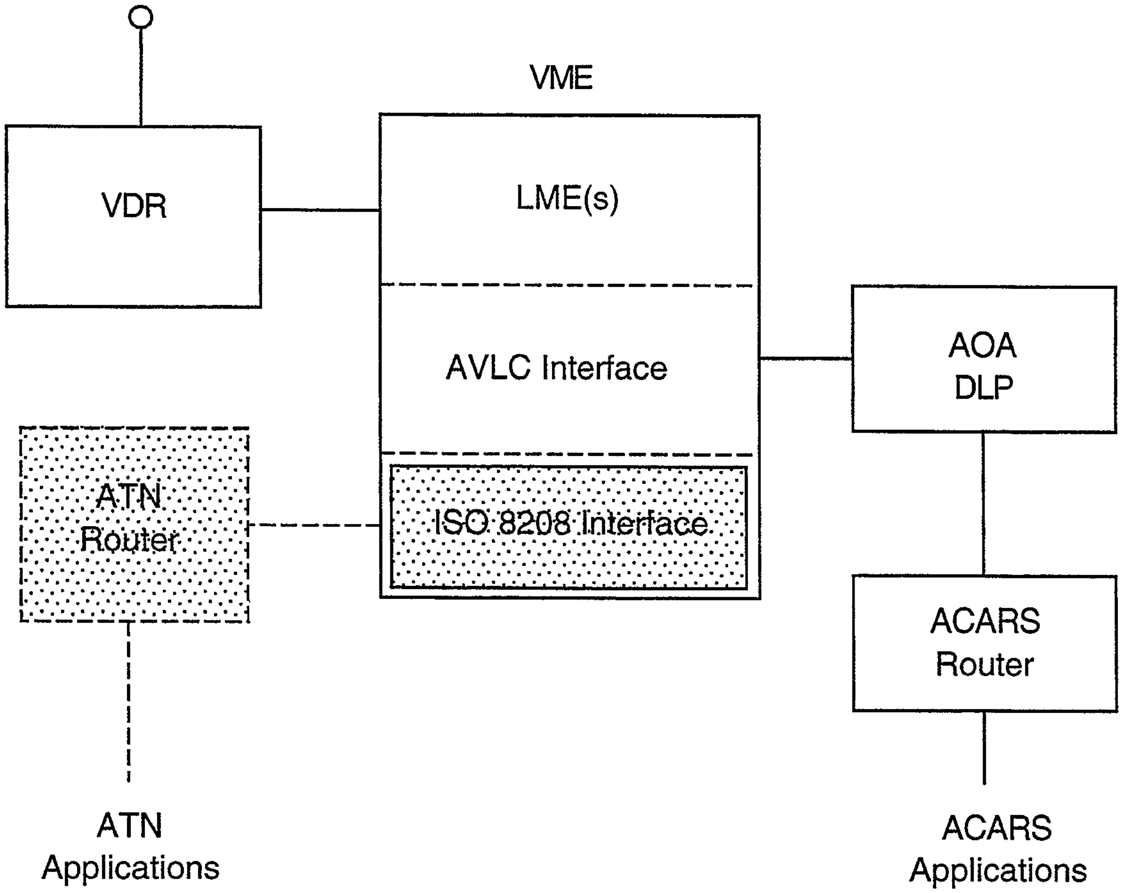


Figure 4-3 Avionics Architecture for AOA/ATN over VDL-2.

5. FIS-B Over AVLC

The FIS-B system (defined in Reference 7) is designed to operate using the AVLC frame interface of VDL-2. (Note: FIS-B may utilize other subnetworks as well as VDL-2.) FIS-B implementations may need to interoperate with ATN implementations (using the ISO 8208 interface) as well as with AOA implementations (using the AVLC interface). As was the case for AOA (see section 4 above), FIS-B frames must be distinguished from all other users of the AVLC link-layer of VDL-2. How this may be done is described below.

All FIS-B frames are of the type "unnumbered information" (i.e. uplink broadcast in this case). Hence, each FIS-B frame has a common value for its link-control field (11000000 binary). (Section 3.4 above describes the link-control field coding for AVLC.) The FIS-B destination address in the AVLC frame is coded for "all aircraft" as described in section 3.3 above. The A/G bit is set to one, since all FIS-B transmitters are on the ground. The C/R bit is cleared to zero, since all FIS-B messages are "commands" requiring no response. (No response is possible from FIS-B avionics, as they are receive-only.)

The FIS-B protocol differentiates its frames from other applications using the AVLC interface (AOA, ATN, etc.) in the same manner as AOA (see Figure 4-2 above). Following ISO 9577 (reference 17), the first byte of the AVLC frame information field is defined to be the "Initial Protocol Identifier" (IPI) that defines the protocol of the data that follows. An IPI of all ones (0xff) is defined by ISO 9577 for extensions to the protocol set. Note that this value of IPI cannot be confused with the start of an ISO 8208 packet, since 1111 binary is not a valid ISO 8208 "General Protocol Identifier" (see section 3.6 above). Following the all-ones IPI, ISO 9577 specifies a second byte for the "Extended Protocol Identifier" (EPI). The FIS-B protocol assumes an EPI value of 11111110 binary to identify an FIS-B packet within the AVLC frame. Hence, the FIS-B protocol entails a 2-byte additional overhead (the IPI and EPI bytes) to provide transparent interoperability of ATN, FIS-B, and ACARS over the AVLC interface of VDL-2. A simple "Data Link Processor" (DLP) at the VDL-2 AVLC interface serves to direct FIS-B frames to their appropriate end-system. AVLC frames containing ISO 8208 datagrams are allowed to go on to their specific interface.

Figure 5-1 below illustrates the architecture components of avionics to support FIS-B over VDL-2. A simple "Data Link Processor" (DLP) software entity performs the routing function that separates FIS-B messages in their AVLC frames from the data stream and directs them to their end-system. Note that this avionics may be configured to support the ATN or not, as desired. The shaded portions of the architecture are omitted if ATN functionality is not required. Note that Figure 5-1 is equivalent to Figure 4-3 (the avionics to support AOA). A common avionics system could support both AOA and FIS-B applications (as well as the ATN) over VDL-2 simply by combining the functions of the AOA and FIS-B DLP into a single software entity.

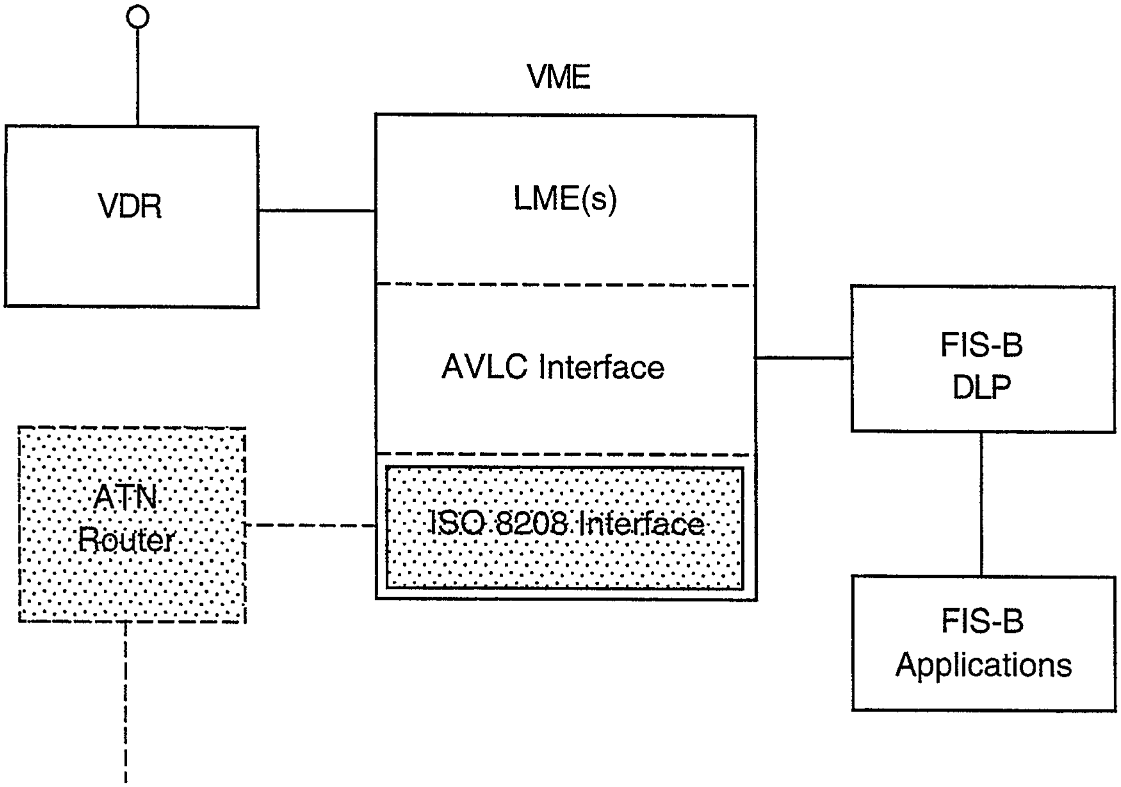


Figure 5-1 Avionics Architecture For FIS-B/ATN over VDL-2.

6. Support for the Address Resolution Protocol (ARP)

The remaining problem to be dealt with in the design of the VDL-2/IP system link layer is the fact that IP addresses only make sense to the TCP/IP protocol suite. A data link such as VDL-2 has its own, independent addressing scheme (see section 3.3 above) to which all links on the subnetwork must conform. These subnetwork-specific addresses may be entirely different from IP addresses in size and structure. It may be very difficult (or impossible) to directly convert from IP addressing to the subnetwork-specific addressing, due to constraints in addressing range, etc. In fact, the subnetwork may need to support multiple different upper-level protocols simultaneously – each requiring its own addressing. In the case of the VDL-2/IP system, the VDL-2 subnetwork must support both the ATN and IP addressing protocols seamlessly.

This problem of subnetwork link-layer address translation was solved early on in the design of the Internet. Internet hosts were (and still are) often connected via an Ethernet. Ethernet addresses are 48-bits long, with a unique structure (as compared to 32-bit IP version 4 addresses, or 128-bit IP version 6 addresses). When an Ethernet frame is sent from one host to another on the local network, it is the 48-bit Ethernet address that determines the frame's destination. The "device drivers" (called DLPs in this paper) within the TCP/IP stack of each host never look at the IP addresses in the Ethernet frames. The "address resolution" mechanism described here provides the mapping between the host addressing (Ethernet, VDL-2) and IP. The Internet "Address Resolution Protocol" (ARP) and its inverse "Reverse Address Resolution Protocol" (RARP) are further defined in reference 2. ARP provides a dynamic mapping from an IP address to its corresponding subnetwork ("hardware") address. RARP maps back from the subnetwork address to the IP address.

6.1 ARP Message Format

The ARP and RARP protocols utilize a common message format that is illustrated in Figure 6-1 below. Note that these messages are sent in subnetwork frames, not in IP packets.

ARP/RARP Message

Network Type	Protocol Type	Network Size	Protocol Size	Operation Code	Sender Network Address	Sender IP Address	Target Network Address	Target IP Address
2	2	1	1	2	m	n	m	n

Figure 6-1 ARP/RARP Message Format.

The first field in the message is a 2-byte "network type" identifier that specifies the particular subnetwork. This field is set to 1 for Ethernet, while the value 17 specifies an HDLC subnetwork. (See reference 19 for the current set of IETF-defined subnetwork identifier codes.) A coding for the VDL-2 subnetwork will need to be defined. The "protocol type" field (also 2 bytes) specifies the type of protocol being mapped by this ARP/RARP message. Its value is specified as 0x0800 for IP addressing. (Note that this value is deliberately chosen to be the same as the type field for an Ethernet frame containing an IP datagram.)

The next two 1-byte fields ("network size" and "protocol size") specify the number of bytes required for the subnetwork address and IP address. IP version 4 addresses require 4 bytes, while IP version 6 addresses require 16 bytes. The VDL-2 subnetwork addresses will be specified as 4 bytes long, although VDL-2 addresses have only 27 actual bits (see section 3.3 above). The fixed ISO 3309 low-order address-extension bits in the VDL-2 address will be suppressed here, along with the A/G and C/R indicators. The 27 remaining bits of the VDL-2 address will be packed down and the high-order 5 bits of the address cleared to zero to form a 32-bit VDL-2 subnetwork address.

The 2-byte "operation code" field specifies whether this message is an ARP request (code 1), an ARP reply (code 2), an RARP request (code 3), or an RARP reply (code 4). This field is required to distinguish ARP from RARP, as well as distinguishing requests from replies.

The next four fields in the message are the sender's subnetwork address (a VDL-2 address), the sender's IP "protocol address" (IP address), the target's subnetwork address (again, a VDL-2 address), and the target's address (again, an IP address). Note that there is some duplication of information here – the sender's subnetwork address is already in the subnetwork frame header that encapsulates the ARP/RARP message.

For an ARP request, all the fields in the message are filled except for the target's IP address (since that's what the message is requesting). When a system receives an ARP request, it fills in the target IP address, swaps the two sender addresses with the two target addresses, sets the operation code value to 2, and returns the ARP reply. RARP works in a similar manner.

6.2 ARP Protocol

An example of the application of the ARP protocol is illustrated in Figure 6-2 below. An Internet application desires to make a TCP connection with another Internet system over a subnetwork (such as VDL-2). The application typically accesses the destination system via its "Internet name". The first step in the process (1) is to obtain the IP address corresponding to the name. This is done via a call to an Internet "Name Resolver" function that acts as a directory. The IP address of the name resolver is "well-known" by all Internet applications (like the 411 number for local telephone directory assistance, or 911 for emergencies).

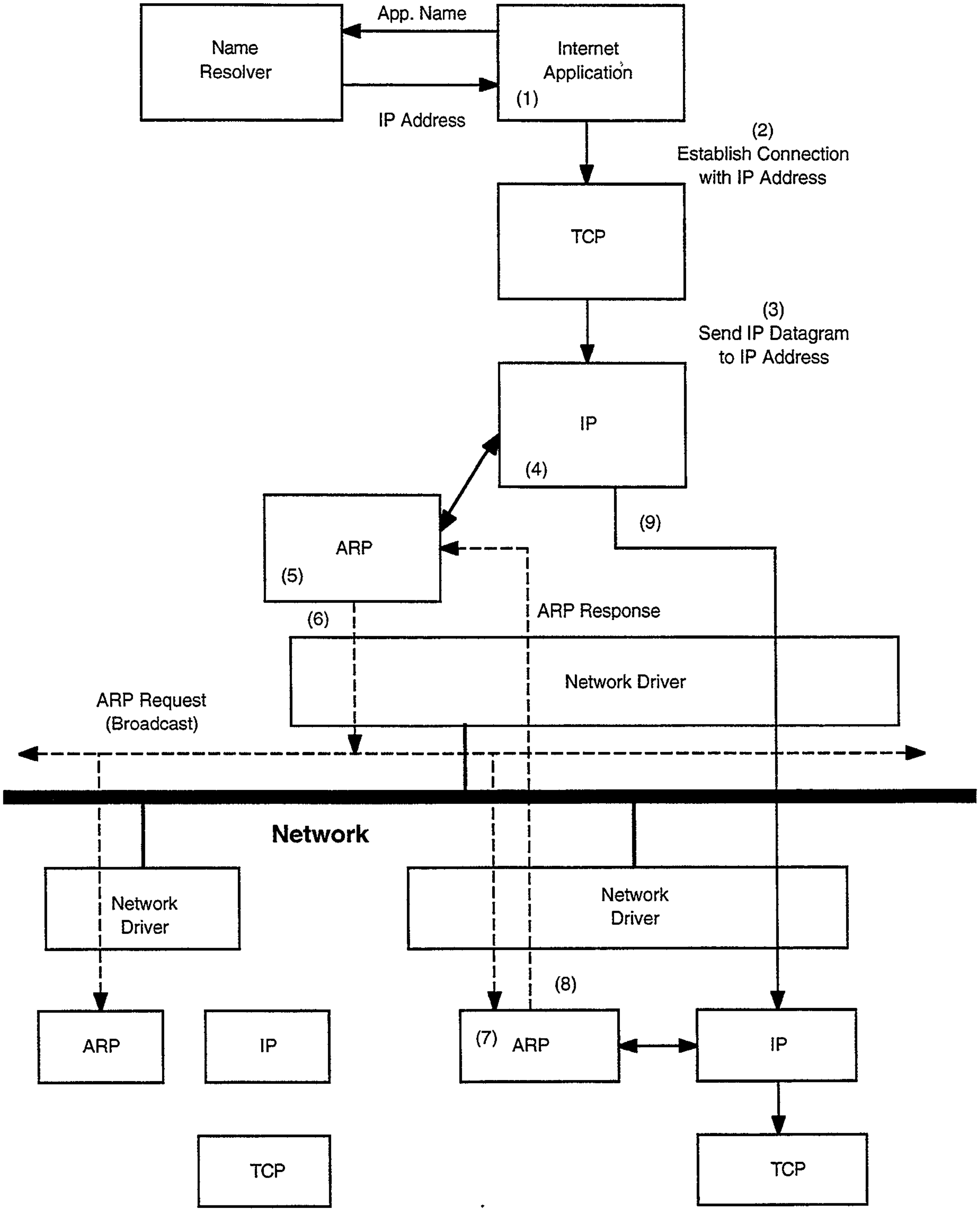


Figure 6-2 Address Resolution Protocol (ARP).

Once the IP address is available, the application attempts to establish the TCP connection through its protocol stack (2). The initial IP datagram is sent to the destination IP address (3). If the destination were on a locally-attached network, the IP datagram would be routed directly to that destination. If the destination is on a remote network, the IP routing function would determine the Internet address of a locally-attached router to forward the IP datagram. In either case, the IP datagram is sent to a host or router on the locally-attached network (4).

Assuming that the subnetwork is VDL-2/IP, the sender must convert the IP address into the corresponding VDL-2 address. The ARP function (5) provides for this translation of addresses. The ARP function maintains a cache of recently-accessed address mappings. The typical expiration time of an entry in the ARP cache is 20 minutes. If the required address translation is already in the ARP cache, then the local ARP function simply returns the requested IP address from its cache. Assuming that the requested address is not in the ARP cache (ex. this is the first time that the IP address is being referenced), then the ARP function broadcasts an ARP request (6) to every host system on the attached subnetwork (via a subnetwork broadcast frame). (Note: the boxes in Figure 6-2 labeled "Network Driver" are actually the VDL-2 radios with their CMU's and DLP entities.) The ARP broadcast request contains the IP address of the desired destination – it states "if you are the owner of this IP address, please respond to me with your VDL-2 address".

Each host system on the attached subnetwork receives the broadcast ARP request. When the destination system's ARP function receives the ARP request, it recognizes that the request is asking for its IP address (7), so it responds to the sender with an ARP reply containing its VDL-2 address (8).

The ARP reply is received back at the sender system, where it causes the generation of an entry in the sender's ARP cache. The IP datagram that forced the ARP request-reply to be exchanged can now be sent (9).

Recall that ARP cache entries typically time out in 20 minutes. A partial entry (one waiting for an ARP reply to complete it) times out in 3 minutes. The entry timer is restarted each time the entry is used.

It must be noted here that the link-layer protocols for VDL-2/IP (using ARP/RARP as described in this section) only provide for a type of mobile connectivity termed "nomadicity". A mobile user (i.e. aircraft) can attach to the system (i.e. a particular VDL-2 ground station), do any desired communications (while remaining connected in place), then disconnect from the system and move to a new location where it may re-attach and begin a new set of communications. This type of system can support some Internet protocols (e.g., UDP) and those applications that do not require unbroken connections as the aircraft move. However, connection-oriented Internet protocols (e.g., TCP) will require more than "nomadicity". The application connections must stay in place even during movement from one link-layer connection to another. The Mobile Routing protocols described in section 8 below are used to provide these network-layer mobile connectivity functions.

7. VDL-2/IP System Design

All the necessary pieces are now in place to define the VDL-2/IP system. The starting point for the VDL-2/IP "Data Link Processor" (DLP) is a standard Internet TCP/IP protocol stack/router implementation. In this section, the term "DLP" will refer to those design elements common to both the ground and aircraft VDL-2/IP systems. Where the implementation detail is specific to the VDL-2/IP ground station, the term GDLP will be used. Similarly, details specific to the VDL-2/IP avionics system will be denoted ADLP. In general, the VDL-2/IP design can be used with either the current IPv4 or the newer IPv6 Internet systems. Where the choice of IP version makes a difference, this section will discuss the appropriate selection of parameters.

Note that theVDL-2/IP DLP implementation might be either an Internet router or an end-system. (The Internet design makes little distinction between hosts and routers.) The ADLP would typically be an end-system (assuming that all the IP applications on the aircraft share a common host or local network). The GDLP would typically be a router, since the connection would likely connect via the ground Internet to remote applications hosts.

Figure 7-1 below illustrates the architecture components of the VDL-2/IP system. The similarities to the AOA/ATN system (Figure 4-3) and the FIS-B system (Figure 5-1) are clear. In fact, since VDL-2/IP, AOA, and FIS-B applications share the AVLC interface, a single DLP serves to support all three applications. The optional AOA and FIS-B components of the VDL-2/IP architecture are indicated as cross-hatched boxes in Figure 7-1 below. The ATN-specific components are indicated as dot-filled boxes.

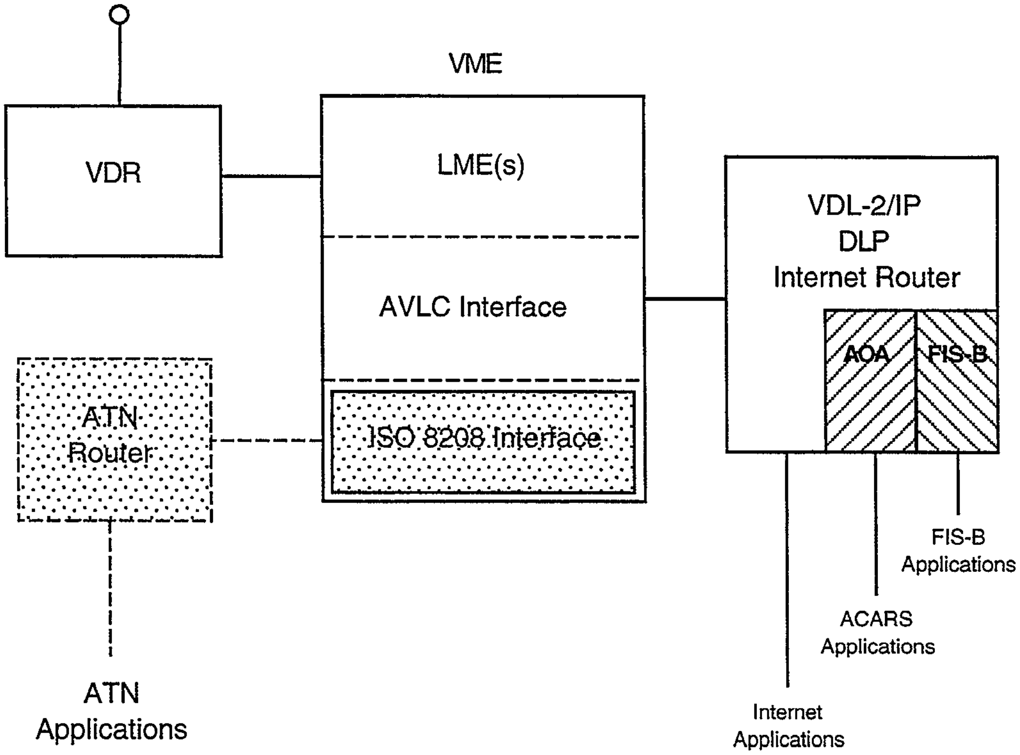


Figure 7-1 Avionics Architecture For VDL-2/IP.

7.1 Setting the IP MTU

The Internet IP protocol requires definition of the "Maximum Transfer Unit" (MTU). This is the length (in bytes) of the largest IP packet that is to be treated as a single unit. Standard IP segmentation and re-assembly protocols are used to deal with IP packets larger than the MTU at a given host or router (the VDL-2/IP DLP in this case). Recall from section 3.1 that the default maximum length of a VDL-2 frame is 8,312 bits – 1,039 bytes. The overhead of an AVLC frame (see section 3.2) is 13 bytes (2 flags, 8 address, 1 Link Control, and 2 FCS). Figure 4-2 illustrates the additional overhead of the AOA frame structure (IPI + EPI). Note: since the encapsulated packets for VDL-2/IP are either IPv4 or IPv6 (i.e. having ISO 9577 standard IPI encodings – see Reference 17), only the IPI byte will be required. Hence, the VDL-2/IP frame has 14 total bytes of overhead, leaving 1,025 bytes for data. The VDL-2/IP system MTU is set less than or equal to 1,024 bytes.

Note that by setting the VDL-2/IP MTU such that all IP packets traversing the VDL-2/IP interface fit entirely into a single AVLC frame, there is no need for the VDL-2/IP system to use the AVLC protocol's frame numbering/sequencing functions. All VDL-2/IP AVLC frames can be "Unnumbered Information" frames (similar to the FIS-B application described in section 5). Any requirement to segment/re-assemble data is done in the standard IP protocol processing of the DLP. This design choice significantly simplifies the VDL-2/IP AVLC interface. The drawback to this design choice is that frame-level error-detection and retransmission is now being done above the VDL-2 link layer, entailing somewhat longer delays for acknowledgements and an increased potential for lost data packets. For typical FIS datalink applications such as aviation weather dissemination (where the requirements for update rate are not critical), the choice of such a simple interface design for VDL-2/IP appears to be warranted. A second option for the VDL-2/IP system design is to utilize the AVLC "Numbered Information" frame type (see section 3.4 above) with a constant default value of 000 binary for the "send" sequence number. This option enables the operation of the VDL-2 link layer error-detection and retransmission algorithms, while dealing with exactly one outstanding VDL-2/IP frame at a time. This second option for VDL-2/IP is little more complex than the first (using unnumbered information frames), but it provides for a more-robust link. There will be shorter overall delays when dealing with lost frames.

Note, also, that the VDL-2/IP system does not make use of the ability of VDL-2 to negotiate the size of AVLC frames. This negotiation function is normally implemented as a part of the VDL-2 ISO 8208 interface (see section 3.6). Since it is assumed that at least some of the VDL-2/IP implementations will not require ATN support (inexpensive GA systems), the ISO 8208 interface components of the VME (see Figure 3-5) may be omitted. VDL-2/IP is designed to be completely independent from the ISO 8208/ATN components of the system. VDL-2/IP assumes that all VDL-2 implementations will support at least the default standard length of 8,312 bits. (Note: the standard upper-layer Internet protocols (TCP and UDP) provide a mechanism for applications to dynamically determine the MTU of a particular path, although this should not be required for VDL-2/IP.)

7.2 Configuring the VDL-2/IP AVLC Interface

The "local network" of VDL-2/IP utilizes the AVLC protocol for its link-layer. Hence, the VDL-2/IP DLP must be configured to encapsulate every Internet packet inside an AVLC frame for transmission across the AVLC interface. Standard Internet implementations usually have support for several "local" (link layer) protocols (see Reference 2). Among these available standard link-layer protocols are:

(a) IEEE 802 (Ethernet)
(b) Serial Line IP (SLIP)
(c) Point-to-Point Protocol (PPP)

PPP is quite similar to HDLC (and, hence, to AVLC). PPP utilizes the same flags, FCS, and bit-stuffing functions as do HDLC and AVLC (see section 3.2 above). VDL-2/IP DLPs will modify the Internet-standard PPP interface to obtain an AVLC interface.

The address field in standard PPP is simply a byte of all-ones – the conversion to AVLC addressing (8 bytes in ISO 3309 format – see Figure 3-3) is described in section 7.3 below. PPP incorporates a 2-byte "protocol identifier" field that has the fixed value 0x0021 hexadecimal for IP datagrams. This fixed identifier in PPP is replaced with the 1-byte AVLC Link Control field as described in section 7.4 below for VDL-2/IP.

7.3 VDL-2/IP AVLC Address Mapping

As illustrated in Figure 3-3 in section 3.3 above, the VDL-2 AVLC address structure actually has 27 encoded bits for each address (source and destination) packed into a 32-bit field. The VDL-2/IP DLP is responsible for reformatting VDL-2/IP addresses to and from the AVLC format. The VDL-2/IP DLP inserts and removes the 5 extra "special bits" as described in this section.

Note (from Figure 3-3) that the low-order bit of each AVLC address byte is reserved for the ISO 3309 "extensible address" protocol flag. For the destination address field, the flag bits are all cleared to zero. For the source address field, the first 3 flag bits are cleared to zero and the last flag bit is set to one, indicating the last byte of AVLC frame addressing. The VDL-2/IP DLP will simply insert or remove these fixed-value bits.

The second bit transmitted in the AVLC frame destination address is reserved for the A/G indicator flag (see Figure 3-3). A VDL-2/IP GDLP sets this bit to one, indicating that the source of the AVLC frame is "on the ground". A VDL-2/IP ADLP would clear this bit to zero by default. If the ADLP has access to the aircraft's "squat switch" (potentially via the VDL-2 CMU), then the A/G bit should be set to one if the switch indicates that the aircraft is not currently "airborne".

The second bit transmitted in the AVLC frame source address is reserved for the C/R indicator flag (see Figure 3-3). A VDL-2/IP DLP will clear this bit to zero, indicating that the AVLC frame is a "command". (Note: the VDL-2 LME managing the particular link may reset this bit in the AVLC frame as required by the VDL-2 link management protocols. If the LME changes bits in the AVLC frame, it will re-compute the FCS accordingly.)

The VDL-2/IP DLP must also perform address mapping for cases of broadcast addressing. As shown in Table 3-2 (section 3.3 above), VDL-2 broadcast addresses incorporate special encoding for various classes of address space. A VDL-2/IP GDLP would map all VDL-2/IP broadcast addresses to use type code 001 binary (since the frame's destination is "all aircraft"). The remaining 24 bits would be set to all ones. The VDL-2/IP ADLP, however, can select from 3 possible cases for its broadcast destination:

(1) All ICAO VDL-2 ground stations
(2) All ground stations of a particular VDL-2 service provider
(3) All VDL-2 ground stations {(1) and (2)}

The simplest approach for the VDL-2/IP design would be to select case (3). The VDL-2/IP ADLP would map all VDL-2/IP broadcast addresses to use type code 101 binary and the remaining 24 bits would be set to all ones. The drawback to this approach is the extra processing required in those ground stations that are not "interested" in VDL-2/IP applications. It would also be straightforward for the VDL-2/IP design to select case (1), assuming that the ICAO-administered VDL-2 ground stations were the only ground stations connected to the VDL-2/IP infrastructure. For case (1), the addresses would use type code 100 binary. If VDL-2/IP was supported by a particular service provider only {case (2)}, then the type code for VDL-2/IP broadcast addresses would be set to 101 binary. The service provider's "identifier code" would be inserted into the high-order bits of the "Specific Address Field" (see Table 3-2), while the remaining bits would be set to all ones.

7.4 VDL-2/IP AVLC Link Control Field

As was described in section 7.1 above, all VDL-2/IP frames will fit in a unique AVLC frame. Hence, all VDL-2/IP frames may use the "Unnumbered Information" encoding for the AVLC Link Control Field (as described in section 3.4 above). The UI frame encoding (as specified in Reference 10) is 1100 P000 binary. The P/F bit functionality of the AVLC protocol is not used in this mode of the VDL-2/IP design – the P/F bit is always cleared to zero in these VDL-2/IP frames. (This is true since all VDL-2/IP messages are complete in a single AVLC frame.) A special encoding of the 5 function bits in the Link Control field specific to VDL-2/IP may need to be assigned. Note: the low-order bit of each byte is transmitted first in the AVLC protocols. (VDL-2 also employs two other types of AVLC unnumbered frames – "exchange identity" (XID) = 1111 P101 binary, and "test" = 1100 P001 binary.)

As was described in section 7.1 above, the VDL-2/IP system may optionally select to use "Numbered Information" frame encoding for the AVLC Link Control Field instead of "Unnumbered Information" frame encoding (see section 3.4 above). The AVLC send/receive sequence numbers in the Link Control Field are defaulted to zero in this option, allowing only one outstanding AVLC frame at a time in the link-layer protocol. The encoding for the Link Control Field in the case of VDL-2/IP "numbered information" frame option is 0000 P000 binary (where the P/F bit default is one for this mode). The benefit of using the "numbered information" option as opposed to the "unnumbered information" coding is the enabling of the VDL-2 link layer error-detection and retransmission algorithms. The VDL-2 link layer will now ensure reliable reception of each AVLC frame – rather than requiring the higher-level IP and TCP functions to detect and handle missing frames. The overall delays entailed in the handling of lost or erroneous frames will be reduced and the end-to-end system performance will be enhanced.

7.5 VDL-2/IP IPI and EPI Settings

As described in sections 4 and 5 above, the basis of the AOA, FIS-B, and VDL-2/IP designs is the encapsulation of alternative protocol messages inside AVLC frames. As shown in Figure 4-2 above, this encapsulation is indicated by the initial bytes (IPI and possibly EPI) of the AVLC "Information Field". No EPI byte is required for VDL-2/IP, since the ISO 9577 standard (see Reference 17) specifies unique IPI encodings for IPv4 and IPv6. The "Initial Protocol Identifier" (IPI) byte will be set to 11001100 binary for an IPv4 implementation of the VDL-2/IP system. The IPI byte is set to 10001100 binary for an IPv6 implementation of the VDL-2/IP system.

7.6 Modifying ARP for VDL-2/IP

The basic design of the Internet "Address Resolution Protocol" (ARP) is described in section 6 above. There will need to be a bit of modification and tailoring of the standard Internet ARP implementation to suit the VDL-2/IP DLP.

First of all, the ARP (or RARP) messages will need to be encapsulated in AVLC frames (see sections 7.2-7-5 above). A new Network Type encoding (see Figure 6-1) for VDL-2/IP will need to be determined. (Note: it is possible that the existing HDLC encoding might be used here.) The Protocol Type is the standard encoding for IP. As stated in section 6.1 above, the Network Size will be 4 bytes for IPv4 systems and 16 bytes for IPv6 systems. The remainder of the ARP is filled in the standard Internet way. As is noted in section 6.1 above, the coding of the "subnetwork" (i.e. VDL-2) addresses requires some format conversion. This conversion processing is described in section 7.3 above.

8. Mobile Routing

The link-layer protocols for VDL-2/IP described so far in this paper {utilizing an extended ARP (see section 6 and 7.6 above), etc.} provide for a type of mobile connectivity termed "nomadicity". A mobile user (i.e. aircraft) can attach to the system (i.e. a particular VDL-2 ground station), do any desired communications (while remaining connected in place), then disconnect from the system and move to a new location where it may re-attach and begin a new set of communications. This type of system can support some Internet protocols (e.g., UDP) and those applications that do not require unbroken connections as the aircraft move. However, connection-oriented Internet protocols (e.g., TCP) will require more than "nomadicity". The application connections must stay in place even during movement from one link-layer connection to another. The Mobile Routing protocols described in this section are used to provide these network-layer routing functions.

<u>**Note:**</u> the Mobile Routing protocols described in this section were developed and standardized for the existing version of the Internet – IPv4. (See Reference 20 for the full specification of IPv4 Mobile Routing.) A new version of the Internet protocols (IPv6) is currently under development. The differences between the IPv4 Mobile Routing standard and the proposed Mobile Routing protocols for IPv6 will be outlined in section 8.6 below.

The Mobile Routing network-layer protocols are built on top of the standard IP routing functions. Mobile Routing operates through extensions to the existing IP protocols that provide the additional functionality in those end-systems and routers that require it. (In the case of VDL-2/IP, these would be the ADLP s and GDLPs.) Other parts of the Internet are unaffected when specific systems choose to use Mobile Routing. Figure 8-1 below illustrates the placement of the two Mobile IP functions "Agent Discovery " (described in detail in section 8.2) and "Agent Registration" (described in detail in section 8.3) in relation to the standard Internet protocols. "Agent Discovery" is an extension to the Internet Control Message Protocol (ICMP) used by the system to generate and respond to system-events (errors, etc.). "Agent Registration" uses the User Datagram Protocol (UDP) to send and receive messages. It is noted that both the ICMP and UDP protocols are built upon the underlying IP protocol.

The Mobile Routing design incorporates a fundamental assumption that unicast IP packets – those destined to a single recipient – are routed without regard to their IP Source Address. Mobile Routing protocols are only concerned with the IP Destination Address. A second basic assumption is that the Internet can successfully route IP packets in some "magical" way. As stated above, Mobile Routing does not replace the existing Internet routing, it simply operates over it.

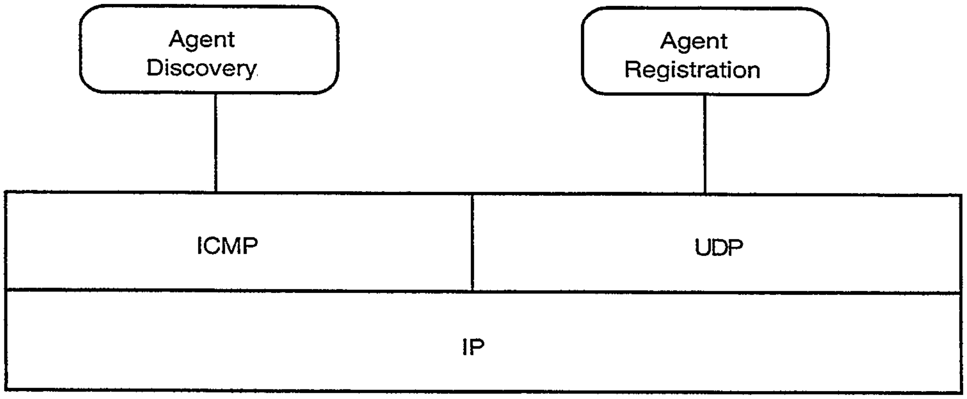


Figure 8-1 Mobile IP Extensions to Standard Protocols.

8.1 Mobile Routing Terms and Concepts

Mobile Routing in the Internet (termed Mobile IP) defines three functional entities where mobility protocols must be implemented. These are termed: (a) Mobile Node, (b) Home Agent, and (c) Foreign Agent. (In the case of VDL-2/IP, these entities will be incorporated into the ADLPs and GDLPs.) This section will provide basic definitions of these terms, and some additional terms and concepts that are necessary to build up the Mobile Routing protocols for VDL-2/IP. More-complete definitions of these terms and concepts will appear in later sections of this paper.

8.1.1 Mobile Node

A "mobile node" is an end-system or router that can change its point of attachment to the Internet from one link to another while maintaining any ongoing communications. (In the case of VDL-2/IP, a mobile node is an aircraft ADLP.) The mobile node is addressed only by its permanent IP home address.

8.1.2 Home Agent

A "home agent" is an IP router with an interface on the mobile node's home link (a VDL-2/IP GDLP) which can provide the following three functions:

(a) The home agent informs the mobile node of its current location as the mobile node moves from link to link. The "location" is represented by a "care-of" address.

(b) The home agent may advertise reachability to the mobile node's home address, thus attracting IP packets that are destined to the mobile node's home address. Alternatively, some other router may advertise reachability to the mobile node's home address.

(c) The home agent intercepts packets destined to the mobile node's home address and tunnels them to the mobile node's current location; i.e. to the care-of address.

In the case of VDL-2/IP, the basic idea is that all aircraft have a home router that supports their permanent home address. All aircraft belonging to "ABC Airlines" are reached via the ABC Airlines home agent. Since the home agent doesn't move, the messages destined for ABC Airline's aircraft can always be directed via ABC Airline's home agent router. An organization such as the FAA or AOPA would need to support home agent(s) for GA aircraft.

8.1.3 Foreign Agent

A "foreign agent" is an IP router with an interface on the mobile node's foreign link (a VDL-2/IP GDLP) which can provide the following three functions:

(a) The foreign agent assists the mobile node in informing its home agent of the mobile node's current care-of address.

(b) The foreign agent provides a care-of address and "de-tunnels" packets for the mobile node that have been "tunneled" by its home agent.

(c) The foreign agent serves as a default router for packets generated by the mobile node when connected to this foreign link.

Figure 8-2 below illustrates the relationship between the home and foreign agents, mobile nodes, etc. The large circle in the center represents the existing Internet. The Mobile Routing concept makes it transparent to Internet applications desiring to communicate with a mobile node whether the mobile node is currently attached to its home agent or to some foreign agent. The message packet is sent to the home agent, and the Mobile Routing protocols "somehow" re-route it to the foreign agent where the mobile node is currently connected. {In a sense, the ACARS system already works this way. All messages intended for aircraft are initially directed to the ACARS main processor (in Annapolis for ARINC) and they are then re-routed to the ACARS ground-station that currently has aircraft connectivity.}

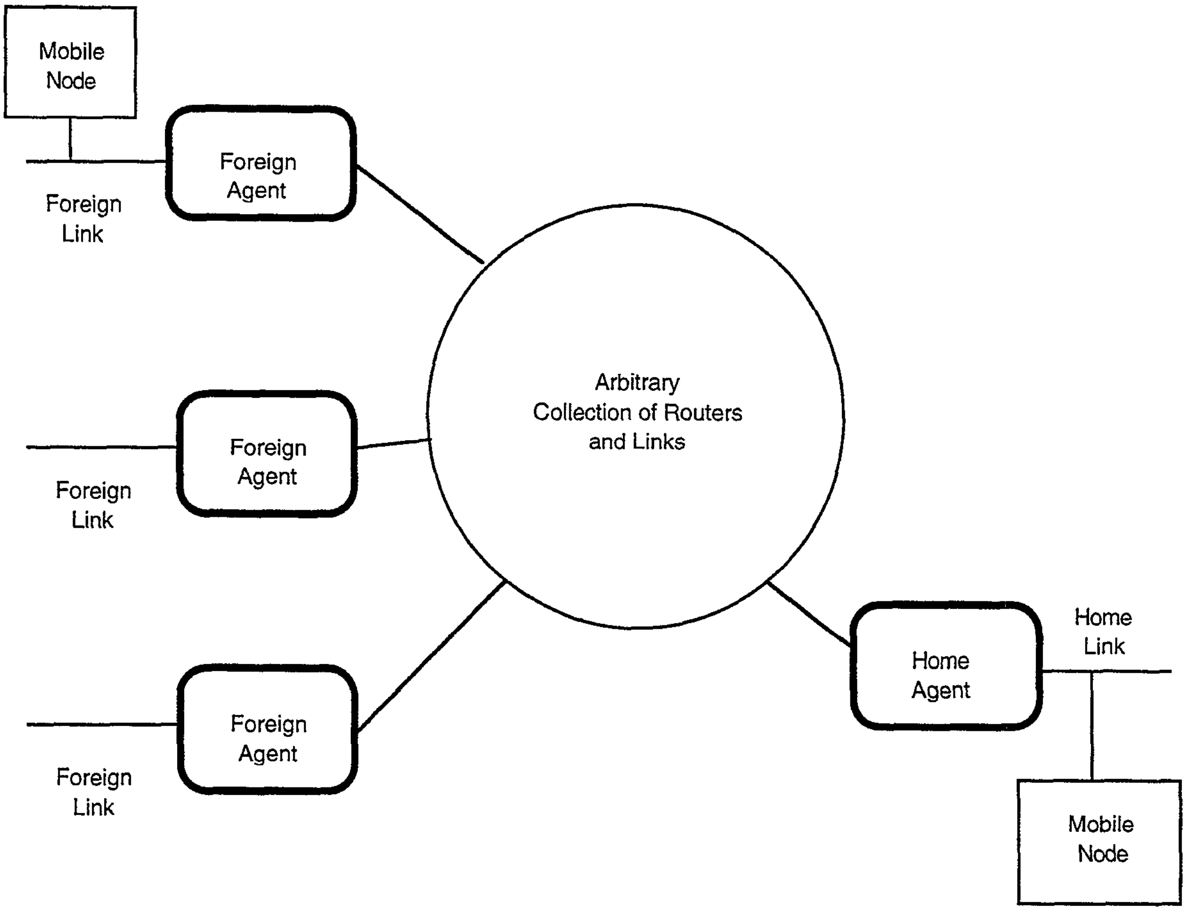


Figure 8-2 Mobile Routing Entities and Relationships.

8.1.4 Tunneling

The term "tunnel" is used to describe the path taken by an IP packet that is encapsulated within the payload portion of a second packet, as shown in Figure 8-3 below. (The concept of encapsulating one packet within another was already described in the discussion of AOA in section 4 above.) The figure illustrates an example of a home agent tunneling a packet to a foreign agent in order to deliver the packet to a mobile node. (The reverse process would be equivalent.) Further descriptions of the IP tunneling mechanism is given in section 8.5 below.

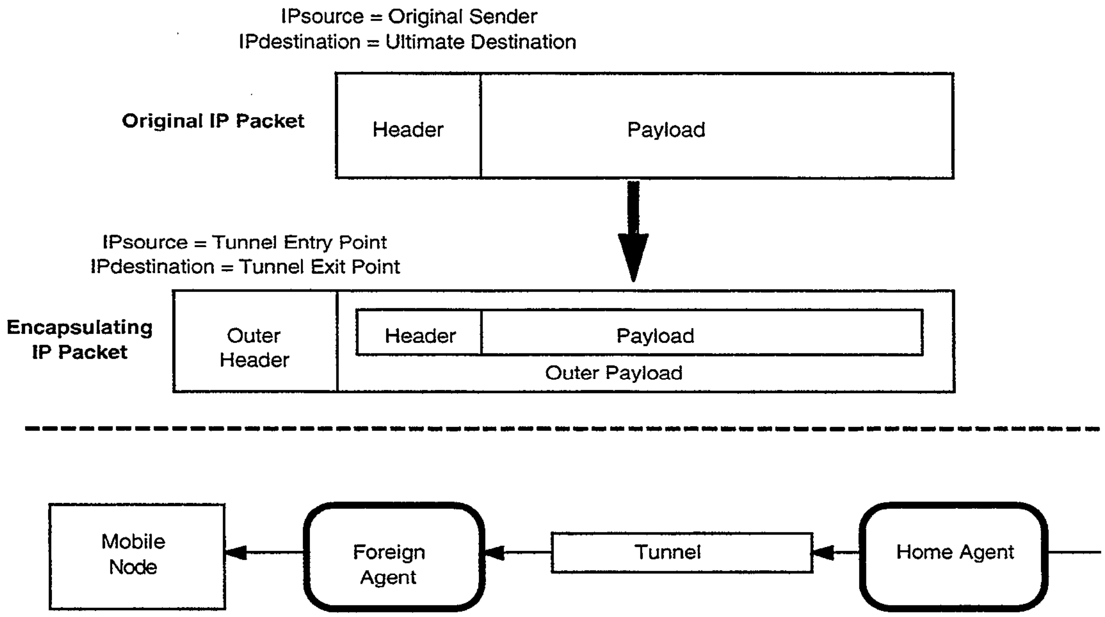


Figure 8-3 IP Tunneling.

8.1.5 Home Address

As was already stated, a mobile node's home address is an IP address permanently assigned to the node. This address is closely related to the mobile node's home agent, and therefore its home link. A mobile node's home agent is a router that has at least one interface on the mobile node's home link. A mobile node communicates with all other nodes using its home address.

Note that a mobile node's home link need not be a physical link on some real, physical medium. The home link can be a virtual link that exists only as software within the mobile node's home agent. The home agent can be considered to have a virtual interface through which it attaches to this virtual home link. Figure 8-4 below illustrates how such a virtual link is used to support the "tunneling" of IP packets. There are three links illustrated in the figure. Interface 1 might be an Ethernet. Interface 2 might be VDL-2/IP as described in this paper. The virtual interface performs tunnel encapsulation (or decapsulation, if the direction of the arrows were reversed). The virtual interface re-routes the packet through a second call to the software supporting standard IP routing. The figure shows how an IP packet would be encapsulated by a home agent in order to tunnel the packet to its foreign agent. Reversing the arrows would make the figure illustrate the decapsulation of the IP packet at the foreign agent.

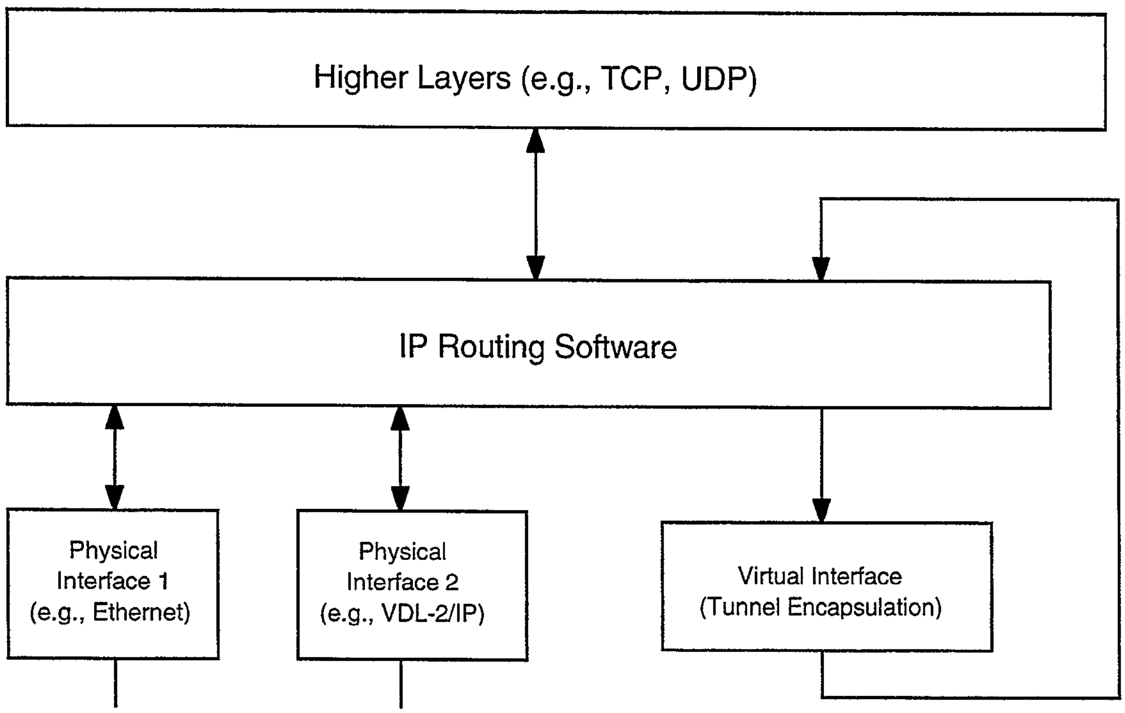

Higher Layers (e.g., TCP, UDP)
IP Routing Software
Physical Interface 1 (e.g., Ethernet)
Physical Interface 2 (e.g., VDL-2/IP)
Virtual Interface (Tunnel Encapsulation)

Figure 8-4 Encapsulation via a Virtual Interface in Home Agent.

8.1.6 Care-of Address

A “care-of” address is an IP address associated with a mobile node that is “visiting” a foreign link. A care-of address has the following properties:

(a) A care-of address is specific to the foreign link currently being visited by the mobile node.

(b) A mobile node’s care-of address generally changes each time that the mobile node moves from one foreign link to another.

(c) Packets destined to a care-of address can be delivered using existing Internet routing mechanisms; i.e. no Mobile IP-specific procedures are required to deliver packets to a care-of address.

(d) A care-of address is used as the exit-point of a tunnel from the home agent to the mobile node.

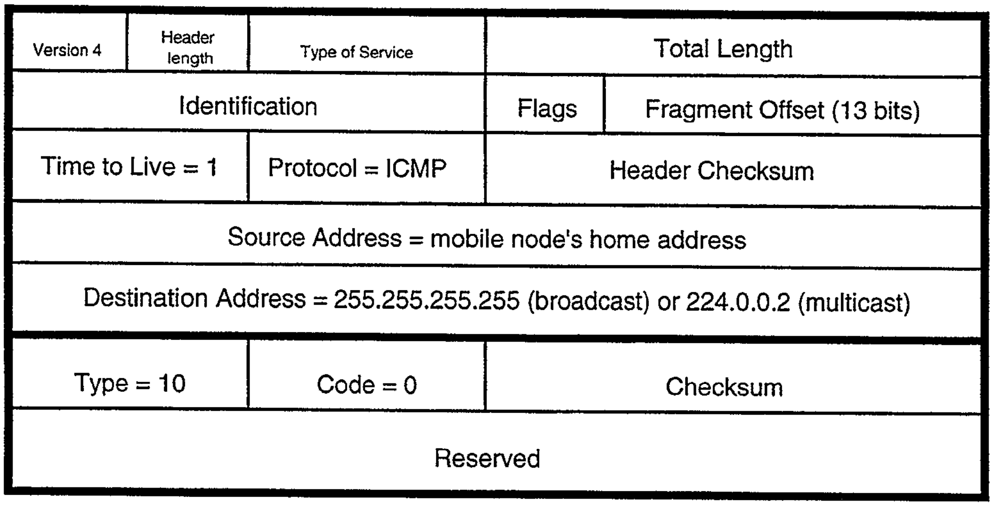
(e) A care-of address is almost never used as the IP source or destination address in a mobile node’s communications with other nodes.

A foreign care-of address is an IP address of a foreign agent which has an interface on the foreign link being visited by a mobile node. A foreign agent care-of address can be shared by many mobile nodes simultaneously. A collocated care-of address is an IP address temporarily assigned to the interface of the mobile node itself. This type of care-of address might be used by a mobile node in situations where no foreign agents are available on a foreign link. A collocated care-of address can be used by only a single mobile node at a time.

8.2 Agent Discovery

The Agent Discovery process is used by a mobile node to determine whether it is currently connected to its home link or to a foreign link. Agent Discovery is also used to detect when a mobile node has moved from one link to another. Finally, Agent Discovery is used to obtain a care-of address when the mobile node is connected to a foreign link.

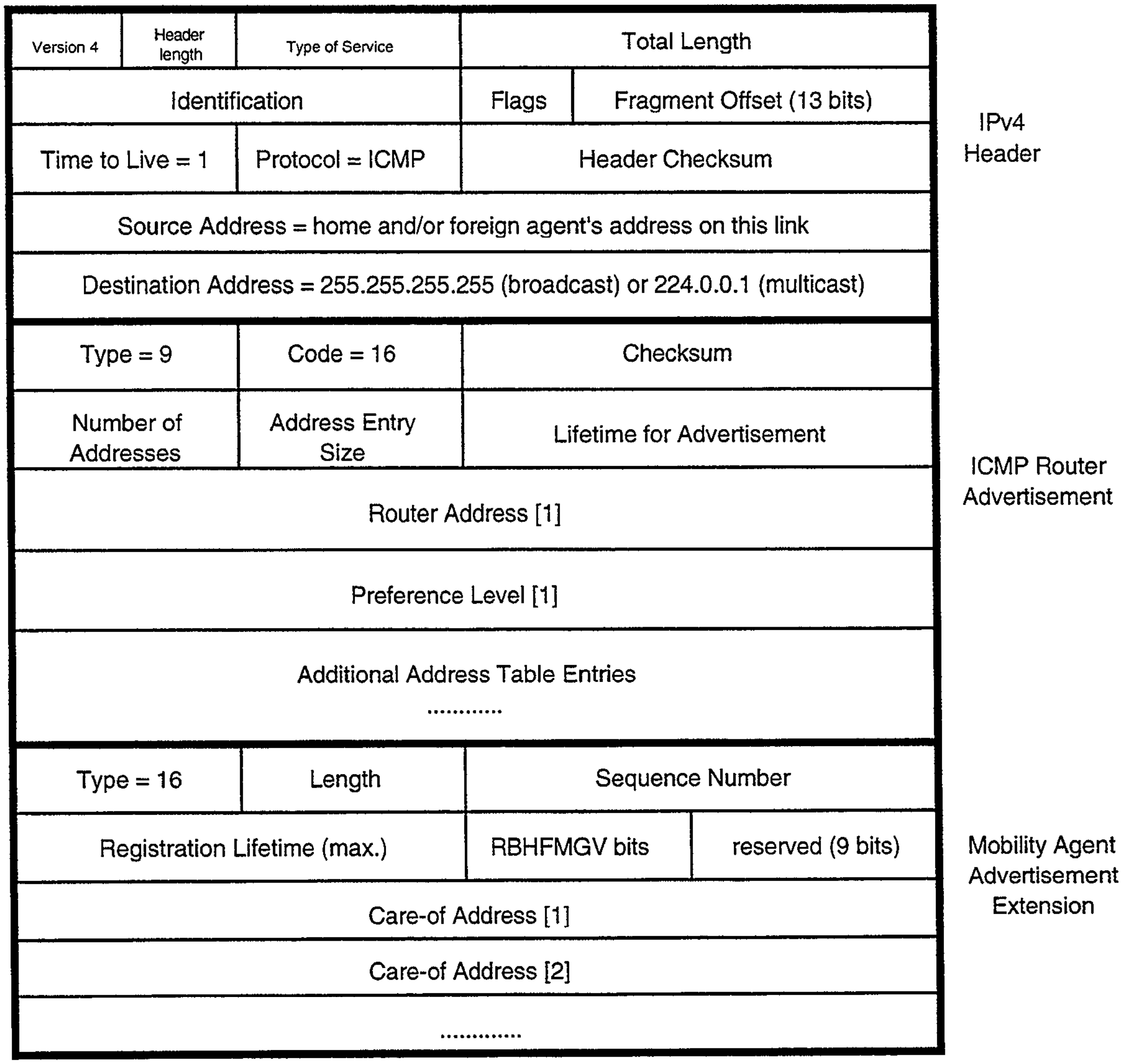
The Agent Discovery process uses two simple messages that are built as extensions to the standard Internet ICMP messages (see figure 8-1 above). The "agent advertisement" message (illustrated in Figure 8-6 below) is used by agents (home and foreign) to announce their capabilities to mobile nodes. These messages are periodically transmitted as broadcasts to each link on which a node is configured to perform as a home agent, foreign agent, or both. This allows a mobile node that is connected to such a link to determine whether there are any agents present, and if so, what their respective identities (IP addresses) and capabilities are. The broadcast rate is typically three times the expected "lifetime" of an "agent advertisement". If the mobile node does not wish to wait until the next "agent advertisement" broadcast, it can send an "agent solicitation" message instead. This message forces any agents on the link to immediately transmit their "agent advertisement" message. The "agent solicitation" message is illustrated in Figure 8-5 below. The "agent solicitation" message is almost identical to the standard ICMP "Router Solicitation" message (see reference 21). The time-to-live field of an "agent solicitation" message is set to 1.



Version 4	Header length	Type of Service	Total Length	
Identification			Flags	Fragment Offset (13 bits)
Time to Live = 1		Protocol = ICMP	Header Checksum	
Source Address = mobile node's home address				
Destination Address = 255.255.255.255 (broadcast) or 224.0.0.2 (multicast)				
Type = 10		Code = 0	Checksum	
Reserved				

Figure 8-5 Agent Solicitation Message.

The "Agent Advertisement" message is formed by appending one or more of the Mobile Routing extensions to the standard Internet ICMP "Router Advertisement" message (see reference 21). The "type" code of 9 identifies the message as an advertisement. The "code" field is used to distinguish the Mobile Routing advertisement from the normal one. (By convention, standard routers use a code=0, while agent routers use code=16.) The "lifetime" field indicates how often this agent broadcasts its advertisements. The "H" and "F" bits in the Mobility Agent Advertisement Extension indicate whether this agent can act as a home or foreign agent. The "B" bit is used by a foreign agent to indicate that it is too busy to accept any additional registrations (see section 8.3 below). The "R" bit indicates whether registration is required for this agent. The "V" bit indicates whether this agent supports the "Van Jacobson" compression algorithm to reduce the overhead involved with TCP and IP headers (see reference 22). There will be some additional discussion about header compression in section 8.5 below. The "M" and "G" bits refer to the selection of the IP tunneling encapsulation algorithm, also discussed in section 8.5 below. Some additional discussion on the contents of the "Agent Advertisement" message will be found in the discussion of Mobile Node registration in section 8.3 below.



<table>
<tr><td>Version 4</td><td>Header length</td><td>Type of Service</td><td colspan="2">Total Length</td><td rowspan="5">IPv4 Header</td></tr>
<tr><td colspan="3">Identification</td><td>Flags</td><td>Fragment Offset (13 bits)</td></tr>
<tr><td colspan="2">Time to Live = 1</td><td>Protocol = ICMP</td><td colspan="2">Header Checksum</td></tr>
<tr><td colspan="5">Source Address = home and/or foreign agent's address on this link</td></tr>
<tr><td colspan="5">Destination Address = 255.255.255.255 (broadcast) or 224.0.0.1 (multicast)</td></tr>
<tr><td colspan="2">Type = 9</td><td>Code = 16</td><td colspan="2">Checksum</td><td rowspan="5">ICMP Router Advertisement</td></tr>
<tr><td colspan="2">Number of Addresses</td><td>Address Entry Size</td><td colspan="2">Lifetime for Advertisement</td></tr>
<tr><td colspan="5">Router Address [1]</td></tr>
<tr><td colspan="5">Preference Level [1]</td></tr>
<tr><td colspan="5">Additional Address Table Entries
............</td></tr>
<tr><td colspan="2">Type = 16</td><td>Length</td><td colspan="2">Sequence Number</td><td rowspan="5">Mobility Agent Advertisement Extension</td></tr>
<tr><td colspan="3">Registration Lifetime (max.)</td><td>RBHFMGV bits</td><td>reserved (9 bits)</td></tr>
<tr><td colspan="5">Care-of Address [1]</td></tr>
<tr><td colspan="5">Care-of Address [2]</td></tr>
<tr><td colspan="5">.............</td></tr>
</table>

Figure 8-6 Agent Advertisement Message.

A mobile node typically determines that it has moved using the "lifetime" value from the "Agent Advertisement" message. This field tells the mobile node how soon it should hear another advertisement from the same agent. (In fact, advertisements are usually broadcast three times as fast to allow for lost messages.) If a mobile node is registered with a foreign agent, and it fails to hear an advertisement from that agent within the specified time, then the mobile node can assume that it has moved to a different link or that its foreign agent has failed. The mobile node will then re-register with the next foreign agent from which it receives an "Agent Advertisement". If no such advertisements are heard, the mobile node can send an "Agent Solicitation". If even the solicitations fail to produce a response, the mobile node can attempt to communicate as though it were connected to its home link. If there is no response from the default home router, then the mobile node can assume that it is connected to some foreign link. In this case, the mobile node can attempt to obtain an address from a "Dynamic Host Configuration Protocol" (DHCP) server (see reference 23 for details). If this request is successful, the mobile node can use this address as a collocated care-of address and register with its home agent. Failing this, the mobile node cannot communicate – it must wait for human intervention, or the receipt of new "Agent Advertisement" messages.

8.3 Mobile Node Registration

A mobile node "registers" whenever it detects that its point of attachment to the network has changed from one link to another. Since these registrations elapse after a specified time period, the mobile node reregisters when it has not moved but when its existing registration is due to expire. Mobile node registration provides for the following processes:

(1) The mobile node requests routing services from a foreign agent on a foreign link.
(2) The mobile node informs its home agent of its current care-of address.
(3) The mobile node renews is current registration when it is due to expire.
(4) The mobile node "de-registers" when it returns to its home link.

The registration process also allows for mobile nodes to have multiple care-of addresses registered with an agent to support multicasting. A given care-of address can be de-registered while retaining others. The mobile node can use the registration process to dynamically determine the address of a potential home agent, if the mobile node did not have prior knowledge of its home agent(s). Note that this section provides only a brief survey of the registration protocols and features (see reference 20 for the full description).

(Note: if the "Sequence Number" field values in its foreign agent's "Agent Advertisement" messages (see Figure 8-6) are out of sequence (indicating that the foreign agent may have rebooted), the mobile node will need to re-register.)

Mobile IP registration involves the exchange of two messages: a "Registration Request" and a "Registration Reply". As was shown in Figure 8-1 above, each of these messages is carried within the data portion of a UDP "datagram" which is, itself, the data in an IP packet. A "Registration Request" is sent by a mobile node to begin the registration process. The "Registration Request" may be sent directly to the mobile node's home agent, or it may be sent via a foreign agent. The home agent receives the "Registration Request" and responds with a "Registration Reply". If the mobile node does not receive a reply in response to its request (in a reasonable time period), the mobile node retransmits the request a number of times until a reply is received. The time period between requests increases up to a maximum time, when the mobile node is forced to give up.

Note that the end purpose of the mobile node registration process is to build and maintain a table in the home agent that maps a mobile node's home address into the mobile node's current care-of address(es). An entry in this table is termed a "binding entry". Binding Entries are only valid for the specified lifetime – a mobile node must re-register if the lifetime of its binding is about to expire. (In a sense, this table is similar to the ARP cache discussed in section 6 above.)

Figure 8-7 below illustrates a "Registration Request" message complete with its IP and UDP headers. Figure 8-8 below illustrates the "fixed length" portion of the "Registration Reply" message (since the rest of the message is identical in format to the "Registration Request" in Figure 8-7). (Reference 24 gives details of the UDP header, while reference 20 details the registration message formats and protocols.)

Note that the UDP destination port is set to the value 434 reserved for Mobile IP registration messages. The "S" bit in the fixed-length portion indicates whether the designated binding is to be created or deleted. The "M", "G", and "V" bits refer to IP Tunneling issues (also included in the "Agent Advertisement" message shown in Figure 8-6 above and discussed in section 8.2 above). The "B" and "D" bits refer to support for broadcast messages (discussed in more detail in section 8.4 below). The "R" bit indicates that the mobile node must register via a foreign agent – even if it is using a collocated care-of address. The "Lifetime" field in the "Registration Request" indicates the number of seconds that the binding entry is desired to be maintained. The "Lifetime" in the "Registration Reply" indicates the number of seconds that the binding entry will actually be maintained. A Lifetime of zero indicates that the mobile node wishes to de-register the care-of address, while a Lifetime of all ones indicates that the binding entry is to be permanent. The 64-bit "Identification" field is chosen by the mobile node to be unique for each attempted registration. It allows the reply message to be matched to its outstanding request. The Identification field (along with the Mobile-Home Authentication Extension) is also used in the registration authentication algorithm described later in this section. The "Code" field in the "Registration Reply" message indicates whether the request was accepted or rejected – and if rejected, the reason for the failure.

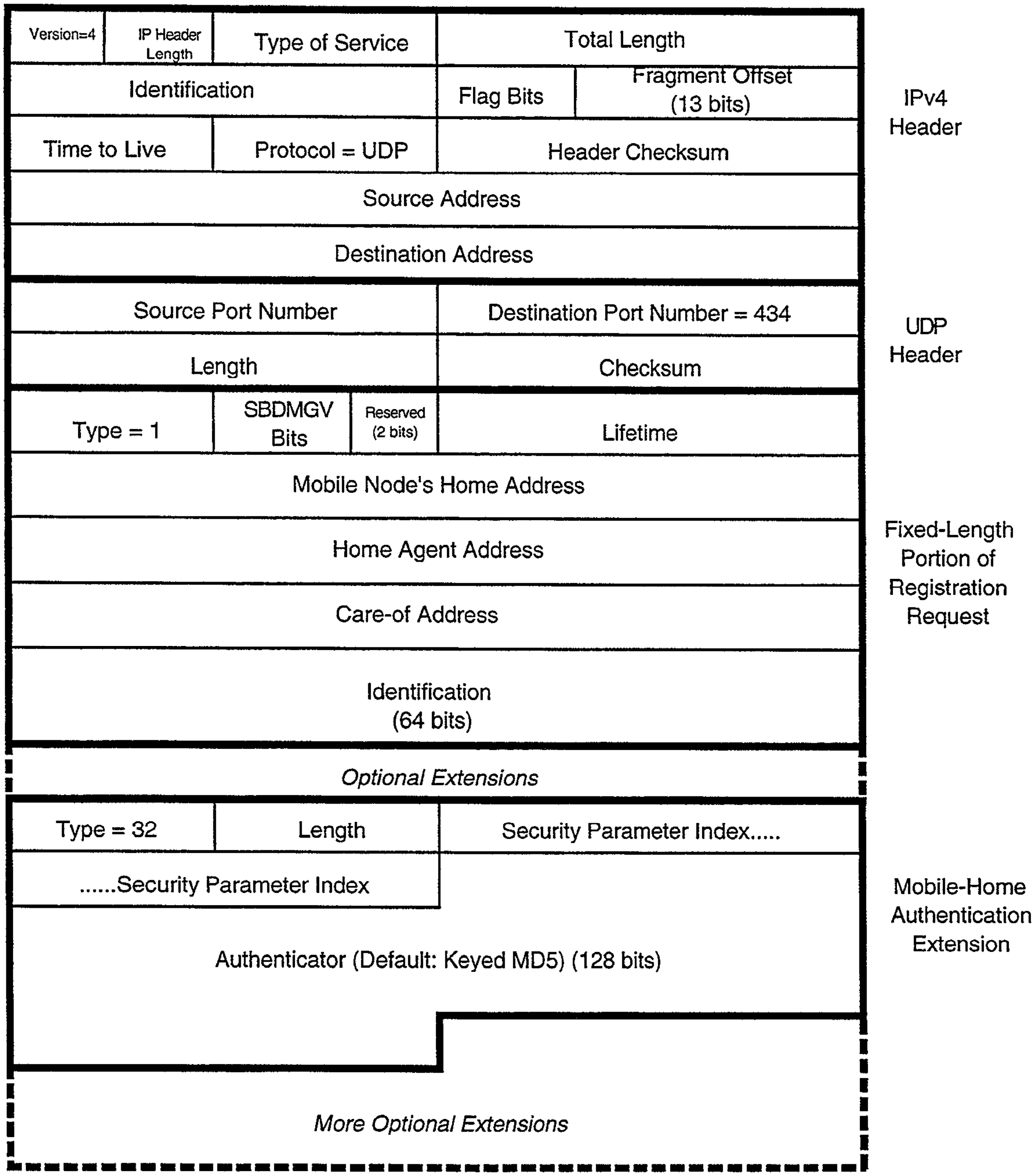


Figure 8-7 Registration Request Message.

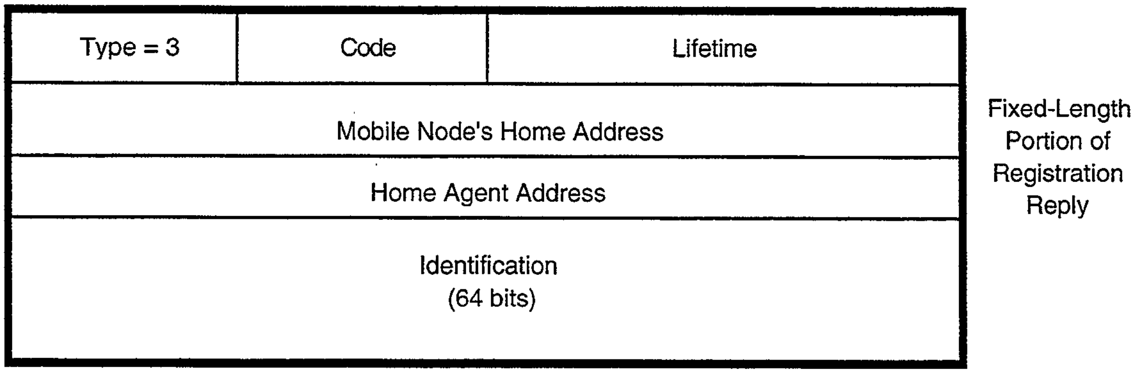


Figure 8-8 Registration Reply Message (Fixed Portion Only).

The Mobile IP registration procedure provides a significant vulnerability to the overall system. A "hacker" could send a false "Registration Request" to a mobile node's home agent and cause all packets to be tunneled to the hacker instead of the mobile node's legitimate care-of address. Besides giving the hacker access to the mobile node's data, this would also prevent the mobile node from receiving any data packets. To prevent this sort of "denial of service" attack, the Mobile IP registration protocols employ a strong authentication algorithm check to insure that registration replies are valid. The "Identification" field in the registration messages combined with the Mobile-Home Authentication Extension (see Figures 8-7 and 8-8 above) support this algorithm.

The basis of this authentication algorithm is the use of a secret key value known only to the mobile node and its home agent. In generating the "Registration Request", the mobile node uses its secret key to generate a 128-bit "message digest" over the entire request message. The "Keyed MD5" algorithm (see Reference 25) is used to generate this "message digest", which is placed in the "Authenticator" field of the "Registration Request" message. The mobile node's home agent repeats the MD5 processing to check that the "Registration Request" is genuine. The reverse process is used to authenticate the "Registration Reply".

Note: a "hacker" could replay a mobile node's valid "Registration Request" messages at later times, thus circumventing the authentication protocol. Mobile IP inserts a time-stamp value in the "Identification" field of the request and reply to prevent this sort of intrusion. The mobile node uses its current date/time for the request identification. If this time is not sufficiently close to the home agent's date/time (possibly indicating a replay attack), then the home agent rejects the registration. The rejection notice contains the home agent's date/time value – allowing the mobile node to resynchronize itself with the home agent.

8.4 Support for Mobile Broadcasting

IP broadcast packets are defined to have destination addresses with the host portion defaulted (set to all ones). A mobile node can obtain a copy of broadcasts that are transmitted on its home link, even when the mobile node is currently connected to a foreign link. The mobile node requests that its home agent forward to it all broadcast messages by setting the "B" bit in its registration requests (see section 8.3 above). This instructs the home agent to deliver a copy of all such broadcasts via a tunnel to the mobile node. The exact mechanism used is different depending on whether the care-of address is "collocated" or "foreign". The "D" bit in the registration request is used to determine which mechanism is to be used (see reference 20 for details.)

Transmission of IP broadcast messages from a mobile node is independent of whether it has a collocated or foreign care-of address. There are three cases to consider.

1. If the destination of the broadcast is link-specific and intended for the mobile node's foreign link, then the mobile node simply transmits the packet on the foreign link, using the link-layer broadcast address to deliver the frame containing this packet to all nodes on the link.

2. If the destination of the broadcast is link-specific and intended for the mobile node's home link, then the mobile node must tunnel this packet to its home agent. The encapsulating packet has an IP source address set to the mobile node's home address, and the IP destination address set to the mobile nodes' home agent.

3. If the destination is a prefix-specific broadcast (i.e. of the form "network-prefix.111...111"), then the mobile node has two alternatives. It can tunnel the packet to its home agent as in (2) above. Second, the mobile node can transmit the packet normally, sending it in a link-layer frame whose destination is that of its router – just like any unicast packet transmitted by the mobile node. However, some intermediate routers might be configured to filter out broadcast packets – the mobile node would be better off to tunnel such packets to its home agent.

In summary, mobile nodes can participate in IP broadcasts (either as senders or receivers) by appropriately setting the "B" and "D" bits in their registration requests. The procedures described above are used to appropriately route and forward broadcasts.

8.5 IP Tunneling Protocols

As was described in section 8.1.4 above, "tunneling" (the encapsulation of one IP packet inside another) is the basic mechanism behind Mobile IP routing. The simplest form of encapsulation was illustrated in Figure 8-3 above. However, as was mentioned in sections 8.2 and 8.3 above (the "M" and "G" bits), Mobile IP routing actually supports multiple encapsulation protocols. This section will describe these protocols and the tradeoffs involved in their usage. The areas of concern are:

- IP Header Compression
- Support for IP Fragmentation
- Relaying ICMP Messages ("Soft State")
- Preventing Recursive Encapsulation
- Support for Alternate Protocols

A standard IPv4 header contains 20 bytes. Hence, each time an IP packet is encapsulated it grows in length by 20 bytes. A significant amount of the extra header overhead is essentially redundant. Mobile IP provides a protocol that minimizes the header overhead – "Minimum Encapsulation" is described in section 8.5.2 below. However, there are drawbacks to this technique. It should be noted that the IP protocols also support an overall algorithm to compress IP (and TCP) headers called "Van Jacobson" compression (see Reference 22). "VJ" compression operates by forming a virtual-connection between end-systems (in contrast to the "datagram" connectionless operation of standard IP). "VJ" compressed packets have a connection identifier in their header instead of repeating all the IP addressing and other fields that are consistent from one message to the next. The "VJ" algorithm must send IP headers uncompressed periodically to keep the end-systems synchronized in case of errors or failure of an end-system.

A major component of the standard IPv4 header is support for packet fragmentation and de-fragmentation. However, not all the encapsulation protocols can support fragmentation of the encapsulated packet within an outer packet that may itself be fragmented.

The Internet Control Message Protocol (ICMP) (see reference 26) defines a set of messages which provide diagnostic and error messages within the system. ICMP messages generated by a host or router in response to an IP packet are sent to the original source of the packet and provide useful information to that source. However, ICMP messages generated in response to a tunneled packet are sent to the entry-point of the tunnel, not necessarily to the original source of the inner packet. The tunnel entry point must relay certain ICMP messages to their original source. The algorithm used to do this is termed "soft state" and is defined in Reference 26. "Soft State" refers to the fact that ICMP messages contain an ICMP header, the IP header of the "offending" packet, and the first 8 bytes of the packet payload. Unfortunately, the 8 bytes are insufficient to carry the source and destination addresses of the inner packet when the "offending" packet is encapsulated. "Soft State" is a set of variables maintained by a tunnel entry-point. The entry-point updates its "soft state" based on ICMP messages received from routers within the tunnel. The tunnel entry-point uses these "soft state" variables to generate ICMP messages for the original source of the packet at the time it encapsulates that packet for delivery into the tunnel. Support for the "Soft State" algorithms is part of the encapsulation algorithm.

As was shown in Figure 8-4 above, IP encapsulation occurs at a "virtual interface". There is a potential infinite-loop here, with the possibility that a given packet may increase in size and circulate through the network indefinitely. Some encapsulation protocols provide algorithms to prevent these loops.

While it is not required for the VDL-2/IP design, some applications of Mobile IP require support for other network protocols in addition to IP. A packet of one protocol might be encapsulated within the payload of a packet from a second protocol. One of the Mobile IP encapsulation algorithms provides for this function.

Figure 8-9 below illustrates the three alternative encapsulation algorithms provided in Mobile IP. A short description of each will be given in the following sections.

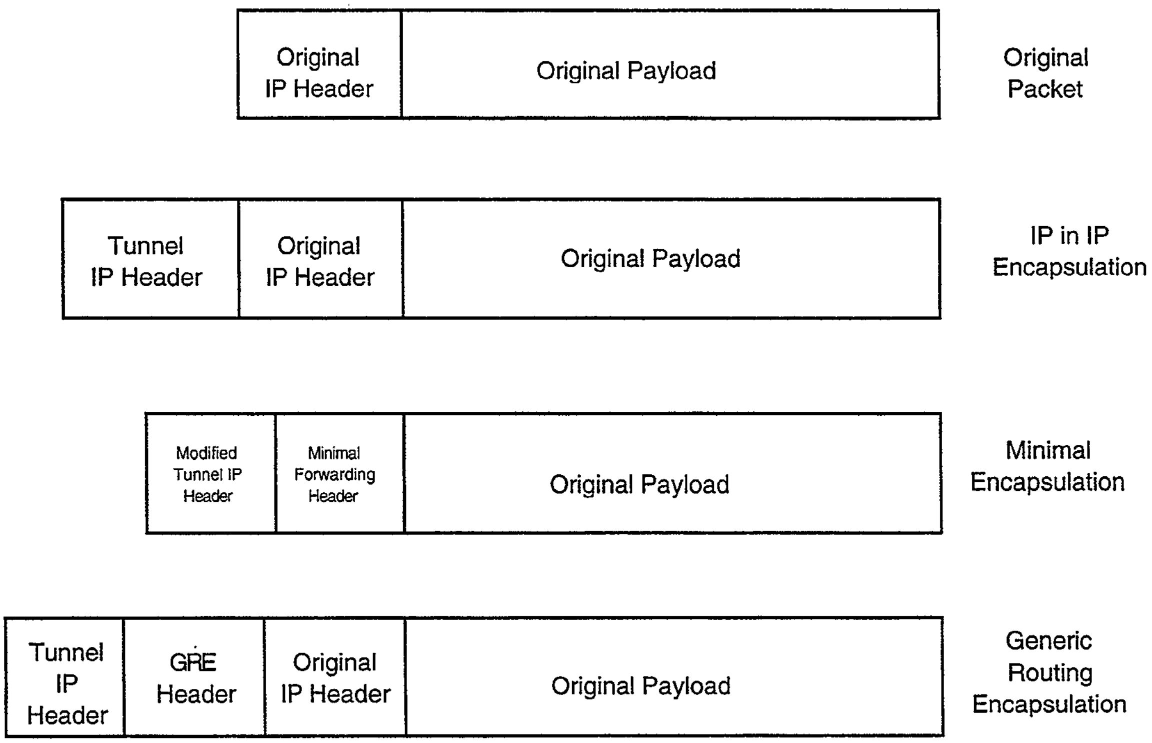


Figure 8-9 Mobile IP Encapsulation Formats.

8.5.1 IP in IP Encapsulation

"IP in IP Encapsulation" (see Reference 27) is the default Internet standard for IPv4 encapsulation. All home and foreign agents are required to implement this encapsulation protocol. IP in IP does not do any header compression (depending on the optional Van Jacobson compression applied later). IP in IP provides "soft state" support. IP in IP can handle fragmentation, but it does not provide recursive encapsulation prevention or support for alternate network protocols.

8.5.2 Minimal Encapsulation

"Minimal Encapsulation within IP" (see Reference 28) is an optional form of tunneling that may be used within Mobile IP (by setting the "M" bit in the agent advertisement and agent registration messages). Minimal Encapsulation strips away the redundant information carried in the encapsulating (outer) IP header and the encapsulated (inner) IP header (as shown in Figure 8-9 above). The Minimal Encapsulation header is illustrated in Figure 8-10 below. The encapsulation overhead is reduced from 20 bytes to 8 (or 12). However, in the process of compression, Minimal Encapsulation has stripped away the information necessary to handle the case where the original IP packet is already fragmented. Minimal Encapsulation must support "soft state", since its minimized header lacks the original source address. Minimal Encapsulation also may cause packets to be lost within a long tunnel, since the "Time to Live" field is stripped from the encapsulated packet. Minimal Encapsulation cannot support alternate network protocols. VDL-2/IP implementations will not use Minimal Encapsulation.

<table>
<tr><td>Protocol (8 bits)</td><td>S</td><td>reserved
(7 bits)</td><td>Header Checksum (16 bits)</td></tr>
<tr><td colspan="4">Original Destination Address (32 bits)</td></tr>
<tr><td colspan="4">Original Source Address (if S bit set) (32 bits)</td></tr>
</table>

Figure 8-10 Minimal Encapsulation Header Format

8.5.3 Generic Routing Encapsulation

"Generic Routing Encapsulation" (GRE) (see Reference 29) is a second optional form of tunneling that may be used within Mobile IP (by setting the "G" bit in the agent advertisement and agent registration messages). GRE is the only encapsulation algorithm that supports alternate network protocols. GRE also provides explicit protection against recursive encapsulation. GRE adds an additional header to the IP headers, thus actually expanding the encapsulation overhead by a minimum of 4 bytes. Figure 8-11 below illustrates the GRE extra header format. Note the 3-bit "Recur" field used to count down encapsulations, and the "Protocol" field used to indicate the encapsulated network protocol.

Figure 8-11 GRE Header Format.

<table>
<tr><td>CRKSs
bits</td><td>Recur
(3 bits)</td><td>Flags
(5 bits)</td><td>Version
(3 bits)</td><td>Protocol (16 bits)</td></tr>
<tr><td colspan="4">Optional Checksum
(16 bits)</td><td>Optional Offset
(16 bits)</td></tr>
<tr><td colspan="5">Optional Key (32 bits)</td></tr>
<tr><td colspan="5">Optional Sequence Number (32 bits)</td></tr>
<tr><td colspan="5">Optional Routing (32 bits)</td></tr>
</table>

8.6 IPv6 Mobile Routing

The current Internet implementation (IPv4) is gradually being superceded by a new design (IPv6). IPv6 is a significant re-design of the Internet protocols, bringing with it many improvements and extensions. This section will briefly touch on aspects of the IPv6 design that impact the operations of Mobile IP routing. A comparison of how IPv6 performs similar functions to those of Mobile IPv4 will be made. Note that IPv6 design work is on-going, and the standards are not yet complete. Reference 30 gives the current draft documentation of IPv6 Mobile Routing.

Figure 8-12 below illustrates the IPv6 header. Contrast this to the IPv4 header shown in Figures 8-5, 8-6, and 8-7. IPv6 omits a number of IPv4 fields, including the "type of service", header checksum and all fragmentation support (in the IP header). IPv6 has added the "priority" and "flow label" fields. The IPv6 "priority" field allows the sender to attach a relative priority to a given packet with respect to other packets sent by that same sender. The "flow label" allows IPv6 to attach identifiers to particular packets within the IP stream to form a sort of "virtual connection". This labeling allows special handling to be performed for a specific "named flow" to support such new IP applications as real-time audio/video. The "Hop Limit" serves a similar function to the IPv4 "Time to Live". The "Next Header" is used to form a linked list of "extension headers" that carry various IPv6 options (only when required). The list of potential IPv6 extension headers includes:

(1) Hop-by-Hop Options Header: contains options that must be examined by every router along the path from the original source to the ultimate destination.

(2) Destination Options Header: contains options that are examined only by the ultimate destination (and by intermediate destinations explicitly called out in a Routing Header.

(3) Routing Header: performs functions similar to the IPv4 Loose/Strict Source and Record Option.

(4) Fragment Header: used to send a packet larger than the path MTU (this was always part of the IPv4 header).

(5) IP Authentication Header: provides authentication, integrity checking, replay protection, etc. (well beyond the IPv4 header checksum).

(6) IP Encapsulating Security Payload Header: provides security and authentication for the entire packet's content.

(7) Upper-Layer Header: forms the transport or application-layer header (an IPv4 or IPv6 Header in the case of a tunnel.

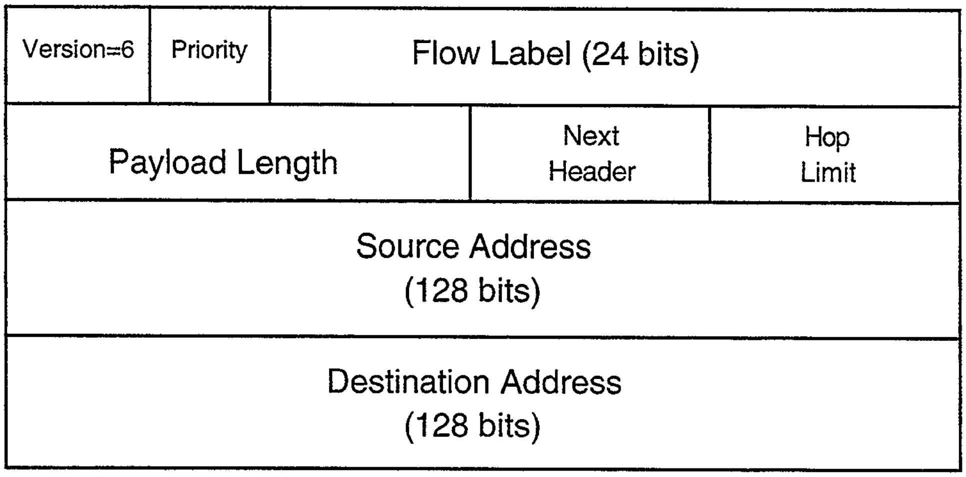


Figure 8-12 IPv6 Header Format.

The major change in the IPv6 header is the quadrupling of the Source and Destination Address lengths – from 32 bits to 128 bits. The address "squeeze" in the current Internet was one of the motivating factors in the design of IPv6. There is little chance of running out of Internet addresses during the lifetime of IPv6. More importantly, the long addresses allow aggregation of many network-prefix routes into a single network-prefix route (with a longer prefix). Finally, the longer addresses allow nodes to auto-configure using simple mechanisms. IPv6 has reserved special addresses to provide compatibility with IPv4 nodes and to embed the network-layer addresses of various other protocols. IPv6 also defines special "local use" addresses. There are three subclasses of IPv6 "unicast" addresses (see Reference 31 for details):

(1) Link-Local Addresses: used on a single link to support address auto-configuration and neighbor discovery. They also support communication between hosts when no routers are present on the link. Packets containing local-link addresses are never forwarded by routers.

(2) Site-Local Addresses: used by a site (or organization) that is not connected to the global Internet. Packets containing site-local addresses are not forwarded by routers outside the site (organization) in which the addresses are defined.

(3) Globally-Routable Addresses: used by nodes wishing to communicate with other nodes outside their own link and/or site. Mobile IPv6 deals primarily with this sort of address.

The virtue of the foreign agent concept in IPv4 Mobile Routing (see section 8.1.3) was that it provided a care-of address that was shared by many mobile nodes. Specifically, foreign agents eliminated the need to assign a unique, collocated care-of address to each and every mobile node. In IPv6, however, the availability of addressing is not considered to be a problem – there are more than 10^{38} possible addresses. The enormous address space of IPv6 allows simple auto-configuration of addresses. This allows a mobile node to quickly and easily acquire a collocated care-of address on any foreign link. Hence, IPv6 mobile routing does away with the "foreign agent" function entirely. All IPv6 care-of addresses are collocated.

IPv6 provides direct support for optimized (source) routing. Figure 8-13 below illustrates the concept. The upper part of the figure shows how Mobile IPv4 generates "triangle routes". A fixed node sends its packets first to the mobile node's home agent from which they are tunneled to the current foreign agent and then to the mobile node. This routing may be considerably longer (possibly involving many more router "hops") than the "optimized" route shown in the lower part of the figure. Why didn't Mobile IPv4 allow for this? The main reason was network security. To provide a direct tunnel to a mobile node, the fixed node must be informed of the mobile node's current care-of address. (Note that this is similar to the Registration protocol described in section 8.3 above.) Without strong authentication procedures that inform a mobile node's correspondent of its care-of address, it would be simple for a "hacker" to generate a "denial of service" attack. The "hacker" would only need to send a bogus registration to the mobile node's correspondent to cut off all communications between the two nodes. The problem is how to distribute security keys between a mobile node and every other node with which it might want to correspond. In the absence of an automated distribution mechanism, key handling in IPv4 proved to be unworkable.

It should be noted that the IPv4 "triangle route" requires that a message sent to a mobile node be transferred through a home-router that is separate from the control of the rest of the system. The organizations dealing with transmission of data to aircraft (FAA, etc.) may not wish to "trust" such components that are outside of their control. (Although they are currently doing this via ARINC's ACARS system.) IPv6 support for source-routing provides for more-complete control of the message path.

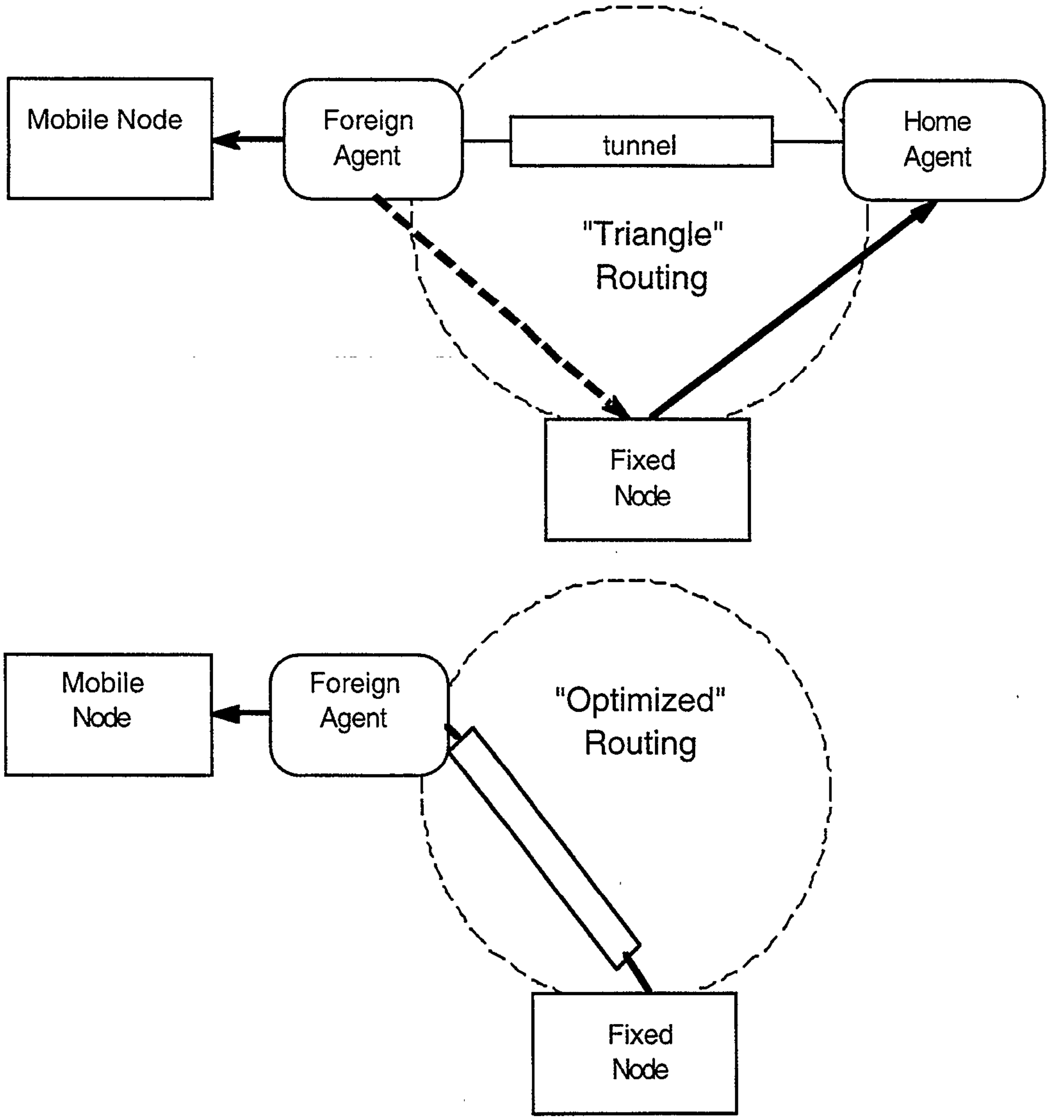


Figure 8-13 The "Triangle Route" problem.

Mobile IPv6 borrows many of the concepts and terminology of Mobile IPv4, as given in Table 8-1 below. The concepts of home address (see section 8.1.5), home link, care-of address (see section 8.1.6), and foreign link are similar to those of IPv4. Mobile IPv6 makes use of tunneling (see section 8.1.4) and source routing to deliver packets to mobile nodes. As was stated above, Mobile IPv6 does not use the concept of foreign agent.

TABLE 8-1 Comparison Between Mobile IPv4 and IPv6 Basic Concepts

Mobile IPv4 Concept	Equivalent Mobile IPv6 Concept
Mobile node, home agent, home link, foreign link	*(same)*
Mobile node's home address	Globally-routable home address and link-local home address
Foreign agent	A "plain" IPv6 router on the foreign link (foreign agent's do not exist in IPv6)
Foreign agent care-of address	All care-of addresses are collocated
Collocated care-of address	" " " " "
Care-of address obtained via Agent Discovery, DHCP, or manually	Care-of address obtained via Stateless Address Autoconfiguration, DHCP, or manually
Agent Discovery	Router Discovery
Authenticated registration with home agent	Authenticated notification of home agent and other correspondents
Routing to mobile nodes via tunneling	Routing to mobile nodes via tunneling and source routing
Route optimization via separate protocol specifications	Integrated support for route optimization

The operation of Mobile IPv6 can be summarized as follows. (1) A mobile node determines its current location using the IPv6 version of Router Discovery. (2) The mobile node acts like any fixed host or router when connected to the home link. If it is connected to a foreign link, the following steps are performed. (3) The mobile node uses IPv6-defined address autoconfiguration to acquire a collocated care-of address on the foreign link. (4) The mobile node informs its home agent of its care-of address. The mobile node also reports its care-of address to selected correspondents, assuming it can do so securely. (5) Packets sent by correspondents that are ignorant of the mobile node's care-of address are routed just as was done in Mobile IPv4 – namely, they are routed to the mobile node's home network, where the home agent tunnels them to the care-of address. Packets sent by correspondents that know the mobile node's care-of address are sent directly to the mobile node using an IPv6 Routing Header which specifies the mobile node's care-of address as an intermediate destination. (6) In the reverse direction (mobile node to fixed node), packets sent by a mobile node are routed directly to their destination using no special mechanisms.

Mobile IPv6 combines a number of various routing functions into the "Neighbor Discovery" protocol (see Reference 32 for details). Table 8-2 below lists these routing functions together with their closest analogs from Mobile IPv4.

TABLE 8-2 Mobile IPv6 Neighbor Discovery Functions and IPv4 Analogs

IPv6 Neighbor Discovery Functions	Similar IPv4 Functionality	Description
Router Discovery	ICMP Router Discovery	How a node locates routers on its link
Prefix Discovery	DHCP or manual configuration	How nodes determine the network-prefix(es) assigned to their current link
Parameter Discovery	Manual configuration	How nodes learn such things as the link MTU and "Time to Live" values
Address Autoconfiguration	DHCP	How a node automatically obtains an IP address for use on an interface
Address Resolution	ARP	How a node determines the link-layer address of a neighbor whose IP address is known
Next-Hop Determination	Routing Table Search	How a node chooses a "Next Hop" for any outgoing packets
Neighbor Unreachability Detection	*(no standard mechanism)*	How a node determines that a neighbor is no longer reachable
Duplicate Address Detection	*(no standard mechanism)*	How a node determines that its respective addresses are unique
Redirect	ICMP Redirect	How routers inform nodes of a better choice for the "Next Hop" to a destination

8.7 TCP Extensions for Mobility

The standard TCP implementation assumes that the overwhelming majority of lost IP packets is due to network congestion. This is a good assumption for the wires and fiber-optic cables of the fixed Internet, but it may not be a valid assumption for the "wireless" links connecting mobile nodes. When a standard TCP implementation discovers that a packet has been lost, it assumes network congestion and dramatically reduces its transmission rate. In a system whose transmission error rate is not very small, this can result in poor system performance. {Note: the error-correction and other link-layer protocols of VDL-2 serve to reduce the error rate substantially to a BER of approximately 10^{-9} – but the VDL-2 error rate is still higher than that of a wire!}

Similarly, consider the case of a home agent that has tunneled a packet to a mobile node's care-of address just before the mobile node moves to a new link. By the time that the tunneled packet arrives at the mobile node's old link, the mobile node is no longer there to receive it. Hence, TCP in the original sender will detect a lost packet, assume that there is network congestion, and reduce its transmission rate. Again, the performance of the system will suffer.

Several enhancements and extensions to TCP have been proposed to minimize the impact of mobility on system performance. A goal of these design proposals has been to limit the extent of software changes to only those systems that require mobility support. (Note: in the case of VDL-2/IP, these are the ADLP and GDLP.)

The most straightforward TCP change is simply to lengthen the TCP timer settings used to detect lost packets (in the VDL-2/IP ADLP and GDLP). Allowing more time for the link-layer re-transmissions (in the case of link errors) or the mobility protocol transactions (in the case of moving nodes) keeps TCP from assuming congestion prematurely.

A mobile node "knows" when it has moved by virtue of the agent discovery process and the re-registration procedure at the new foreign link. A proposed TCP extension termed "fast retransmit" (see reference 33) uses this mobility knowledge to keep TCP from assuming congestion when packets are lost during movement. Another proposal, termed "connection segmentation" (see references 34 and 35), separates the "wireless" TCP connection (i.e. between the VDL-2/IP ADLP and GDLP) from the "conventional" TCP connection in the existing Internet. The "wireless" connection can be more aggressive in treating transmission errors without causing system congestion. Yet another proposed TCP extension is termed "selective acknowledgment" (see reference 36). Standard TCP acknowledges receptions in sequence. Selective acknowledgment allows a node to inform another node of all segments it has successfully received, even if they are not sequential. This has the potential of preventing unnecessary retransmissions when a "hole" appears in the data stream.

9. Conclusions

This paper has described the proposed design of the VDL-2/IP datalink system which implements a standard Internet (IP) interface to the VDL-2 aviation datalink subnetwork. VDL-2/IP provides an architecture based on industry-standard techniques that will allow any combination of the following aviation (and non-aviation) datalink protocols to be transported as data over the VDL-2 subnetwork:

(1) ACARS
(2) ATN (ISO 8208)
(3) AOA
(4) FIS-B (Unnumbered Information Frames)
(5) Internet (IPv4, IPv6)

Any given VDL-2/IP implementation may "pick and choose" from among those protocols that it will support. (Note: the VDL CMU design may also contain support for SATCOM.) VDL-2/IP meets the goal of simplicity and provides for a straightforward, standardized implementation of a DLP. VDL-2/IP applications co-exist transparently with the ATN. VDL-2/IP imposes a minimal link overhead. VDL-2/IP allows the benefits of Internet infrastructure support for aviation applications. VDL-2/IP appears to provide the maximum in aviation datalink flexibility with a minimum of effort and expense.

Note: this paper has identified several technical details that would need to be completely specified before a final VDL-2/IP DLP could be built. As described in section 7.4 above, a VDL-2/IP special encoding for the "Unnumbered Information" frame AVLC Link Control field "function bits" might need to be assigned. Alternatively, the option of using "Numbered Information" AVLC frames with default sequence numbering for VDL-2/IP might be used instead. As was stated in section 7.4 above, this option enables the operation of the VDL-2 link-layer error-detection and retransmission protocols – resulting in a more-robust link. A special VDL-2/IP ARP Network Type encoding might need to be assigned (see section 7.6 above). Experimental VDL-2/IP systems could be built and tested by "sharing" the existing encodings (as described in section 7 above). The selection of an encoding for VDL-2/IP link layer broadcast addressing requires the final definition of the VDL-2/IP infrastructure (see section 7.3 above). The simple choice of "all ground stations" could be used for testing. The TCP transport timers might need to be modified for VDL-2/IP to account for the end-to-end delays and transmission errors inherent in the VDL-2/IP link (see section 8.6 above). If the delays get too long (or if the error rate gets too high), TCP will not get its acknowledgements in time and will re-transmit. This could result in unacceptable link performance. System simulation and testing will be required to determine the appropriate timer settings for TCP in this application.

Acronyms

A/G:	Air/Ground bit (AVLC)
AAC:	Airline Administrative Control
ACARS:	Airline Communications and Reporting System
ADLP:	Airborne Data Link Processor
AEEC:	Airlines Electronic Engineering Committee
AM-MSK:	Amplitude Modulation – Minimum Shift Keying
AOA:	ACARS over AVLC
AOC:	Aeronautical Operational Control
AOPA:	Aircraft Owners and Pilots Association
APC:	Airline Passenger Communications
ARINC:	Aeronautical Radio INCorporated
ARP:	Address Resolution Protocol
ATC:	Air Traffic Control
ATN:	Aeronautical Telecommunications Network
ATS:	Air Traffic Services
AVLC:	Aviation VHF Link Control
BER:	Bit Error Rate
C/R:	Command/Response bit (AVLC)
CLNP:	Connectionless Network Protocol (ATN-OSI)
CMU:	Communications Management Unit
COTS:	Commercial Off-The-Shelf
CRC:	Cyclic Redundancy Check
CSC:	Common Signaling Channel
CSMA:	Carrier-Sense Multiple Access
D8PSK:	Differential 8-Phase Shift Keying
DCE:	Data Communications Equipment (ISO 8208)
DHCP:	Dynamic Host Configuration Protocol
DLP:	Data Link Processor
DTE:	Data Terminating Equipment (ISO 8208)
EPI:	Extended Protocol Identifier
FCS:	Frame Check Sequence
FEC:	Forward Error Correction
FIS:	Flight Information Service
FIS-B:	FIS via Broadcast
GA:	General Aviation
GDLP:	Ground Data Link Processor
GRE:	Generic Routing Encapsulation

GSIF:	Ground Station Information Frame
HDLC:	High-level Data Link Control
HF:	High Frequency
ICAO:	International Civil Aviation Organization
ICMP:	Internet Control Message Protocol
IETF:	Internet Engineering Task Force
IP:	Internet Protocol
IPI:	Initial Protocol Identifier
IPv4;	IP version 4
IPv6:	IP version 6
ISO:	International Organization for Standards
LME:	Link Management Entity
MAC:	Media Access Control
MASPS:	Minimum Aviation System Performance Standard (RTCA)
MD5:	Message Digest algorithm 5
MIB:	Management Information Base
MOPS:	Minimum Operational Performance Standard (RTCA)
MTU:	Maximum Transfer Unit
OSI:	Open Systems Interconnection
P/F:	Poll/Final bit (AVLC)
PPP:	Point-to-Point Protocol
RARP:	Reverse Address Resolution Protocol
RF:	Radio Frequency
RFC:	Request For Comment (IETF)
RTCA:	Requirements and Technical Concepts for Aviation
SARPS:	Standards And Recommended Practices (ICAO)
SATCOM:	SATellite COMmunications
SC:	Special Committee (RTCA)
SITA:	State-of-the-art Information Technologies in Aviation
SLIP:	Serial Line Protocol
TCP:	Transmission Control Protocol
UDP:	User Datagram Protocol
UHF:	Ultra High Frequency
VDL:	VHF Data Link
VDL-2:	VHF Data Link Mode 2

VDL-2/IP:	VDL-2 Internet Protocol
VDL-3:	VHF Data Link Mode 3
VDR:	VHF Data Radio
VHF:	Very High Frequency
VME:	VDL Management Entity
XID:	eXchange Identity (AVLC)

References

(1) "iChip S7600A", Seiko Instruments, Torrance, CA.

(2) "TCP/IP Illustrated, Volume 1 – The Protocols", W. Richard Stevens, Addison-Wesley Publishing, 1994.

(3) "Issues Involved in the Development of an Open Standard for Data Link of Aviation Weather Information", Robert D. Grappel, M.I.T. Lincoln Laboratory 92PM-WINCOMM-0001, October 1999.

(4) ARINC/AEEC Document 618, "Air/Ground Character-Oriented Protocol Specification, January 2000.

(5) ARINC/AEEC Document 619-1, "ACARS Protocols for Avionics End Systems", October 1999.

(6) ARINC/AEEC Document 620, "Data Link Ground System Standard and Interface Specification (DGSS/IS), August 1999.

(7) "Minimum Aviation System Performance Standards (MASPS) for Flight Information Services-Broadcast (FIS-B) Data Link", RTCA Special Committee 195, Draft version 6.2, June 2000.

(8) ARINC Document 622-3, "ATS Data Link Applications Over ACARS Air-Ground Network", September 1998.

(9) ISO 3309 "HDLC Procedures – Frame Structure, Version 3", December 1993.

(10) ISO 4335 "HDLC Elements of Procedures, Version 3", December 1993.

(11) ISO 7809 "HDLC Procedures – Consolidation of Classes of Procedures, Version 1", December 1993.

(12) ISO 8885 "HDLC Procedures – General Purpose XID Frame Information Field Context and Format, Version [1]", December 1993.

(13) ARINC/AEEC Document 758 "Mark II Communications Management Unit", October 1999.

(14) ARINC/AEEC Document 750 "VHF Digital Radio (VDR)", October 1999.

(15) ICAO Annex 10, Volume 3, "Appendix to Chapter 9: A World-Wide Scheme for the Allocation, Assignment and Application of Aircraft Addresses"

(16) "Handbook of Computer-Communications Standards, Volume 1, The Open Systems (OSI) Model and OSI-Related Standards", W. Stallings, SAMS, 1990.

(17) ISO/IEC Technical Report 9577, “Information Technology – Protocol Identification in the Network Layer”, Fourth Edition, December 1999.

(18) ICAO “VDL Mode 2 SARPS”, July 1995.

(19) IETF RFC 826, “An Ethernet Address Resolution Protocol”, D.C. Plummer, 1982.

(20) IETF RFC 2002, “IP Mobility Support”, C. Perkins, October 1996.

(21) IETF RFC 1256, “ICMP Router Discovery Messages”, S. Deering, September 1991.

(22) IETF RFC 1144, “Compressing TCP/IP Headers for Low-Speed Serial Links”, V. Jacobson, February 1990.

(23) IETF RFC 2131, “Dynamic Host Configuration Protocol”, R. Droms, March 1997.

(24) IETF RFC 768, “User Datagram Protocol”, J. Postel, August 1980.

(25) IETF RFC 1321, “The MD5 Message-Digest Algorithm”, R. Rivest, April 1992.

(26) IETF RFC 792, “Internet Control Message Protocol”, J. Postel, September 1981.

(27) IETF RFC 2003, “IP Encapsulation within IP”, C. Perkins, October 1996.

(28) IETF RFC 2004, “Minimal Encapsulation within IP”, C. Perkins, October 1996.

(29) IETF RFC 1701, “Generic Routing Encapsulation (GRE)”, S. Hanks, T. Li, D. Farinacci, and P. Traina, October 1994.

(30) “draft-ietf-mobileip-ipv6-02.txt”, “Mobility Support in IPv6”, D. Johnson and C. Perkins, November 1996.

(31) IETF RFC 1884, “IP Version 6 Addressing Architecture”, R. Hinden and S. Deering, January 1996.

(32) IETF RFC 1970, “Neighbor Discovery for IP Version 6 (IPv6)”, T. Narten, E. Nordmark, and W. Simpson, August 1996.

(33) R. Caceres and L. Iftode, “Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments”, “IEEE Journal on Selected Areas in Communications”, 13(5), June 1995.

(34) A. Bakre and B. Badrinath, “I-TCP: Indirect TCP for Mobile Hosts”, Proceedings of the 15th International Conference on Distributed Computing Systems (ICDCS), May 1995.

(35) R. Yavatkar and N. Baghwat, “Improving End-to-End Performance of TCP over Mobile Internetworks”, Mobile 94 Workshop on Mobile Routing Systems and Applications, December 1994.

(36) IETF RFC 2018, “TCP Selective Acknowledgement Options”, M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, October 1996.