



**AIRCRAFT DATA NETWORK
PART 7
AVIONICS FULL-DUPLEX SWITCHED
ETHERNET NETWORK**

ARINC SPECIFICATION 664 P7-1

PUBLISHED: September 23, 2009

AN **ARINC** DOCUMENT

Prepared by AEEC
Published by
AERONAUTICAL RADIO, INC.
2551 RIVA ROAD, ANNAPOLIS, MARYLAND 21401-7435

DISCLAIMER

THIS DOCUMENT IS BASED ON MATERIAL SUBMITTED BY VARIOUS PARTICIPANTS DURING THE DRAFTING PROCESS. NEITHER AEEC, AMC, FSEMC NOR ARINC HAS MADE ANY DETERMINATION WHETHER THESE MATERIALS COULD BE SUBJECT TO VALID CLAIMS OF PATENT, COPYRIGHT OR OTHER PROPRIETARY RIGHTS BY THIRD PARTIES, AND NO REPRESENTATION OR WARRANTY, EXPRESS OR IMPLIED, IS MADE IN THIS REGARD.

ARINC INDUSTRY ACTIVITIES USES REASONABLE EFFORTS TO DEVELOP AND MAINTAIN THESE DOCUMENTS. HOWEVER, NO CERTIFICATION OR WARRANTY IS MADE AS TO THE TECHNICAL ACCURACY OR SUFFICIENCY OF THE DOCUMENTS, THE ADEQUACY, MERCHANTABILITY, FITNESS FOR INTENDED PURPOSE OR SAFETY OF ANY PRODUCTS, COMPONENTS, OR SYSTEMS DESIGNED, TESTED, RATED, INSTALLED OR OPERATED IN ACCORDANCE WITH ANY ASPECT OF THIS DOCUMENT OR THE ABSENCE OF RISK OR HAZARD ASSOCIATED WITH SUCH PRODUCTS, COMPONENTS, OR SYSTEMS. THE USER OF THIS DOCUMENT ACKNOWLEDGES THAT IT SHALL BE SOLELY RESPONSIBLE FOR ANY LOSS, CLAIM OR DAMAGE THAT IT MAY INCUR IN CONNECTION WITH ITS USE OF OR RELIANCE ON THIS DOCUMENT, AND SHALL HOLD ARINC, AEEC, AMC, FSEMC AND ANY PARTY THAT PARTICIPATED IN THE DRAFTING OF THE DOCUMENT HARMLESS AGAINST ANY CLAIM ARISING FROM ITS USE OF THE STANDARD.

THE USE IN THIS DOCUMENT OF ANY TERM, SUCH AS SHALL OR MUST, IS NOT INTENDED TO AFFECT THE STATUS OF THIS DOCUMENT AS A VOLUNTARY STANDARD OR IN ANY WAY TO MODIFY THE ABOVE DISCLAIMER. NOTHING HEREIN SHALL BE DEEMED TO REQUIRE ANY PROVIDER OF EQUIPMENT TO INCORPORATE ANY ELEMENT OF THIS STANDARD IN ITS PRODUCT. HOWEVER, VENDORS WHICH REPRESENT THAT THEIR PRODUCTS ARE COMPLIANT WITH THIS STANDARD SHALL BE DEEMED ALSO TO HAVE REPRESENTED THAT THEIR PRODUCTS CONTAIN OR CONFORM TO THE FEATURES THAT ARE DESCRIBED AS MUST OR SHALL IN THE STANDARD.

ANY USE OF OR RELIANCE ON THIS DOCUMENT SHALL CONSTITUTE AN ACCEPTANCE THEREOF "AS IS" AND BE SUBJECT TO THIS DISCLAIMER.

©2009 BY
AERONAUTICAL RADIO, INC.
2551 RIVA ROAD ANNAPOLIS, MARYLAND
21401-7435 USA

ARINC SPECIFICATION 664P7-1

AIRCRAFT DATA NETWORK
PART 7
AVIONICS FULL-DUPLEX SWITCHED ETHERNET NETWORK

Published: September 23, 2009

	Prepared by the AEEC	
Specification 664P7	Adopted by the Airlines Electronic Engineering Committee	April 26, 2005
	Summary of Document Supplements	
Supplement	Adoption Date	Published
Specification 664P7-1	April 15, 2008	September 23, 2009
A description of the changes introduced by each supplement is included on Goldenrod paper at the end of this document.		

FOREWORD

Aeronautical Radio, Inc., the AEEC, and ARINC Standards

ARINC organizes aviation industry committees and participates in related industry activities that benefit aviation at large by providing technical leadership and guidance. These activities directly support aviation industry goals: promote safety, efficiency, regularity, and cost-effectiveness in aircraft operations.

ARINC Industry Activities organizes and provides the secretariat for international aviation organizations (AEEC, AMC, FSEMC) which coordinate the work of aviation industry technical professionals and lead the development of technical standards for airborne electronic equipment, aircraft maintenance equipment and practices and flight simulator equipment and used in commercial, military, and business aviation. The AEEC, AMC, and FSEMC develop consensus-based, voluntary standards that are published by ARINC and are known as ARINC Standards. The use of ARINC Standards results in substantial benefits to the aviation industry by allowing avionics interchangeability and commonality and reducing avionics cost by promoting competition.

There are three classes of ARINC Standards:

- a) ARINC Characteristics – Define the form, fit, function, and interfaces of avionics and other airline electronic equipment. ARINC Characteristics indicate to prospective manufacturers of airline electronic equipment the considered and coordinated opinion of the airline technical community concerning the requisites of new equipment including standardized physical and electrical characteristics to foster interchangeability and competition.
- b) ARINC Specifications – Are principally used to define either the physical packaging or mounting of avionics equipment, data communication standards, or a high-level computer language.
- c) ARINC Reports – Provide guidelines or general information found by the airlines to be good practices, often related to avionics maintenance and support.

The release of an ARINC Standard does not obligate any organization or ARINC to purchase equipment so described, nor does it establish or indicate recognition or the existence of an operational requirement for such equipment, nor does it constitute endorsement of any manufacturer's product designed or built to meet the ARINC Standard.

In order to facilitate the continuous product improvement of this ARINC Standard, two items are included in the back of this volume:

An Errata Report solicits any corrections to the text or diagrams in this ARINC Standard.

An ARINC IA Project Initiation/Modification (APIM) form solicits any recommendations for addition of substantive material to this volume which would be the subject of a new Supplement.

ARINC SPECIFICATION 664 PART 7
TABLE OF CONTENTS

1.1	Purpose of Document	1
1.2	Scope	1
1.3	Document Organization	1
1.3.1	ARINC Specification 64, Aircraft Data Network.....	1
1.4	Related Documents	2
1.4.1	Relationship of this Document to Other ARINC Standards	2
1.4.2	Relationship to Industry Standards	2
1.4.3	RTCA and EUROCAE Documents.....	2
1.5	Document Precedence	3
1.6	Regulatory Approval	3
2.0	OVERVIEW.....	4
2.1	Comparative Model.....	4
2.2	Switched Ethernet Networks	5
2.3	Scalability	7
2.4	Ordinal Integrity.....	7
2.5	Fault Performance	8
2.6	Switching.....	8
2.7	System Performance	8
3.0	END SYSTEM SPECIFICATION	9
3.1	Introduction	9
3.1.1	ES Identification	10
3.2	Interoperability and Determinism at the Media Access Control (MAC) Layer	10
3.2.1	Virtual Link.....	10
3.2.2	Flow/Traffic Control	11
3.2.3	Scheduling.....	12
3.2.4	End System Performance	13
3.2.4.1	Latency	13
3.2.4.2	MAC Constraints	15
3.2.4.3	Jitter	16
3.2.5	MAC Addressing	17
3.2.5.1	MAC Destination Address	17
3.2.5.2	MAC Source Address	18
3.2.6	Redundancy Concept.....	19
3.2.6.1	Sequence Numbers and the Sending End System	22
3.2.6.2	Sequence Numbers and the Receiving End System.....	23
3.2.6.2.1	Integrity Checking	23
3.2.6.2.2	Redundancy Management.....	24

ARINC SPECIFICATION 664 PART 7
TABLE OF CONTENTS

3.3	Interoperability at the IP Layer and Above	26
3.3.1	Avionics Services	27
3.3.1.1	Communication Ports	28
3.3.1.1.1	Avionics Sampling Services.....	28
3.3.1.1.1.1	In Transmission	28
3.3.1.1.1.2	In Reception	29
3.3.1.1.2	Avionics Queuing Service	29
3.3.1.1.2.1	In Transmission	29
3.3.1.1.2.2	In Reception	30
3.3.1.2	SAP Port.....	30
3.3.1.2.1	Services to Compliant Network.....	30
3.3.1.2.2	SAP Port Error Management	30
3.3.1.2.3	File Transfer Services	30
3.3.1.3	The Sub-VL	31
3.3.2	Trivial File Transfer Protocol Example	33
3.3.3	ES Communication Stack.....	34
3.3.3.1	ES MAC Profile.....	34
3.3.3.2	ES IP Profile	35
3.3.3.2.1	IP Packet Structure	35
3.3.3.2.2	IP Fragmentation/ Reassembly.....	35
3.4	Network Level Interoperability	36
3.4.1	Addressing	36
3.4.1.1	Introduction.....	36
3.4.1.2	Structure of an AFDX Frame Without Fragmentation.....	36
3.4.1.3	Identification for End-to-End Communication	39
3.4.1.3.1	Intra-AFDX Communication	40
3.4.1.3.2	Extra-AFDX Communication.....	41
3.4.1.4	IP Addressing Format.....	41
3.4.1.4.1	IP Source Address	41
3.4.1.4.2	IP Destination Address	42
3.4.1.5	AFDX Communication Port, SAP and UDP/TCP Addressing Format	43
3.4.1.5.1	AFDX Communication Ports	44
3.4.1.4.2	SAP Ports	44
3.4.1.5.3	Allocation of the SAP and AFDX Communication Port Numbers	44
4.0	SWITCH SPECIFICATION	48
4.1	Basic Concepts	48
4.1.1	Filtering and Policing Function Introduction	49
4.1.1.1	Policing and Filtering Parameters.....	49

ARINC SPECIFICATION 664 PART 7
TABLE OF CONTENTS

4.1.1.2	Frame Filtering	49
4.1.1.3	Traffic Policing	50
4.2	Filtering and Policing Function	52
4.2.1	Frame Filtering	52
4.2.2	Traffic Policing	53
4.4	Switching Function	55
4.5	Switch End System Function	56
4.5.1	Overview	56
4.5.2	Addressing Policy	56
4.6	Monitoring Function	56
4.7	Configuration Files	58
4.7.1	Introduction	58
4.7.2	Default_Configuration_Table	59
4.7.2.1	Default Physical Port	59
4.7.2.2	Default Reception Configuration	59
4.7.2.3	Default Transmission Configuration	59
4.7.3	Field Loadable Configuration Tables: OPS_Configuration_File	60
4.7.3.1	EndSystem_Configuration_Table	61
4.7.3.2	Filtering_Policing and_ForwardingConfiguration_Table	61
4.8	Operating Modes	62
4.8.1	Overview	62
4.8.2	INIT	63
4.8.2.1	Initialization Sequence	64
4.8.2.2	Ground_Condition	65
4.8.3	OPS: Operational Mode	65
4.8.4	DL: Dataloading Mode	65
4.8.5	SHOP (Optional)	66
4.8.6	PASSIVE	66
4.8.7	QUIET	67
4.9	Dataloading	68
4.9.1	General Dataloading Requirements	68
4.9.2	Configuration Identification	68
4.9.2.1	Definition of Switch Configuration	68
4.9.2.2	Switch-On Configuration Identification	68
4.9.3	Dataloader IP Address	68
4.10	Pin Programming	69
4.10.1	Pin Programming Processing	69
4.10.1.1	When Pins Programming Should be Read	69
4.10.1.2	Parity Check and Processing	69

**ARINC SPECIFICATION 664 PART 7
TABLE OF CONTENTS**

4.10.2	List of Pin Programming	69
4.10.2.1	Position Coding	69
4.11	Performance Characteristics.....	69
4.11.1	General Characteristics.....	69
4.11.2	Physical Layer Characteristics	70
4.11.3	Processing Capabilities	70
5.0	SYSTEM ISSUES	72
5.1	Performances.....	72

ATTACHMENTS

1	Data Format	74
2	IP/ICMP, UDP, and TCP Profile Provisions	94

APPENDICES

A	An Example of ES Identification.....	125
B	Guidelines for ARINC 429 to AFDX Formatting.....	126
C	Network Terminology	130
D	Services to Protocol Mapping	132

ARINC Standard – Errata Report

ARINC IA Project Initiation/Modification (APIM)

1.0 INTRODUCTION

1.1 Purpose of Document

The purpose of this document is to define a deterministic network: Avionics Full Duplex Switched Ethernet (AFDX™). AFDX™ is a trademark of Airbus and is used with permission. This document also highlights the additional performance requirements of avionics systems, within the context of AFDX.

This specification allows:

- System integrators to design flight critical systems using AFDX
- Equipment designers to specify equipment interoperable with AFDX

1.2 Scope

The requirements listed in this document aim at specifying interoperable functional elements:

- Data link: MAC, VL addressing concept
- Network: IP, ICMP
- Transport: UDP, TCP (optional)
- Network application layers: Sampling, Queuing, SAP, TFTP, and SNMP

This document will be limited to the description of the normative protocols listed above.

COMMENTARY

System designers may add other protocols on a case-by-case basis (example FTP at network application layer), but there is no requirement for an AFDX compliant LRU to implement this additional protocol.

Physical layer is not specified here, but should be any of the ARINC Specification 664, Part 2, defined solutions.

This means that any AFDX compliant LRU (including switch) can be connected to any AFDX network, as far as the Form and Fit requirements (specific to the System and not specified in this document) are respected.

1.3 Document Organization

1.3.1 ARINC Specification 664, Aircraft Data Network

ARINC Specification 664 defines an Ethernet data network for aircraft installation. It is developed in multiple parts, listed as follows:

- Part 1 - Systems Concepts and Overview
- Part 2 - Ethernet Physical and Data Link Layer Specifications
- Part 3 - Internet-based Protocols and Services
- Part 4 - Internet-based Address Structures and Assigned Numbers
- Part 5 - Network Interconnection Services and Functional Elements
- Part 6 - Reserved
- Part 7 - Avionics Full Duplex Switched Ethernet (AFDX) Network
- Part 8 - Upper Layer Services

1.0 INTRODUCTION

1.4 Related Documents

1.4.1 Relationship of this Document to Other ARINC Standards

When standards for avionics systems and subsystems that use the capabilities provided by this specification are developed, they should incorporate the provisions of this specification by reference. References to this specification should assume the application of the most recent version of this Specification. A list of other ARINC documents that are related to this specification are listed below.

ARINC Specification 653: *Avionics Application Software Standard Interface*

ARINC Report 615A: *Software Data Loader using Ethernet Interfaces*

ARINC Report 665: *Loadable Software Standards*

1.4.2 Relationship to Industry Standards

IEEE Standard 802.3, 2000 Edition, is considered an integral part of this specification and is considered required reading. In this document, when referencing this standard, the title is shortened to simply “IEEE 802.3.”

1.4.3 RTCA and EUROCAE Documents

RTCA and EUROCAE develop Minimum Operational Performance Standards (MOPS) that are applicable to avionics equipment, systems and processes. The latest revision of the following RTCA and EUROCAE documents pertain to this Specification:

RTCA DO-160/EUROCAE ED-14: *Environmental Conditions and Test Procedures for Airborne Equipment.* In this document, when referencing this standard, the title is shortened to simply “DO-160.”

RTCA DO-254: *Design Assurance Guidance for Airborne Electronic Hardware.*

RTCA DO-178B: *Software Considerations in Airborne Systems and Equipment Certification.*

COMMENTARY

Specific performance levels defined in the RTCA/EUROCAE documents are specified by the aircraft systems integrator according to application.

1.5 Document Precedence

This Specification is based on IEEE Std 802.3. The philosophy of this Specification is to define changes to the provisions of IEEE 802.3 only when the aeronautical environment or user desires conflict with the provisions of IEEE 802.3, or when it is necessary to remove ambiguity by restricting the options available to implementers. The contents of this Specification are limited to describing these changes and option restrictions. In case of a conflict between this Specification and the applicable ISO and IEEE standards, this Specification should have precedence.

1.0 INTRODUCTION

1.6 Regulatory Approval

Implementation of this standard should meet all applicable regulatory requirements. Manufacturers are urged to obtain all necessary information for such regulatory approval. This information is not contained in this specification, nor is it available from ARINC.

2.0 OVERVIEW

An aircraft data network has been described elsewhere in this standard as a profiled version of an IEEE 802.3 Ethernet utilizing IP addressing and related transport protocols. Part 7 describes a subset of this network, where quality of service including timely delivery is paramount.

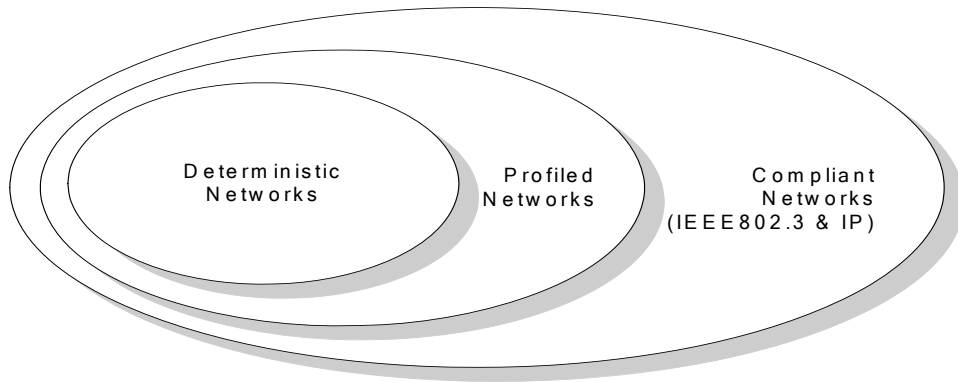


Figure 2-1 – Network Hierarchy

The AFDX network is a special case of a profiled network. A deterministic network may communicate with a wider profiled network and by inference, with a compliant network through routers or gateways. Figure 2-1 depicts this network hierarchy. For more information on the details of how the network services map to the network protocols, refer Appendix C.

Control systems generally, and avionics systems in particular, rely on having complete and up to date data delivered from source to receiver in a timely fashion. For safety critical systems reliable “real time” communication links are essential.

2.1 Comparative Model

As a comparison of timing issues, it is useful to compare an Ethernet network with a traditional aircraft bus. The following examples, illustrated in Figure 2-2, assume sequential messages with no error/retry.

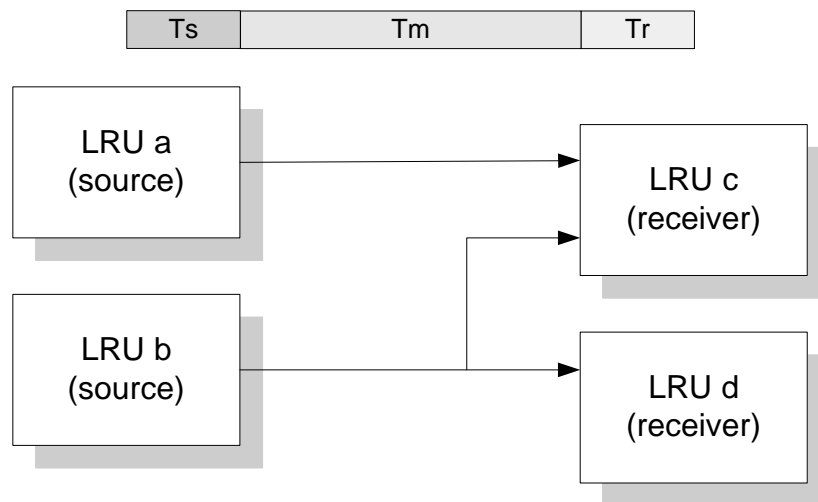


Figure 2-2 – ARINC 429 Bus

2.0 OVERVIEW

In this example the parameters are:

T_s = latency of delivery through the source end system

T_m = message timing ($\text{Length} \div \text{Bandwidth}$)

T_r = latency of delivery through the receiver end system

and the total latency, L , is :

$$L = T_s + T_m + T_r$$

Since the bandwidth is fixed, there are no collisions, and the end system latency is constant, the time taken to deliver a message, through the network, from the emitter to the receiver is constant. Other receiver end systems can sink the same message without impact on this timing, while the second source connected to LRU c will effectively be received by an independent end system.

This point-to-point system is almost ideally deterministic. The time for delivery of a message can be calculated and is constant. Increased bandwidth would lead to reduced message timing. Reliability can be determined by analysis or field measurements and a redundant system could be used.

Since each end-to-end link is independent of any other, there is no contention delay, and a fault in one end system will not affect the ability of fault free end systems to communicate.

2.2 Switched Ethernet Networks

In a system with many end points, point-to-point wiring is a major overhead. Ethernet networks can offer significant advantages and a suitable model for a deterministic network can be obtained through emulating the point-to-point connectivity. A segment of a star topology switched Ethernet network providing the same connectivity as the ARINC 429 example is shown in Figure 2-3.

The delay time through such nodes is not fixed by hardware delays alone; there is a variable quantity, jitter, which is caused by contention with other data on the network. It is common to analyze the network in terms of accumulated latency (including hardware delays and jitter effects) and link bandwidth. Ethernet Timing components are illustrated in Figure 2-3.

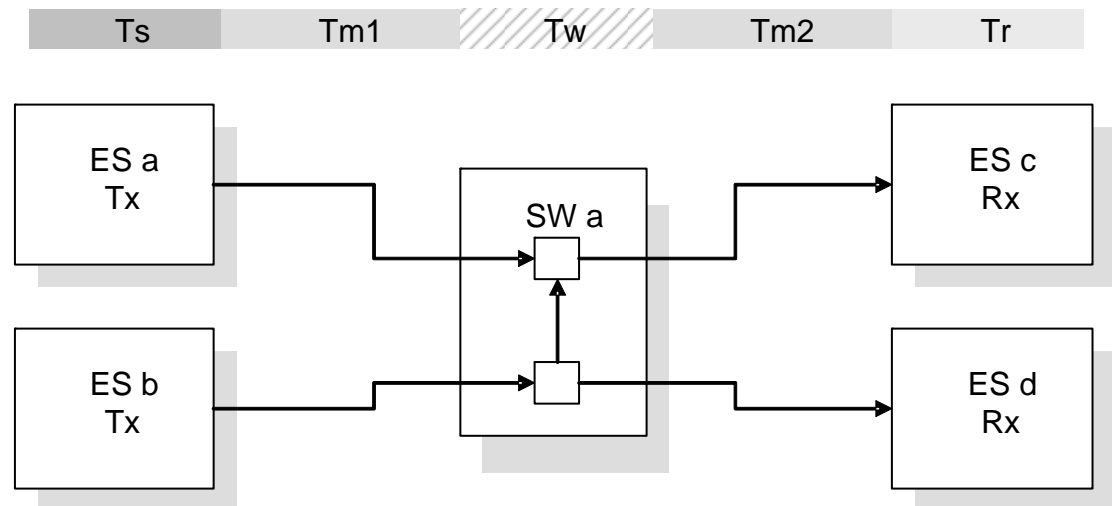


Figure 2-3 – Ethernet Timing

2.0 OVERVIEW

In this simple example, two end system transmitters (ES a and ES b) are offering data for use by ES c, the frames are delivered via switch SWa. Simultaneous reception of frames from ESa and ESb at the switch will be dealt with by sequencing the two frames for re-transmission to ES c, with a maximum jitter dependent on the message length:

$$T_j = (8 \times M) / N_{bw} + T_{min_gap} \text{ (it is assumed that the frames are of equal size).}$$

with: T_j = jitter time

M = frame size in octets

N_{bw} = media bandwidth, in bits/s

T_{min_gap} = minimum inter-frame gap, in seconds

Latency for the first frame:

$$L_a = T_s + T_{m1} + T_{sw} + (8 \times M) / N_{bw} + T_{m2} + T_r$$

with: T_{sw} = hardware latency in switch, in seconds

T_{mi} = message timing (length / bandwidth)

Latency for the delayed frame:

$$L_b = L_a + T_j$$

Hence, T_w being the total latency in the switch, in seconds (T_{sw} + output buffer latency), then T_w , referenced to an output port, depends on the traffic flow for that output port. In an asynchronous network, for any particular data stream, T_w will be time variant.

COMMENTARY

The contention in the switch and the store and forward capacity results in the need to buffer complete frames.

Hence the frame delay, $(8 \times M) / N_{bw}$, appears twice in the expression for L_b .

COMMENTARY

The notion of jitter exposed here is just an introduction. The reader should refer to Sections 3 and 4 for a detailed definition of jitter in an AFDX network.

In normal operation, the system remains deterministic for both data links (a – c) and (b – c), with a latency incorporating this jitter. For the system to maintain maximum integrity in the event of a fault on either ESa or ESb, the Quality of Service (QoS) definition may include continued delivery of data from the non-faulty node. This implies filter characteristics for the switch, which are not usual in commercial switch units, and is one method of fault isolation that can ensure the deterministic communication between two fault free end systems.

In a typical Ethernet network, the analysis assumes that the transmitting end-systems are not synchronized, and that frame delivery follows a random arrival distribution. This implies a Poisson emission distribution and although on average the bandwidth is known, over any arbitrarily short period of time the bandwidth is unlimited. Therefore, the distribution of latency on each connection through the network follows a distribution which is also unlimited. This results in a probabilistic description of latency in the network. Furthermore, there is no known method on a typical Ethernet network to limit the actual arrival distribution of data from any one

2.0 OVERVIEW

end-system within a Poisson distribution over any arbitrarily small interval of time. Therefore it is not possible to limit a bandwidth fault effect of an end system.

In an AFDX network, the analysis also assumes that the transmitting end-systems are not synchronized, but that frame delivery follows a bounded arrival distribution. This bounded arrival implies exact bandwidth regulated traffic control. Therefore, the distribution of latency on each connection through the network follows a distribution, which is bounded. This results in a calculable maximum latency in the network, rather than a probabilistic latency. Using exact bandwidth regulated traffic control it is possible to limit the bandwidth usage over any arbitrarily small interval of time. Therefore, it is possible to limit a bandwidth fault effect of an end system.

To continue the analogy with ARINC 429, each point-to-point wired data link can be replaced in the deterministic Ethernet network with a “Virtual link” (VL). The characteristics of each VL can be defined by analysis, assuming exact bandwidth regulated traffic control as illustrated in Figure 2-4. This is achieved by bounding both the bandwidth and the frame delivery interval for each VL.

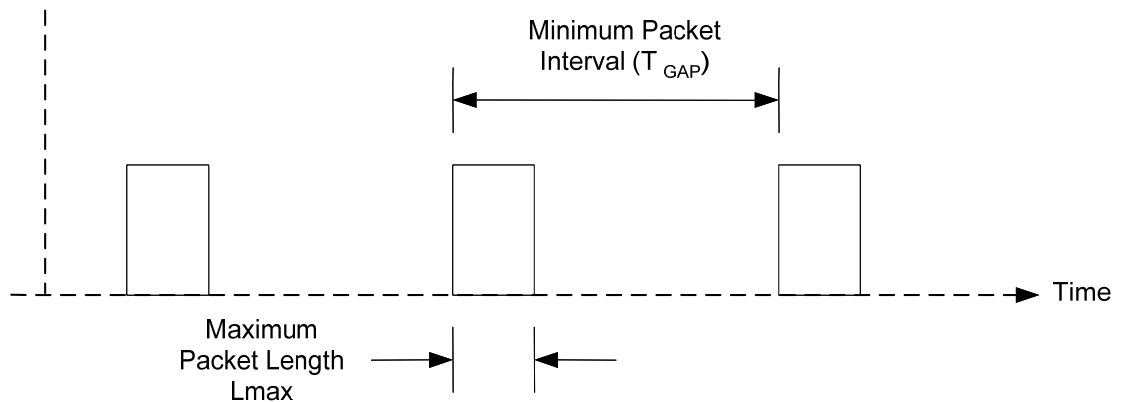


Figure 2-4 – Bandwidth Regulated Traffic Control

The maximum bandwidth allocated to this VL is: L_{max}/T_{GAP} .

The VL may use less bandwidth through an increased frame interval or smaller frames at any time. However, by bounding the characteristics of each VL the deterministic properties of the complete network can be analyzed.

2.3 Scalability

The choice of network topology will also impact the scalability of any system design. For AFDX, cascaded star topology was used because of its easy scalability.

In any practical network there will be incremental timing costs at expansion boundaries, for example when the number of ports on a single switch is exceeded.

2.4 Ordinal Integrity

Profiled networks provide no guarantee regarding the maintenance of ordinal integrity of frames across the network.

Avionics networks often contain data for which order is significant. Where there is an association between sequenced frames, ordinal integrity should be maintained. For this reason AFDX requires ordinal integrity of frames within a Virtual Link.

2.0 OVERVIEW

2.5 Fault Performance

In an Ethernet network topology it is not possible to treat each data link in isolation, for the purposes of performance analysis. A mechanism is needed to ensure that if one data link becomes faulty, the deterministic characteristics of the rest of the network are maintained. AFDX provides this mechanism in its relevant functional blocks.

2.6 Switching

In a commercial switch, the connection paths are established through the use of a “learning and aging” algorithm. Establishing a path for a newly arriving or aged data source will result in unknown latency.

2.7 System Performance

The deterministic characteristics of a network with asynchronous end points can only be evaluated by analysis of a given network topology. For each data stream a maximum latency can be derived as a guarantee of performance.

3.0 END SYSTEM SPECIFICATION

3.1 Introduction

The main function of the End System (ES) is to provide services, which guarantee a secure and reliable data exchange to the partition software.

Quality of Service (QoS) provides a method for categorising traffic and for ensuring that particular categories of traffic will always flow across the network at the service level to which they are entitled, regardless of competing demands.

For the Aircraft Network, there is no need to differentiate between several categories or traffic classes. Each network transmission request must be serviced regardless of the data type and a maximum network transit delay (also called end-to-end latency) must be guaranteed. Therefore, the only service class needed in the Aircraft Network is a guaranteed service.

A guaranteed service provides a firm, mathematically provable, upper bound on end-to-end frame transit delay. As a result, to guarantee a bounded delay implies to guarantee a certain bandwidth at the link level.

Hence a guaranteed service provides both upper bounded delay and constant bandwidth leading to a logical open connection between one transmitting node and one or more receiving nodes. Frames belonging to the same connection define a flow.

To summarise, a guaranteed service leads to the following characteristics:

- Bandwidth and bounded latency are guaranteed
- Particular delay jitter for a flow (end-to-end transit delay variation between any two frames of the same flow) is not fixed since it depends on the global network traffic at a given time. Nevertheless, a bound to the delay jitter can be mathematically computed

A description of the End System communication stack is shown in Figure 3-1:

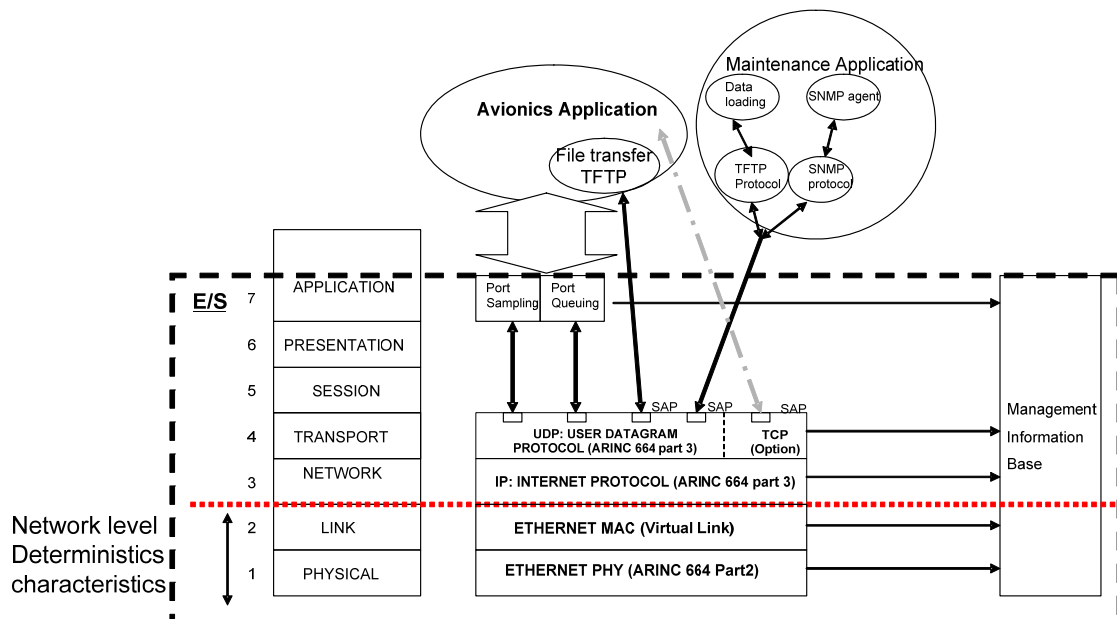


Figure 3-1 – End System Protocol Layers

3.0 END SYSTEM SPECIFICATION

3.1.1 ES Identification

A maximum of 16-bits only are available to the System Integrator to identify the ES. An example of ES identification using 12 bits is given in Appendix B.

3.2 Interoperability and Determinism at the Media Access Control (MAC) Layer

3.2.1 Virtual Link

A description of the “Virtual Link” concept is presented in Figure 3-2, since it is widely used in the document.

An end system may be designed to only receive VLs and not transmit VLs, or the contrary; therefore, an ES can originate or receive zero VLs. End-systems exchange Ethernet frames through VL. Only one End System within the Avionics network should be the source of any one VL.

A Virtual Link is a conceptual communication object, which has the following properties:

- A Virtual Link defines a logical unidirectional connection from one source end-system to one or more destination end-systems, shown in Figure 3-2.

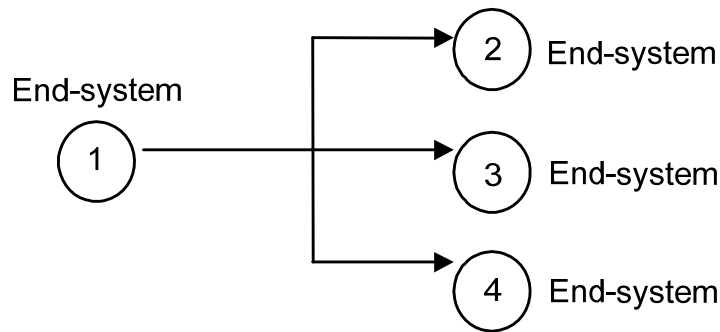


Figure 3-2 – A Virtual Link = A Path

- Each Virtual Link has a dedicated maximum bandwidth. This bandwidth is allocated by the System Integrator.

The ES should provide logical isolation with respect to available bandwidth among the Virtual Link(s) it supports. Regardless of the attempted utilization of a VL by one partition, the available Bandwidth on any other VL is unaffected.

For each Virtual Link, the End System should maintain the ordering of data as delivered by a partition, for both transmission and reception (ordinal integrity).

COMMENTARY

The Virtual Link processing is achieved through a flow control mechanism that regulates the flows of data produced by the different sources belonging to this ES. This mechanism provides partitioning at the network layer.

The End-system communication stack should guarantee in transmission the allocated bandwidth of each Virtual Link regardless of the attempted use of

3.0 END SYSTEM SPECIFICATION

Bandwidth by other Virtual Links, in order to preserve segregation between partitions at the network level. One Virtual Link should not be shared by two or more source partitions.

3.2.2 Flow/Traffic Control

At the output of the End System, the flow of frames associated with a particular Virtual Link is characterised by two of the parameters: Bandwidth Allocation Gap (BAG) and Jitter.

If the frames experienced no jitter from the scheduler (see Section 3.2.3, Scheduling), the BAG represents the minimum time interval between the first bits of two consecutive frames from the same VL, as illustrated in Figures 3-3 and 3-4.

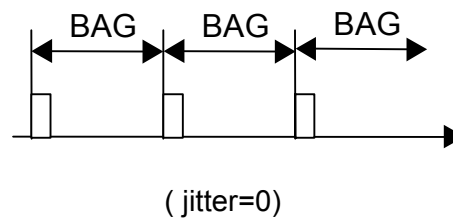


Figure 3-3 – The BAG in a VL for a Maximum Bandwidth Data Flow

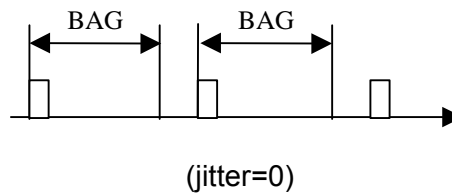


Figure 3-4 – The BAG in a VL For a Non Maximum Bandwidth Data Flow

COMMENTARY

A frame will not be transmitted while a VL is eligible but has no data for transmission.

To guarantee the BAG for each VL the frame flow is regulated as shown in Figure 3-5.

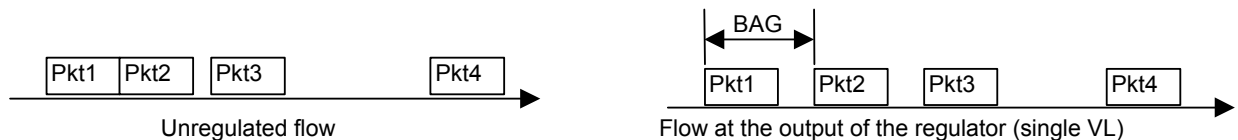


Figure 3-5 – Virtual Link Flow Regulation

3.0 END SYSTEM SPECIFICATION

3.2.3 Scheduling

In a transmitting end system with multiple VLs, the Scheduler multiplexes the different flows coming from the regulators, as illustrated in Figure 3-6.

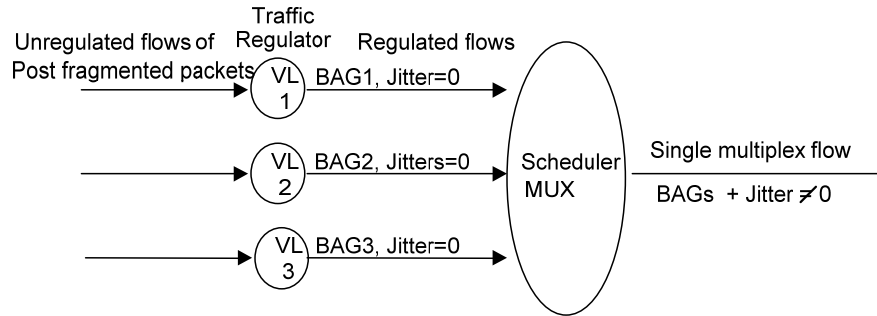


Figure 3-6 – Model of the Scheduled Flow Control Mechanism

At the output of the scheduler, for a given Virtual Link, frames can appear in a bounded time interval. This interval is defined as the maximum admissible jitter. This jitter is introduced by the scheduler and not by the traffic flow itself and is illustrated in Figure 3-7.

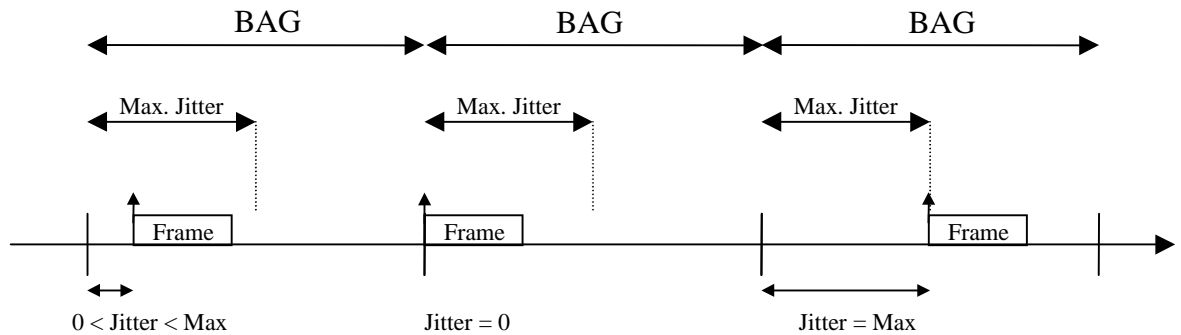


Figure 3-7 – The Jitter Effect for a Maximum Bandwidth Data Flow

The End System should regulate transmitted data on a per VL basis, since this Traffic Shaping Function (exact knowledge of flow characteristics) is the basis of the determinism analysis.

On a per VL basis the traffic regulator (traffic shaping function) should shape the flow to send no more than one frame in each interval of BAG milliseconds.

COMMENTARY

The aim of the traffic shaping function is to limit the instantaneous frame rate of the Virtual Links by spacing the frames. The regulator is responsible for controlling the bandwidth given to a Virtual Link according to the BAG.

3.0 END SYSTEM SPECIFICATION

The maximum usable bandwidth of each VL is characterised by its BAG and its authorized Lmax (maximum VL frame size). The maximum usable bandwidth = L_{max} / BAG in Kbytes per seconds.

The End System should accommodate VL frames up to a size of 1518 bytes in both transmission and reception.

For each VL, the End System should have one BAG value given by the End System configuration table.

COMMENTARY

The end system is configured using a configuration table file. The detailed contents of this table are beyond the scope of this document.

The Traffic shaping function of the ES should be able to handle BAG values in range 1 ms to 128 ms. These values should satisfy the following formula: $BAG = 2^k$ [in ms], (k integer in range 0 to 7).

COMMENTARY

If a partition needs to transmit data less often than 128 ms, the BAG value 128 will be used. BAG values are limited to powers of 2 in order to simplify the ES design.

3.2.4 End System Performance

The main goal of the system integrator is to be able to use an AFDX End System in a deterministic way. By creating a measure of ES performance, AFDX offers the system integrator a reduced burden of certification, and a flexible solution with well defined constraints.

3.2.4.1 Latency

The latency in transmission is defined as the duration between the following points of measurement as illustrated in Figure 3-8.

- Start - last bit of an hosted partition data is available to the communication services of the end-system
- End - last bit of the corresponding Ethernet frame is transmitted on the physical media

Measurements of the technological latency are made with empty buffers, with no conflicting resource access and no IP fragmentation as shown in Figure 3.8.

3.0 END SYSTEM SPECIFICATION

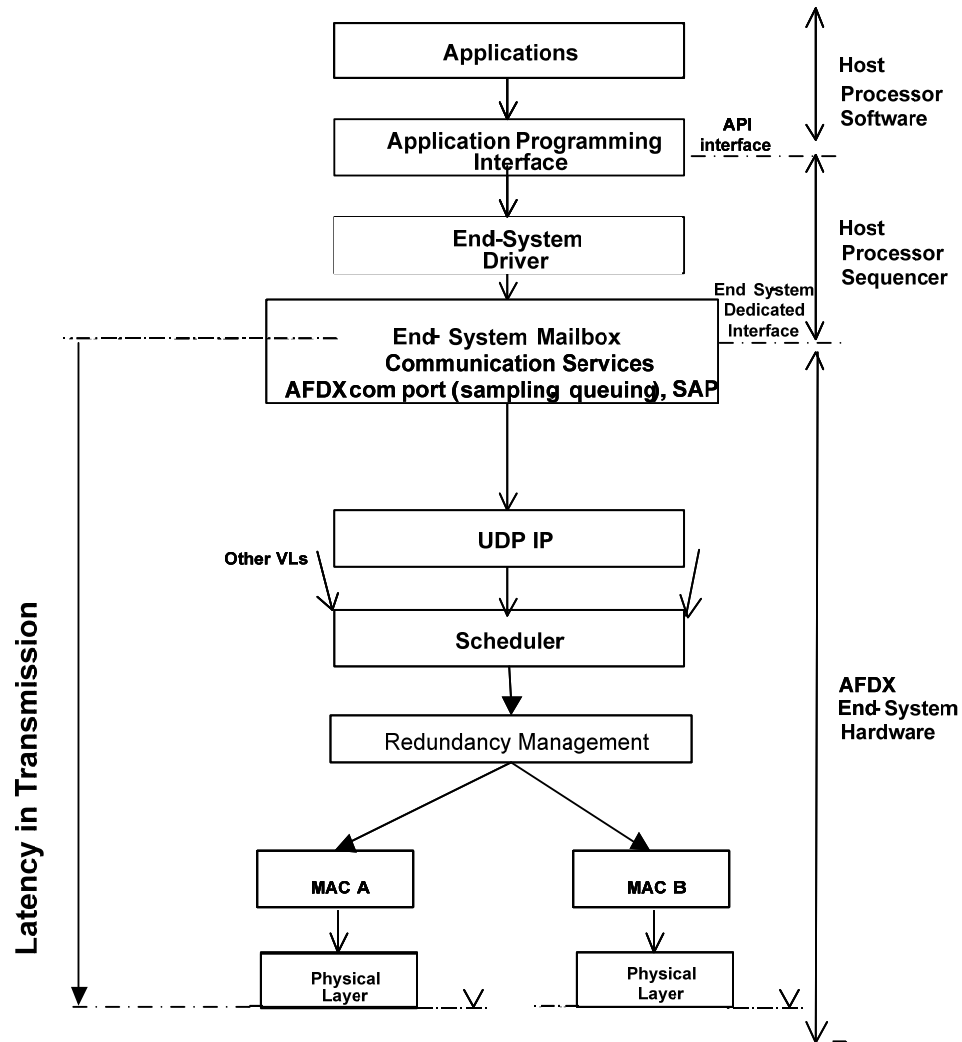


Figure 3-8 – Tx - Points of Performance Measurement

The technological latency of the end-system in transmission should be bounded and lower than $150 \mu\text{s} + \text{contention delay}$.

COMMENTARY

It is assumed that the total latency of the ES consists of technological latency (independent of traffic load) and configuration latency (depending on configuration and traffic load).

Technological latency is defined as the time required to accept, process, and begin transmission of application data when the end system is performing no other task.

The "**contention** delay" is added to cover the time taken to deliver the frame to the physical layer.

3.0 END SYSTEM SPECIFICATION

The latency in reception is defined between the following points of measurement:

- Start - last bit of an Ethernet frame is received on the physical media attachment.
- End - last bit of the corresponding data is available to the end-system hosted partition.

Measurements of the technological latency are made with empty buffers and without any conflicting resource access, as shown in Figure 3-9.

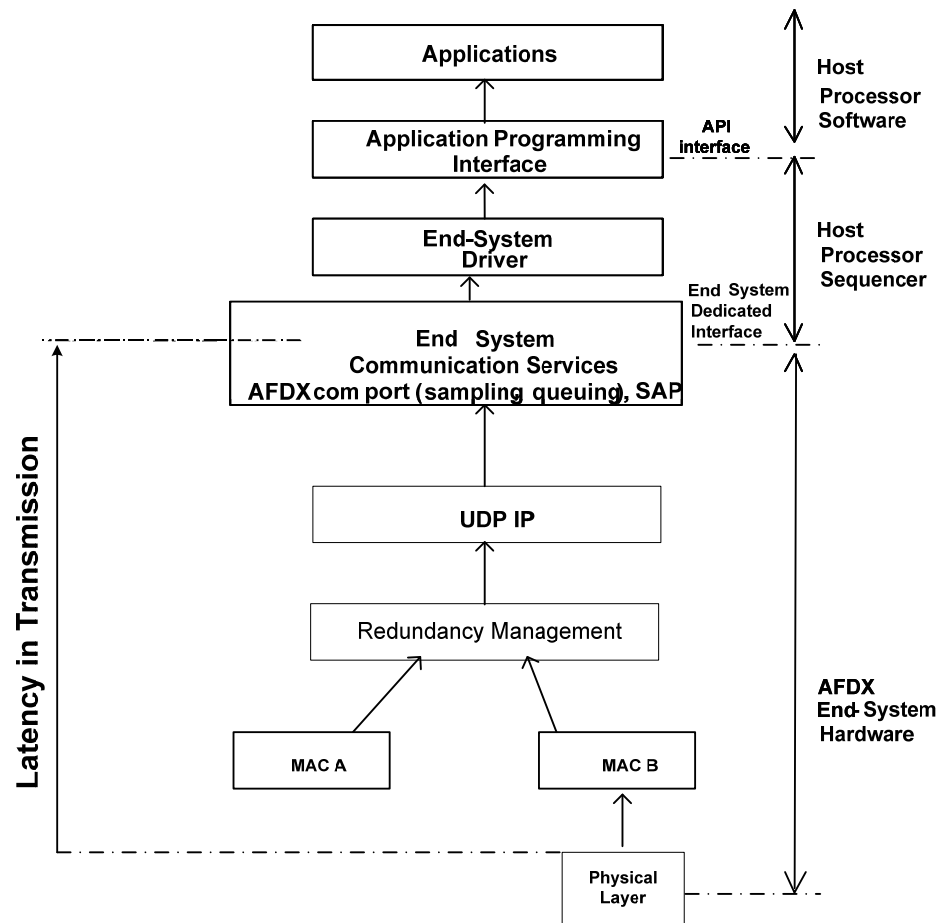


Figure 3-9 – Rx - Points of Performance Measurement

The technological latency of the end-system in reception should be bounded and lower than 150 μ s.

3.2.4.2 MAC Constraints

To avoid losing incoming frames during a burst, and to fix the IFG in transmission, the MAC layer of the end-system should be able to:

- Process received frames at full frame rate of the medium and appropriate (selected) frames are made available to the partition at full frame rate of the medium
- Transmit frames back to back

3.0 END SYSTEM SPECIFICATION

For the shortest frame this corresponds to a maximum frame rate per attachment of:

64 bytes (frame) + 12 bytes (IFG) + 7 bytes (Preamble) + 1 byte (SFD) = 84 bytes to transmit at 100 Mb/s.

Equivalent to a duration of 6.72µs per frame (about 148800 frames per second).

COMMENTARY

This requirement could be relaxed for transmission. Nevertheless, the designer should very carefully consider the impact for compliance on maximum jitter in transmission.

This requirement is more stringent in terms of processing capabilities for the shortest frame (64 octets) with a minimum inter-frame gap (12 octets).

3.2.4.3 Jitter

In transmission, the maximum allowed jitter on each VL at the output of the end-system should comply with both of the following formulas:

$$\begin{cases} \max_jitter \leq 40\mu s + \frac{\sum_{i \in \{\text{set of VLs}\}} (20 \text{ bytes} + L^{\max}_{\text{bytes}}) \times 8 \text{ Bits/bytes}}{\text{Nbw}_{\text{bits/s}}} \\ \max_jitter \leq 500\mu s \end{cases}$$

Note: max_jitter is in micro-seconds (µs); Nbw is medium bandwidth in bits/s; Lmax is in octets, 40µs is a typical minimum fixed technological jitter

According to the formula, the maximum allowed jitter will be lower for end-systems having few VLs and small frame sizes to process. In all cases, the jitter is bounded at 500µs to limit the impact on determinism for the whole network.

COMMENTARY

For heavily loaded ES (in transmission), optimized scheduling in transmission may make it possible to cope with the second formula. It is the system integrator's responsibility to determine that, for the chosen End System configuration and implementation, the 500 µs limit is not exceeded.

These values are fundamental to the demonstration of determinism for AFDX, and can be used to evaluate the limitations of an end system. A non-optimised ES (regarding jitter) will have bandwidth limitations resulting from limited processing capabilities.

To mathematically treat the allowed latency in the End System in transmission, two limiting cases are defined.

3.0 END SYSTEM SPECIFICATION

For the first case it is assumed that the hosted partitions transmit on a given VL evenly spaced data (no bursts) and that no data needs to be fragmented. Then, if the end-system has no other data to process on this virtual link, the total allowed latency for a given VLi is:

$$\text{MAX_Latency}_i \leq \text{BAG}_i + \text{Max_jitter} + \text{Technological_Latency_in_transmission}$$

The second case applies when the hosted partition transmits bursts of data or long messages requiring fragmentation. In this case, if the end-system has other data to process on this virtual link, the next data to transmit will be delayed. For the given VLi in transmission, and if (p-1) frames are being processed, the maximum latency of the frame number p should be bounded according to the following formula:

$$\text{MAX_Latency}_i (\text{frame}_p) \leq p * \text{BAG}_i + \text{Max_jitter} + \text{Technological_Latency_in_transmission}$$

The transmission will be delayed according to the bandwidth limitation of the VL (traffic shaping).

COMMENTARY

Some implementations could lead to an optimized solution regarding configuration latency. In all cases the values stated above should be adhered to.

3.2.5 MAC Addressing

3.2.5.1 MAC Destination Address

A Virtual Link should only be identified by the MAC destination address as illustrated in Figure 3-10, and the MAC source address of AFDX frames should be the MAC unicast address used to identify the physical Ethernet interface.

A MAC destination address in the AFDX frame should be a Group and Locally Administered address and should be compliant with the following format.

48 bits	
Constant field 32 bits	Virtual Link Identifier 16 bits
xxxx xx11 xxxx xxxx xxxx xxxx xxxx	

Figure 3-10 – MAC Multicast Addressing Format

Each ES should get "constant field" and "Virtual Link Identifier" values from the system integrator. The values are not specified in ARINC Specification 664.

The constant field should be the same for each ES in any given AFDX network. The least significant bit of the first byte indicates the group address (always = 1).

In order to use the standard Ethernet frame, MAC group addresses should be used to send frames from End System to End System(s).

The second to least significant bit of the first byte indicates the locally administered address (always = 1).

3.0 END SYSTEM SPECIFICATION

COMMENTARY

Even if at MAC layer, only group addresses could be used, unicast communication can be seen at IP layer level by using IP unicast destination address.

3.2.5.2 MAC Source Address

The MAC Source address should be an Individual and Locally Administered address compliant with IEEE 802.3. The structure of the address is specified in the following paragraphs.

COMMENTARY

No specific source MAC address construction algorithm is recommended. Therefore, it may be necessary for AFDX End Systems to have a means to determine the address construction algorithm being used in the network they are placed in. For example, pin programming might be used as a means to indicate which address construction rule is used.

Ethernet MAC Controller Identification (48-bits)			
Constant field: 24-bits	User_Defined_ID 16-bits	Interface_ID 3-bits	Constant field: 5-bits
"0000 0010 0000 0000 0000 0000 "	"nnnn nnnn nnnn nnnn"	"mmm"	"0 0000"

Figure 3-11 – MAC Source Addressing Format

The Constant field is set to "0000 0010 0000 0000 0000 0000" as shown in Figure 3-11.

The least significant bit of the first byte indicates the Individual Address = 0.

The second to least significant bit of the first byte indicates the locally administered address = 1.

The User_Defined_ID is a single 16-bit field. It should be used as the system integrator deems appropriate to give each IP addressable host on the network a unique and meaningful IP address.

The Interface_ID, defined in Figure 3-12, indicates to which redundant AFDX network(s) the Ethernet MAC controller is connected.

Interface_ID	Meaning
0 0 0	Not used
0 0 1	The Ethernet MAC controller is connected to the network A
0 1 0	The Ethernet MAC controller is connected to the network B
0 1 1	Not used
1 0 0	Not used
1 0 1	Not used
1 1 0	Nor used
1 1 1	Not used

Figure 3-12 – Interface_ID Definition

3.0 END SYSTEM SPECIFICATION

3.2.6 Redundancy Concept

End Systems communicate over multiple independent and redundant networks such that data flows are protected against the failure of any network component such as a link or a switch. The effect of this is to protect communication between End Systems against the loss of one complete network.

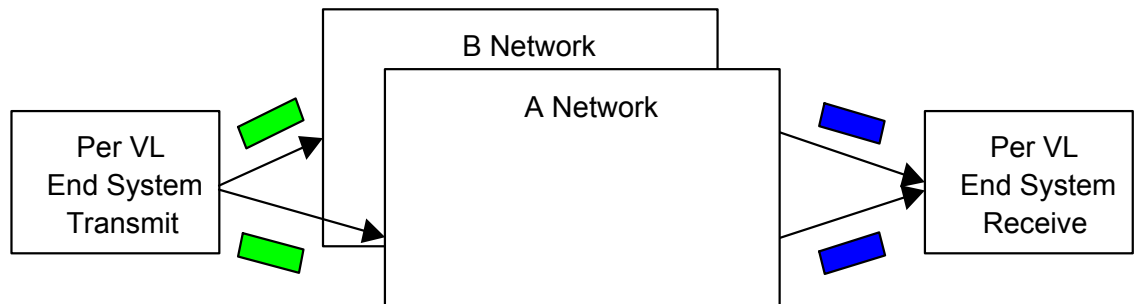


Figure 3-13 – Network Redundancy Concept

Figure 3-13 shows the basic concept for network redundancy. The redundancy scheme is operated on a per Virtual Link basis. A transmitting End-system and a receiving End-system communicate via a specific Virtual Link in the following manner:

A partition using transmitting End System prepares some data and passes it to the communications protocol stack. Here a sequence number field is added to each frame, and the sequence numbers are incremented on each successive frame. The sequence number is added to enable the receive function to reconstruct a single ordered stream of frames without duplication before delivery to the receiving partition. In this way the partition is unaware of the underlying network redundancy, and a simple interface can be built between the communications stack and partitions that utilize the network service.

In default mode each frame is sent across both of two networks. Upon reception, an algorithm in the communications stack (below IP layer) uses a “First Valid wins” policy. This means that the first frame to be received from either network with the next valid sequence number is accepted and passed up the stack to the receiving partition. When the second frame is received with this sequence number, it is simply discarded.

As the flow of frames given in Figure 3-14 below indicates, RM (Redundancy Management) is placed after IC (Integrity Checking). Under fault-free network operation, the IC just passes on the frames that it has received from a network on to the RM. The function of the AFDX redundancy management is merely to eliminate frames that are redundant copies of frames that it has already passed on to the partition.

3.0 END SYSTEM SPECIFICATION

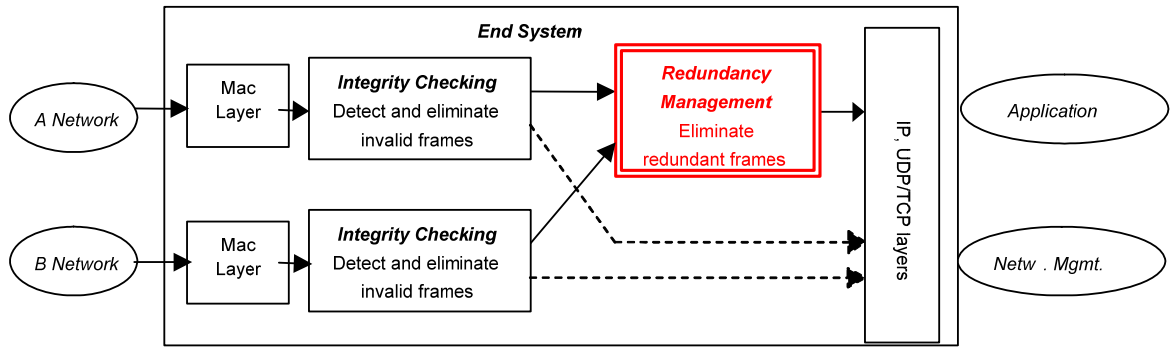


Figure 3-14 – Integrity Checking and Redundancy Management in the End System

Expected behavior is illustrated in Figures 3-15 through 3-18. The “RMA” line refers to the frames transmitted to the partition by the Redundancy Management Algorithm (RMA).

Example 1: Abnormal Transmitted Frame

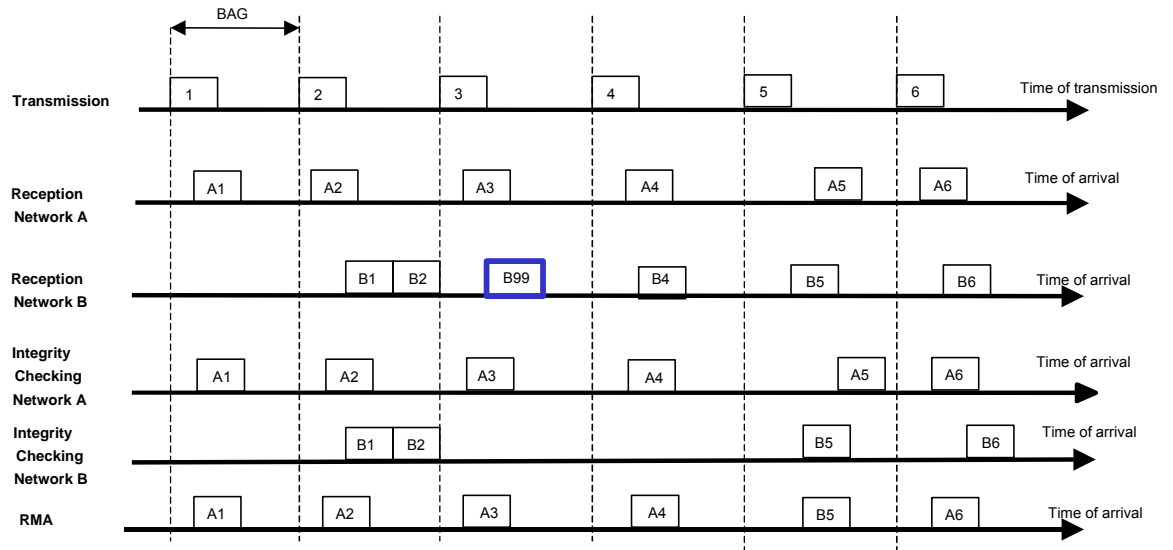


Figure 3-15 – Network B Transmits an Abnormal Frame

Redundancy management result: abnormal frame is not forwarded to partition.

Example 2: Loss of a Frame

3.0 END SYSTEM SPECIFICATION

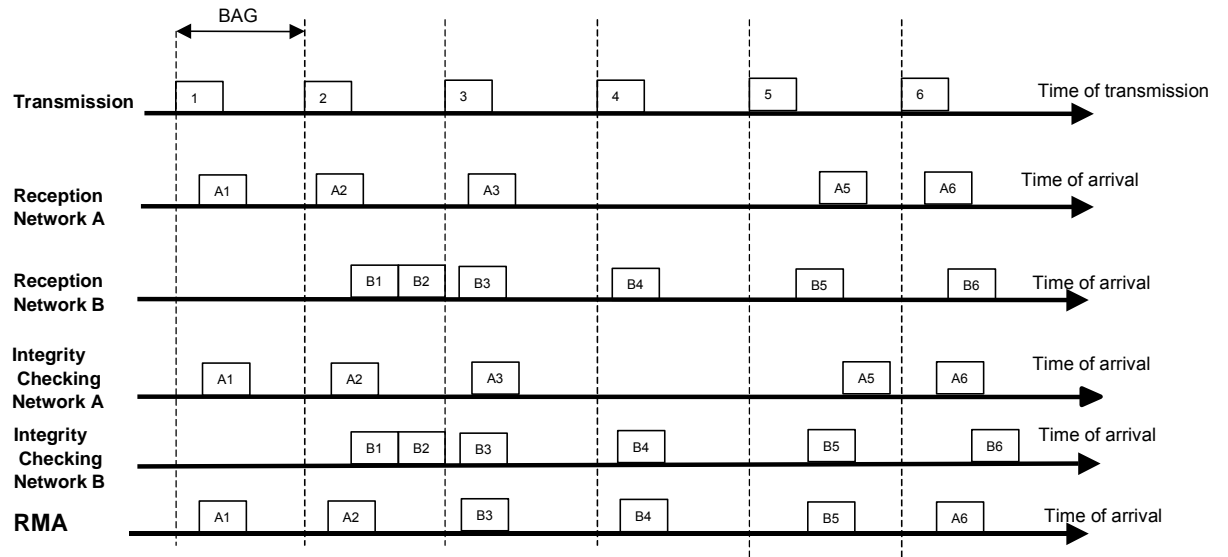
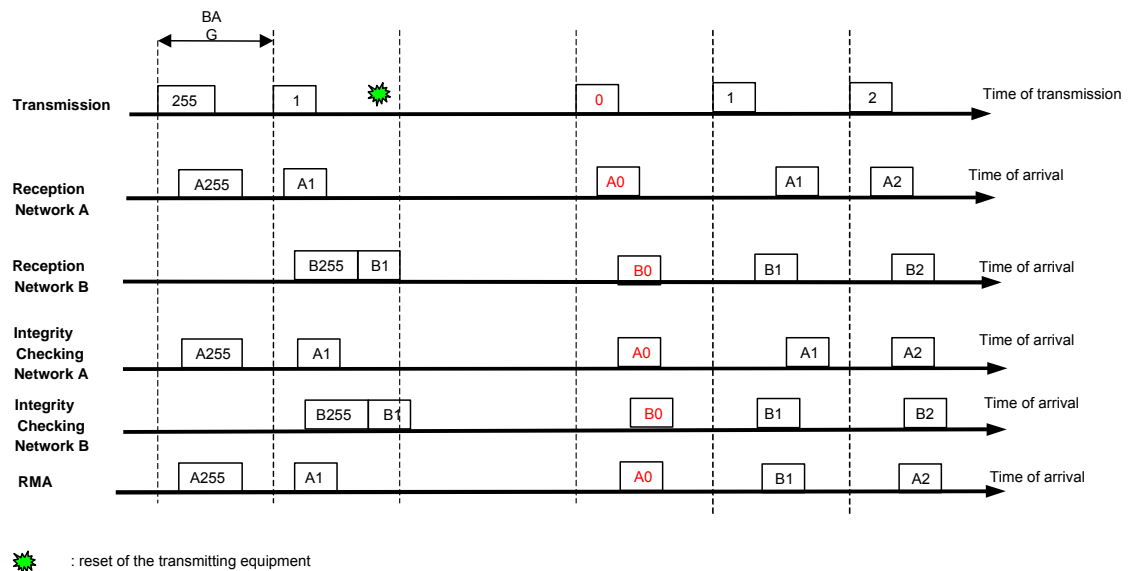


Figure 3-16 – A Frame is Lost on Network A

Due to a Bit Error, frame “A4” is lost.

Redundancy management result: The frame arriving on the B Network is accepted.

Example 3: Reset of the Transmitter




 : reset of the transmitting equipment

Figure 3-17 – Reset of the Transmitting End System

No frame is lost.

3.0 END SYSTEM SPECIFICATION

Example 4: Babbling Switch (stuck frame)

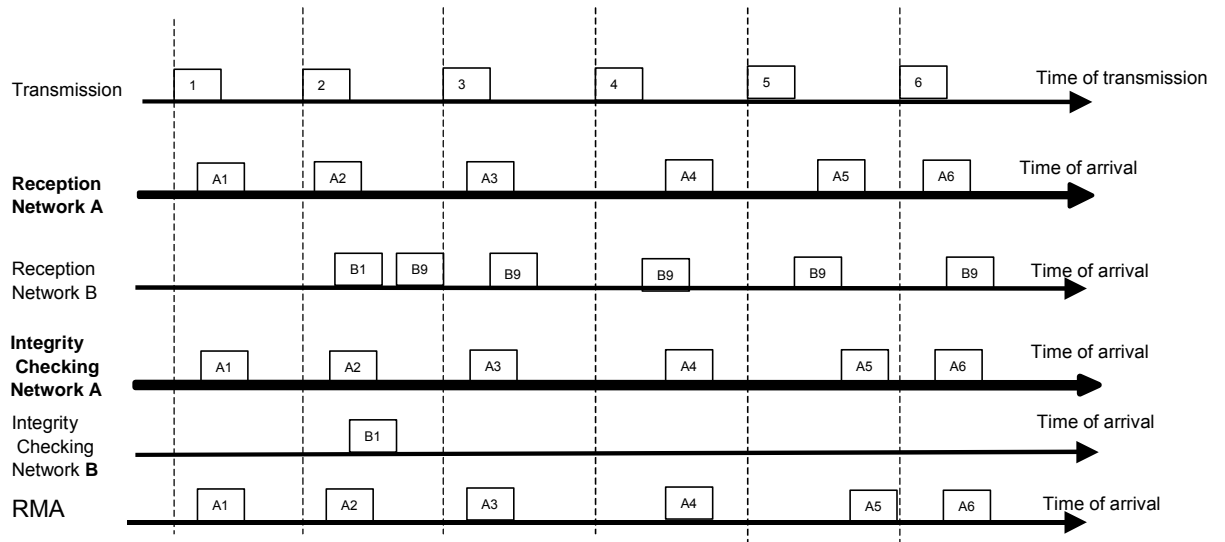


Figure 3-18 – Babbling on Network B

In example 4 the Redundancy Management result is that frames are not forwarded to the IP layer due to Integrity Checking

3.2.6.1 Sequence Numbers and the Sending End System

Per VL, the End System should add a sequence number for each transmitted frame on the AFDX network.

The frame sequence number should be one octet long with a range of 0 to 255.

COMMENTARY

This is sufficiently big to detect redundant frames under normal operation, but still compact. For example, in the worst case with BAG = 1 ms, SkewMax = 5 ms, the maximum SN offset between two received frames is:

$$\text{Int}\left[\frac{\text{SkewMax}}{\text{BAG}}\right] + 2 = 7 \quad \text{which is well below 128 to take into account wrap around.}$$

For each VL, the sequence number should initially be set to 0. The sequence number is always assigned this initial value following a reset of the transmitting ES.

The frame sequence number should be incremented by one for each consecutive frame of the same VL and wrap-around to 1 following the value 255.

3.0 END SYSTEM SPECIFICATION

COMMENTARY

Increment by one makes it possible to detect missing frames. Wrap-around to 1 allows the maximum range of sequence numbers, while reserving SN=0 for a reset condition. This improves integrity checking (see Section 3.2.6.2).

The frame sequence number should be located just before the MAC CRC field, as part of the MAC payload, as illustrated in Figure 3-19.

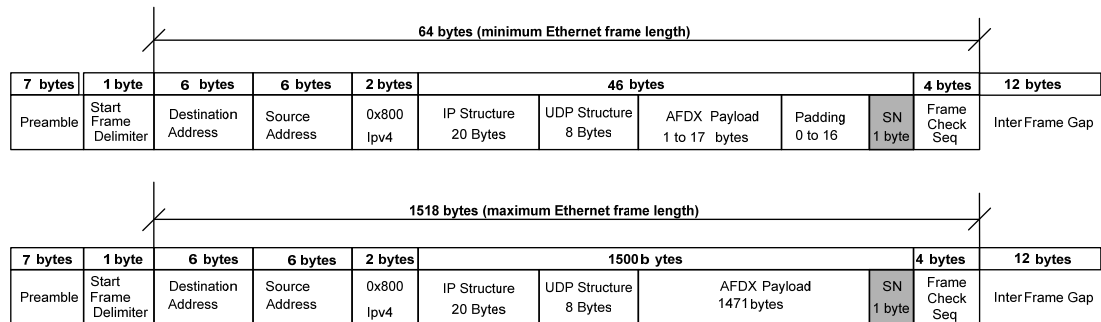


Figure 3-19 – Sequence Number Location in Frames for Minimum and Maximum Length

In order to simplify the algorithm on the receiving End System, redundant copies of a frame should be sent within a maximum time difference of 0.5 ms.

On a per VL basis, the ES should be able to send messages on either or both networks. This property should be configurable.

COMMENTARY

The system integrator is free to configure each Virtual Link with or without redundancy. If redundancy is turned off for any Virtual Link, a careful assessment of its impact should be performed on system integrity and availability.

3.2.6.2 Sequence Numbers and the Receiving End System

3.2.6.2.1 Integrity Checking

Under fault-free network operation, the Integrity Checking simply passes the frames that it has received on to the Redundancy Management, independently for each network. If there are faults (based on sequence number), the Integrity Checking has the task of eliminating invalid frames, and informs the network management accordingly. Refer to Section 3.2.6, Redundancy Concept, for sequence number usage.

For each network the Integrity Checking tests each frame for a sequence number in the interval:

$$[PSN + 1, PSN + 2]$$

3.0 END SYSTEM SPECIFICATION

Where:

- Previous Sequence Number (PSN) is the sequence number of the previous frame received (but not necessarily forwarded) on this VL.
- The operator “+” takes the wrap-around of Sequence Numbers into account. So, for example if PSN = 254 then PSN+1 = 255 and PSN+2 = 1.

COMMENTARY

This function increases integrity robustness. e.g.: by eliminating stuck frames or single abnormal frames and reducing the impact of a babbling switch. Loss of one single frame is considered as a normal event due to a non-zero Bit Error Rate.

The Integrity Checking should also accept the frame as valid in the following special cases:

- The Received Sequence Number (RSN) is equal to 0
- The frame is the first frame received after any reset of the receiving ES

Frames that do not meet these criteria are discarded.

COMMENTARY

These special cases improve the integrity of a-periodic data in particular. There would otherwise be systematic loss of a frame following a reset of either the transmitting ES or the receiving ES.

The sequence number 0 is transmitted only after a reset of the transmitting equipment.

It should be possible on a per VL basis to disable the integrity checking on both networks through the configuration table. Disabling Integrity Checking allows the receiver to accept all frames from both Networks, A and B.

3.2.6.2.2 Redundancy Management

The Redundancy Management (RM) assumes that the network is working properly and, in particular, the deterministic properties are verified.

Definitions:

- Redundant VL means that the same frames are sent through both network, A and B
- Non-redundant VL means that (possibly different) frames are sent through either network A or B

On a per VL basis, the ES should be able to receive:

- A redundant VL and deliver to the partition one of the redundant data (RM active)
- A redundant VL and deliver to the partition both redundant data (RM not active)
- A non redundant VL on either interface and submit data from it to the partition (in this case, RM can be active or not)

3.0 END SYSTEM SPECIFICATION

This RM function should be configurable.

The AFDX sequence number and SkewMax parameter should be used to identify redundant frames.

The use of SkewMax by a receiving End System is clarified as follows. The terms “Network A” and “Network B” are used arbitrarily.

If a frame sequence number is received from Network B within a time period SkewMax of the same frame sequence number having been received from Network A, the RM should identify the frame from Network B as a redundant copy and discard it.

If a frame sequence number is not received from Network B within a time period SkewMax of the same frame sequence number having been received from Network A, the next occurrence of the frame sequence number on either Network A or Network B should be identified as a new frame (not a redundant copy).

COMMENTARY

It will take at least 255ms (for BAG = 1ms) for the same sequence number to be received by an End System (sequence number to wrap around).

Per VL, if there is a large time period (> SkewMax) where a frame has not been received by the RM, the RM should accept any frame whatever its sequence number.

Appropriate values for SkewMax will depend on the topology of the network (the number of switch hops a frame must traverse), they should be provided by the system integrator. Values of SkewMax are in units of milliseconds (ms).

When redundancy management is active, it should deliver the first of the redundant frames received.

Reset of any equipment involved in the communication (transmitting End system, receiving End system or the AFDX switch) should not affect this property. This “First Win” philosophy enables availability of the network in the case of one AFDX switch loss.

COMMENTARY

The hardware reset time is assumed to be larger than SkewMax. This prevents the RM algorithm from being disturbed by frames sent on the network by the ES before the reset while frames sent after this reset arrives on the ES.

The Redundancy Management Algorithm should only use the RSN of a frame **and SkewMax** as the **criteria** for rejection or acceptance. Integrity checking is a separate task, which is performed even if no redundancy is used.

For each VL at the receiver, the Redundancy Management (RM) function should ensure that frames are forwarded in an increasing RSN order. This will still apply in

3.0 END SYSTEM SPECIFICATION

case of resets and occasional lost frames. **An exception to this is the case when SkewMax has been exceeded for the arrival of a redundant frame. RM will accept the next frame whatever its sequence number is.**

The Redundancy Management function must forward only in-sequence frames, but reordering is not required. As a consequence, in some cases, the loss of one frame on one network could also lead to the loss of its copy.

For example, in Figure 3-20, the frame “A2” is lost on the A network and the frame “A3” arrive on this network before the copy “B2” of the lost frame arrives on the B network. In this case, the copy B2 will not be forwarded to the partition despite being the first #2 frame received.

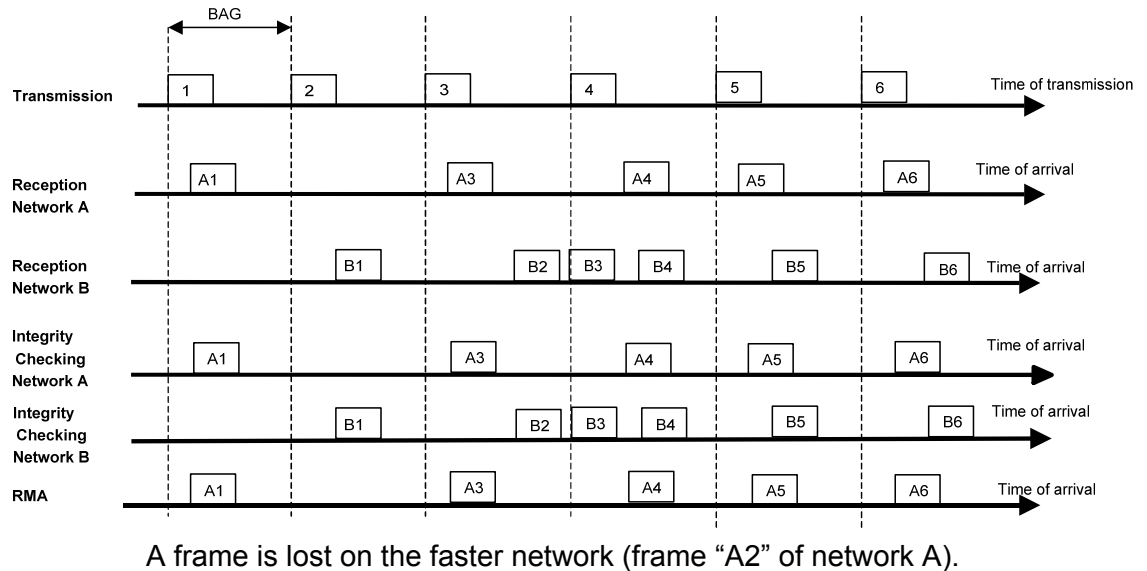


Figure 3-20 – Loss of a Frame

3.3 Interoperability at the IP Layer and Above

COMMENTARY

There is no standard mapping between Partition ports (Application ports) and the ports of the ES. These requirements will be written into the specification of particular equipment.

Nevertheless, having two data consumers of an AFDX Com or Service Access Points (SAP) port may result in a loss of data as illustrated in Figure 3-21.

3.0 END SYSTEM SPECIFICATION

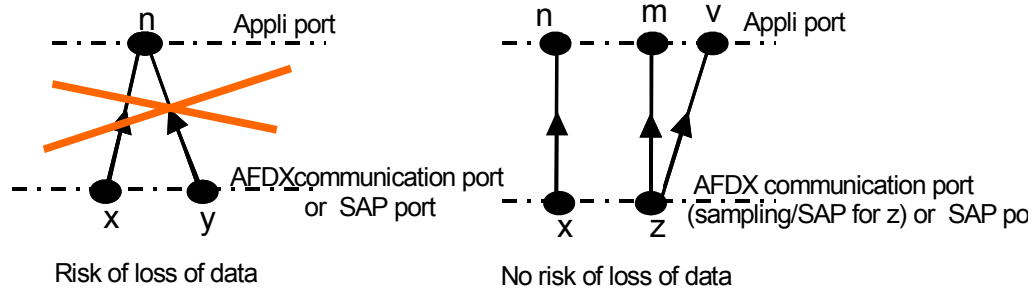


Figure 3-21 – Sharing Receive AFDX Com Ports

Having two data sources for one transmit AFDX com or SAP port may also result in a loss of data as illustrated in Figure 3-22.

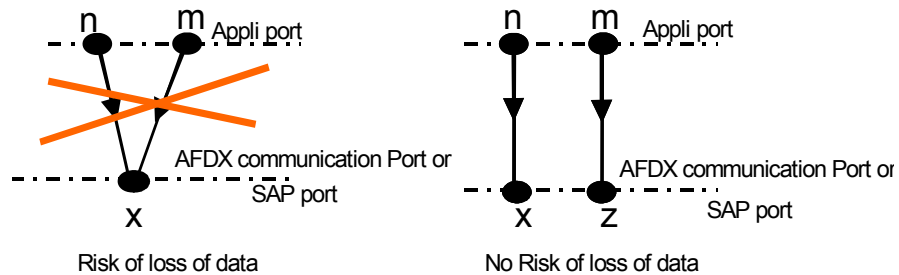


Figure 3-22 – In Transmission, Ports are not Shared

A user port is configured to either transmit or receive, but not both.

The example with red 'X' should not be used.

3.3.1 Avionics Services

The ES provides different modes of transfer from an Avionics Partition point of view with two types of ports:

1. Communication port: Sampling or queuing modes (e.g., ARINC 653)
2. SAP ports: Used for TFTP transfers and communication with compliant networks

3.0 END SYSTEM SPECIFICATION

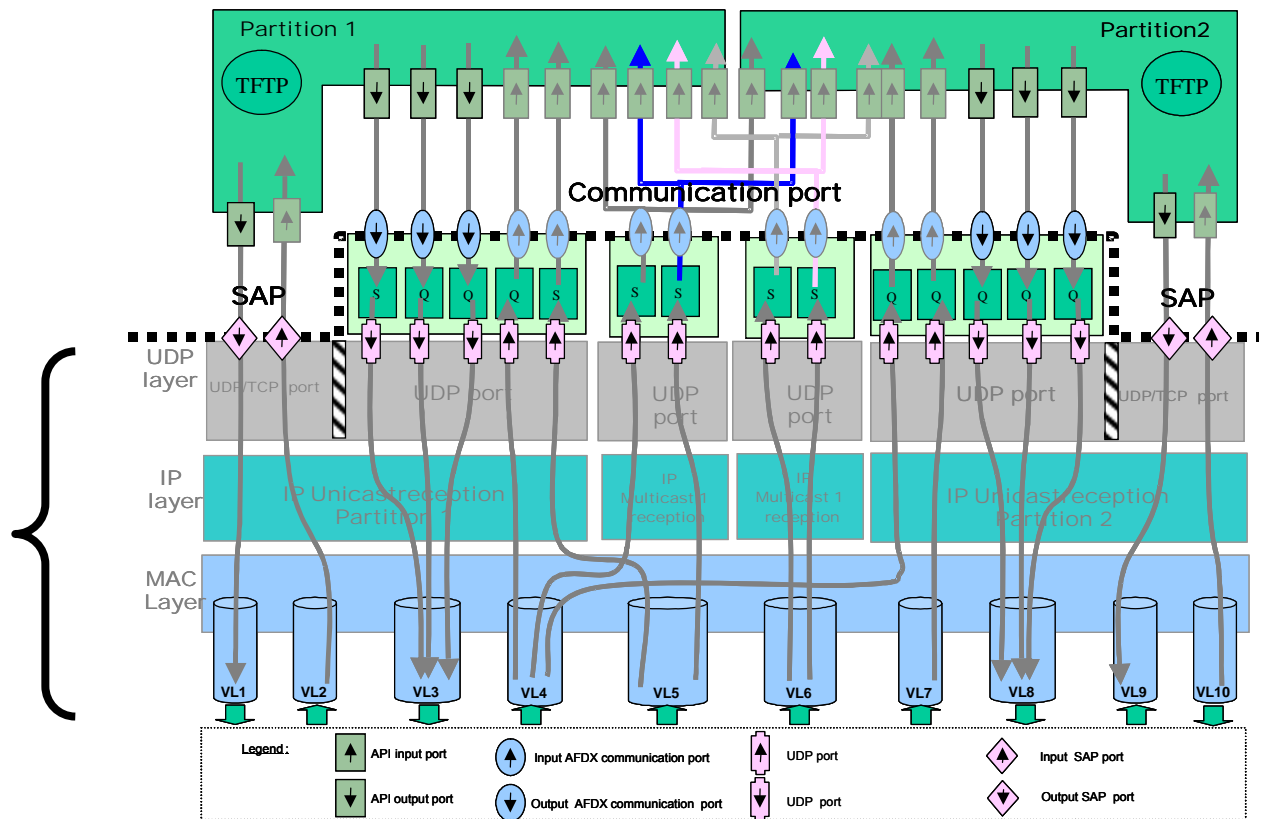


Figure 3-23 – Interface Between Partition and End System

Figure 3-23 describes equipment which has two partitions (e.g., ARINC 653 for a definition of a partition) and an End System. Each partition has an IP address. To communicate with a partition, the End System uses two port types: Communication Port and SAP.

3.3.1.1 Communication Ports

The ES provides two types of services via the Communication Ports: Sampling and queuing. UDP has been chosen for both services due to its relative efficiency.

3.3.1.1.1 Avionics Sampling Services

The End System should provide sampling services as defined in ARINC 653 (Section 2.3.5.6.1).

3.3.1.1.1.1 In Transmission

Sampling Com Ports should not use IP fragmentation, therefore the size of each sampling message should be less than or equal to the payload size of the associated Virtual Link.

Sampling service should be based on multicast and unidirectional communication to send data from one source to one or several receptors.

3.0 END SYSTEM SPECIFICATION

The sampling service is simple, connectionless without acknowledge. It does not add error control on transmitted data and does not require flow control in addition to the VL flow control. This transmission type is similar to services provided by classical ARINC 429 links.

3.3.1.1.2 In Reception

The last message stored in a particular sampling port should be able to be read by several partitions (i.e. several partitions may subscribe to this sampling port).

A freshness indication should be associated to each sampling port. This freshness indication should be available to each partition reading the message.

3.3.1.1.2 Avionics Queuing Service

The End System should provide queuing services to an avionics partition, as defined in ARINC 653 (Section 2.3.5.6.2).

The queuing service is simple, connectionless, without acknowledgement.

The queuing service should be able to manage messages of different sizes for the same queuing Communication Port.

To guarantee the sequence of messages, the queuing service should manage the messages with FIFO discipline in transmission and reception.

Each instance of Queuing Service should be able to manage up to 8 k octets of application data (IP fragmentation is thus needed).

COMMENTARY

A connectionless queuing service without acknowledgement is acceptable for a variety of communications due to the low probability of frame loss in the redundant AFDX network.

3.3.1.1.2.1 In Transmission

When fragmentation is used, the fragments should be transmitted in-order to the AFDX network.

If a buffer overflow occurs in transmission, an error indication should be sent to the transmitting partition and the frame should be discarded, as illustrated in Figure 3-24.

3.0 END SYSTEM SPECIFICATION

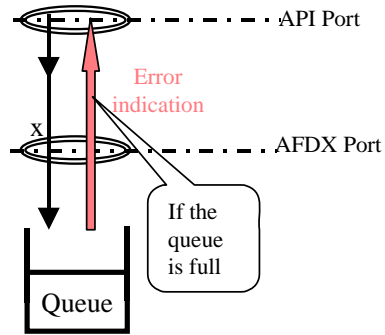


Figure 3-24 – Error Indication for TX Buffer Overflow

3.3.1.1.2.2 In Reception

In case of fragmentation, no data should be presented to a partition in the queuing services FIFO until whole message has been reassembled.

If a buffer overflow occurs in reception, an error message should be sent to the receiving partition and the frame should be discarded.

3.3.1.2 SAP Port

3.3.1.2.1 Services to Compliant Network

An ES can be a Service Access Point with the following characteristics:

- The SAP ports could be used for communications within the AFDX network.
- Access to the compliant network can be through either a gateway or a router, as part of the ES design.
- The ES should provide UDP services to communicate with Compliant Network.
- Each instance of the UDP Service Access Point should handle up to 8k octets.
- As an option, TCP could be used by accessing the IP layer directly through an appropriately configured SAP port.

To communicate with a Compliant Network, a transmitting ES has the ability to specify the destination address: IP address and port number. For this purpose, these addresses are made available when the ES receives a request from the compliant network.

3.3.1.2.2 SAP Port Error Management

Degradation of the Quality of Service in a SAP port is monitored at the receiver. If receiver buffer overflow occurs, an error message is sent to the receiving partition, and the frame should be discarded.

3.0 END SYSTEM SPECIFICATION

3.3.1.2.3 File Transfer Services

The Trivial File Transfer Protocol “TFTP” should be used to transfer files.

The specification of TFTP is defined in the RFCs identified in Table 3-1.

Table 3-1 – RFC Definitions for the TFTP

RFC	Title	Category
783	The TFTP protocol (Revision 2)	Standard, Updated by RFC 1350
1123	Requirements for internet Hosts Application and support	Standard
1350	TFTP protocol (Revision 2)	Standard
2347	TFTP option extension	Standards track, Updates 1350
2348	TFTP Blocksize Option	Standards track, Updates 1350
2349	TFTP Timeout Interval and Transfer Size Option	Standards track, Updates 1350
1785	TFTP option negotiation analysis	Informational, Updates 1350

Each instance of File Transfer service should be able to manage up to 8 k octets blocks.

3.3.1.3 The Sub-VL

A VL can be composed of a number of Sub-VLs, as illustrated in Figures 3-25 through 3-27 and in this case the VL is made up only from these Sub-VLs.

Each Sub-VL has a dedicated First In, First Out (FIFO), and the Sub-VL FIFO queues are read on a round robin basis by the (main) VL FIFO queue. This round robin function is done on a MAC frame basis, therefore IP fragmentation (if any) would have been done prior to loading the Sub-VL FIFOs.

COMMENTARY

Implementation of Sub-VL is an optional feature that has no impact on the determinism of the network. It can be used to optimise the bandwidth utilisation of a VL.

3.0 END SYSTEM SPECIFICATION

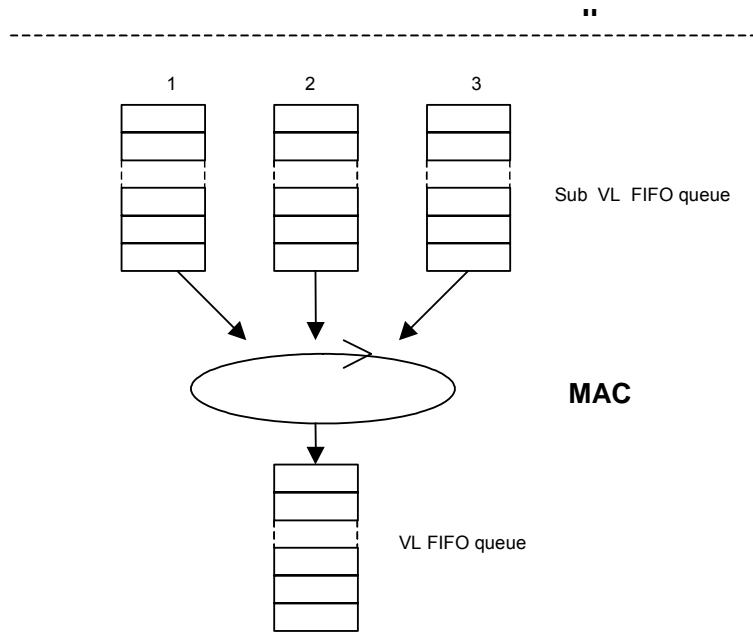


Figure 3-25 – The Sub-VL FIFO Queue

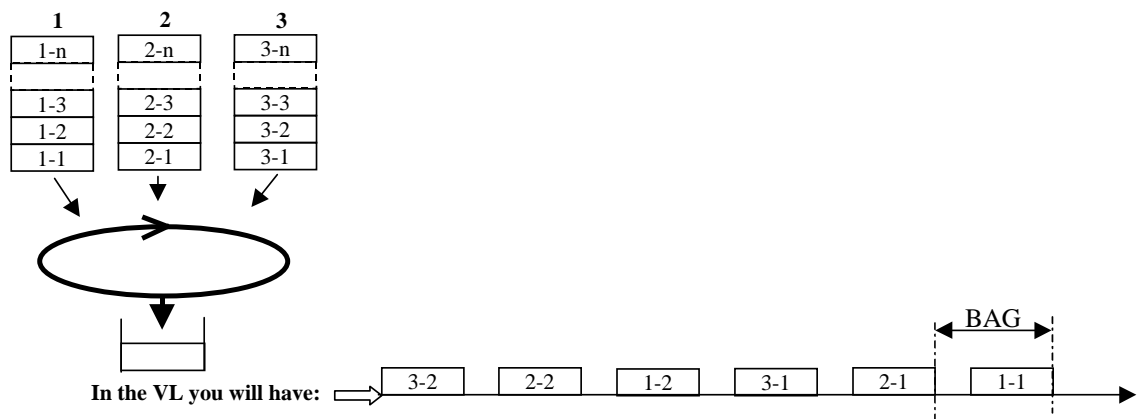


Figure 3-26 – 1st Example of Traffic on the VL

3.0 END SYSTEM SPECIFICATION

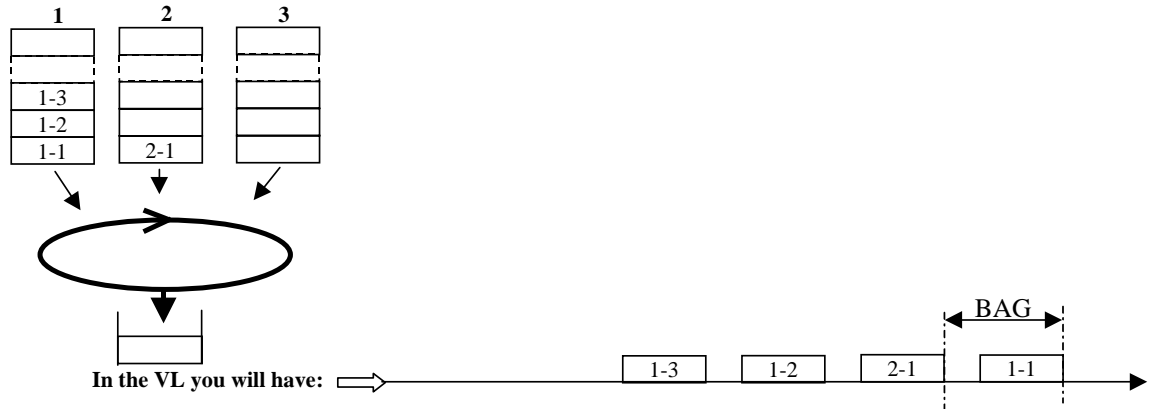


Figure 3-27 – 2nd Example of Traffic on the VL

A VL FIFO queue should be able to manage at most 4 Sub-VL FIFO queues.

Each Sub-VF FIFO queue should be read in round-robin sequence such that if any Sub-VF FIFO queue has traffic, one frame per BAG is sent to the main VL. Once a frame is sent, the sequence is halted until the BAG ends and the sequence is then restarted with the next Sub-VF FIFO queue.

A Sub-VL FIFO queue should only be read by one VL FIFO queue. IP fragmentation should be performed at the IP layer, if it is needed. This will avoid for example short sampling messages being delayed by long queuing messages. The round robin continues in the presence of IP fragmentation so that one fragment is fetched from one Sub-VL, and then a frame or fragment is taken from the following Sub-VL.

Sub-VLs are below the IP layer.

3.3.2 Trivial File Transfer Protocol Example

This example describes the utilisation of TFTP to send a file from LRU 1 to LRU 2, as illustrated in Figure 3-28. For this, two VLs are defined: VL1 and VL2.

VL1: LRU1 to LRU2, VL2: LRU2 to LRU1

In the initialization phase:

1. The transfer is initiated by LRU 1, which sends a request from source port 45 000 on destination port 69, dedicated to TFTP in the LRU2.
2. LRU 2 activates a TFTP session, which responds to the request by sending a message to port 45000 of LRU 1. It indicates the port chosen to receive the transfer (Port 47 000).
3. At this moment the connection is established. The LRU 1 can send the file in data packets. Communication from LRU 1 to LRU 2 uses source port 45 000 and destination port 47 000.
4. The acknowledgement of each data packet is sent by LRU2 which uses source port 47 000 and for destination port 45 000.

3.0 END SYSTEM SPECIFICATION

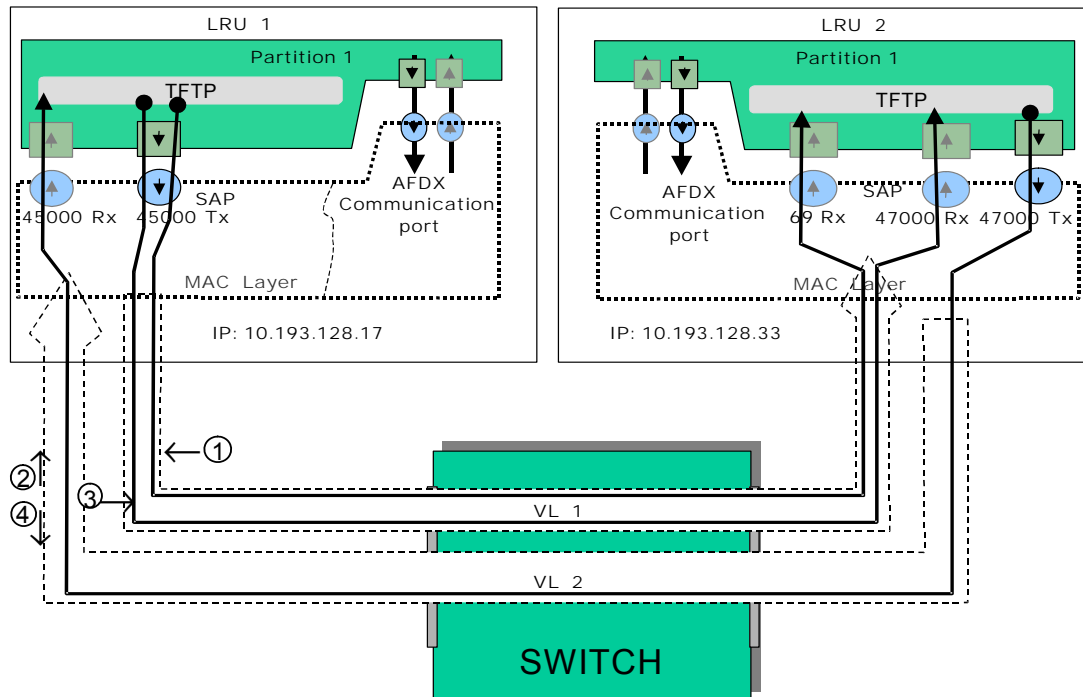


Figure 3-28 – Example of TFTP Communication in the AFDX Network

3.3.3 ES Communication Stack

Figure 3-29 illustrates the ES Communication Stack.

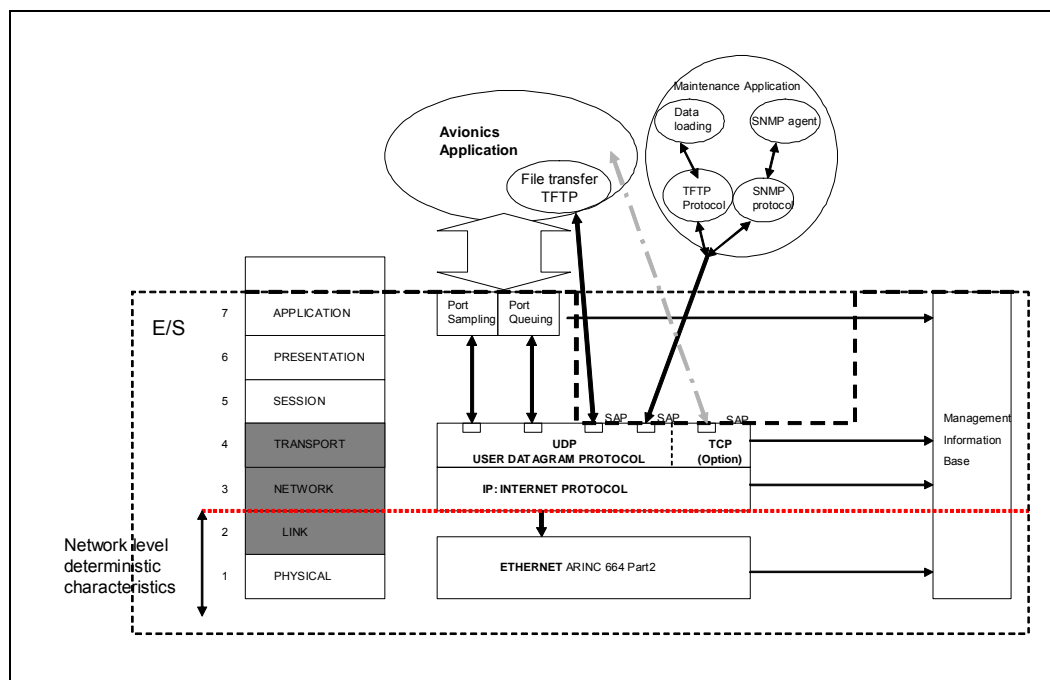


Figure 3-29 – ES Stack

3.0 END SYSTEM SPECIFICATION**3.3.3.1 ES MAC Profile**

The Data-link layer of the ES should be based on the use of Full-duplex Ethernet links as defined in IEEE 802.3 standard.

Any Ethernet Frame generated by the End System should be compliant with IEEE 802.3 standard.

All output interfaces will continue to transmit, even in the case of a physical layer link failure.

COMMENTARY

This avoids the sending of old buffered frames after a long link loss (following a switch reset or intermittent physical layer failures for instance). It may also help in avoiding propagation of a failure from the switch to the End System as well as between switches.

A maximum AFDX frame length should be defined on a per VL basis.

In reception, if the AFDX frame format and Frame Check Sequence (FCS) and Cyclic Redundancy Check (CRC) are valid (without preamble and Start Frame Delimiter fields) the frame should be forwarded to the upper layer.

3.3.3.2 ES IP Profile**3.3.3.2.1 IP Packet Structure**

The packet structure version should be IPv4.

IPv4 packet structure should be compliant with Figure 3-30.

4 bits	4 bits	8 bits	16 bits	16 bits	3 bits	13 bits	8 bits	8 bits	16 bits	32 bits	32 bits	1-1479 bytes
Version	IHL	Type of service	Total length	Fragment identification	Control flag	Fragment offset	Time to live	Protocol	Header checksum	IP Source address	IP Destination address	IP payload

Figure 3-30 – IPv4 Packet Structure

Ordinarily, in the IPv4 packet structure, the Total Length field should range from 21 to 1500 bytes. In AFDX, this range is from 21 to 1499 due to the Sequence Number (see Section 3.2.6.2.2, Redundancy Management Function).

COMMENTARY

The Total length field does not take into account the Sequence Number.

Refer to Attachment 2 for supplementary provisions to fulfill when implementing IP/ICMP, UDP, and TCP in the End System.

3.0 END SYSTEM SPECIFICATION

3.3.3.2.2 IP Fragmentation/Reassembly

Refer to Figure 3.4.1-1 in ARINC Specification 664, Part 3, for guidance in implementing IP fragmentation and reassembly for AFDX. Refer also to Table 3.4.1-2 IP Deviation Table for additional guidance.

IP fragmentation should occur on transmit when IP datagrams exceed the maximum payload size of the Ethernet frame as defined by the assigned MTU (maximum transfer unit). MTU is a configuration item that is maintained for each VL. The default value of MTU at 576 bytes is not used for AFDX.

The IP header for fragmentation should be created as defined in the RFCs to maintain interoperability with standard Ethernet/IP/UDP protocol stacks.

The IP layer of AFDX should support reassembly of IP datagrams. Out-of-order reassembly of IP datagrams is not supported (as defined in Table 3.4.1-2 item 1.2 in ARINC Specification 664, Part 3).

The IP layer of AFDX should support as many simultaneous reassemblies of IP datagrams on the receive side as there are subVLs per VL on the transmit side.

COMMENTARY

SubVLs are sometimes used as a means of separating periodic data flows from large block data transfers. Large block data transfers usually fragment at the IP layer. If a periodic message frame is placed in a subVL queue behind a large fragmented block of data, the periodic data flow will potentially be delayed by many BAG periods, inducing large jitter in the flow. SubVL queues are available to allow system integrators to configure transmit ports to use different subVL queues to isolate those periodic data flows from large messages that fragment. Therefore, an ES might expect to have as many reassemblies going on simultaneously as there are subVL queues at the transmitting ES, but never more.

3.4 Network Level Interoperability**3.4.1 Addressing****3.4.1.1 Introduction**

A data flow is uniquely identified within AFDX network by the set of UDP/TCP destination port, IP destination address, MAC destination address and the physical Ethernet connection(s) of the receiving ES.

Frame based filtering is done such that the receiving End System will only forward valid frames to a communication or a SAP port. Valid frame being defined by analysis of destination (TCP/UDP, IP, MAC) addresses and physical Ethernet connections.

3.0 END SYSTEM SPECIFICATION

3.4.1.2 Structure of an AFDX Frame Without Fragmentation

Figure 3-31 illustrates the structure of an AFDX frame. Minimum and maximum frames are illustrated

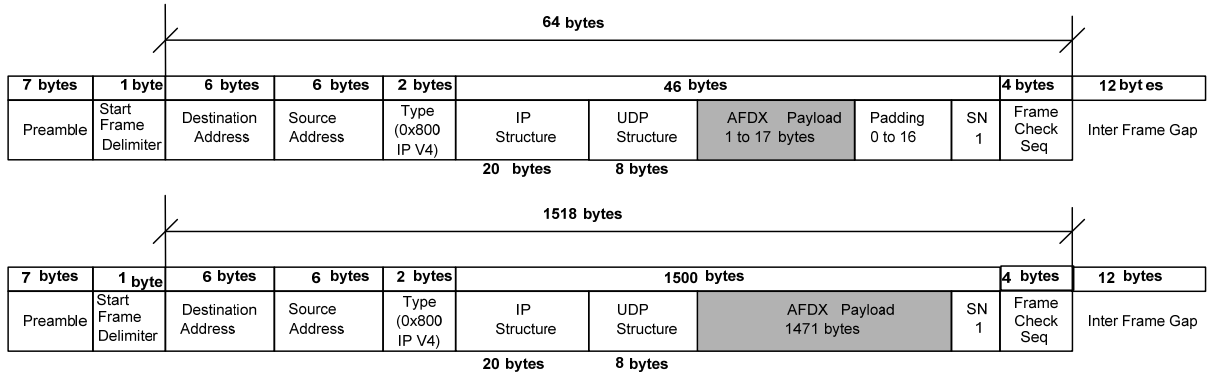


Figure 3-31 – Structure of an AFDX Frame

An example of the Addressing Principle is shown in Figure 3-33.

In Figure 3-32, End System 1 has three Virtual Links: VL1, VL2 and VL3.

Partition 1 of End System 1 has access to one Virtual Link: VL1

Partition 2 of End System 1 has access to two Virtual Links: VL2 and VL3.

3.0 END SYSTEM SPECIFICATION

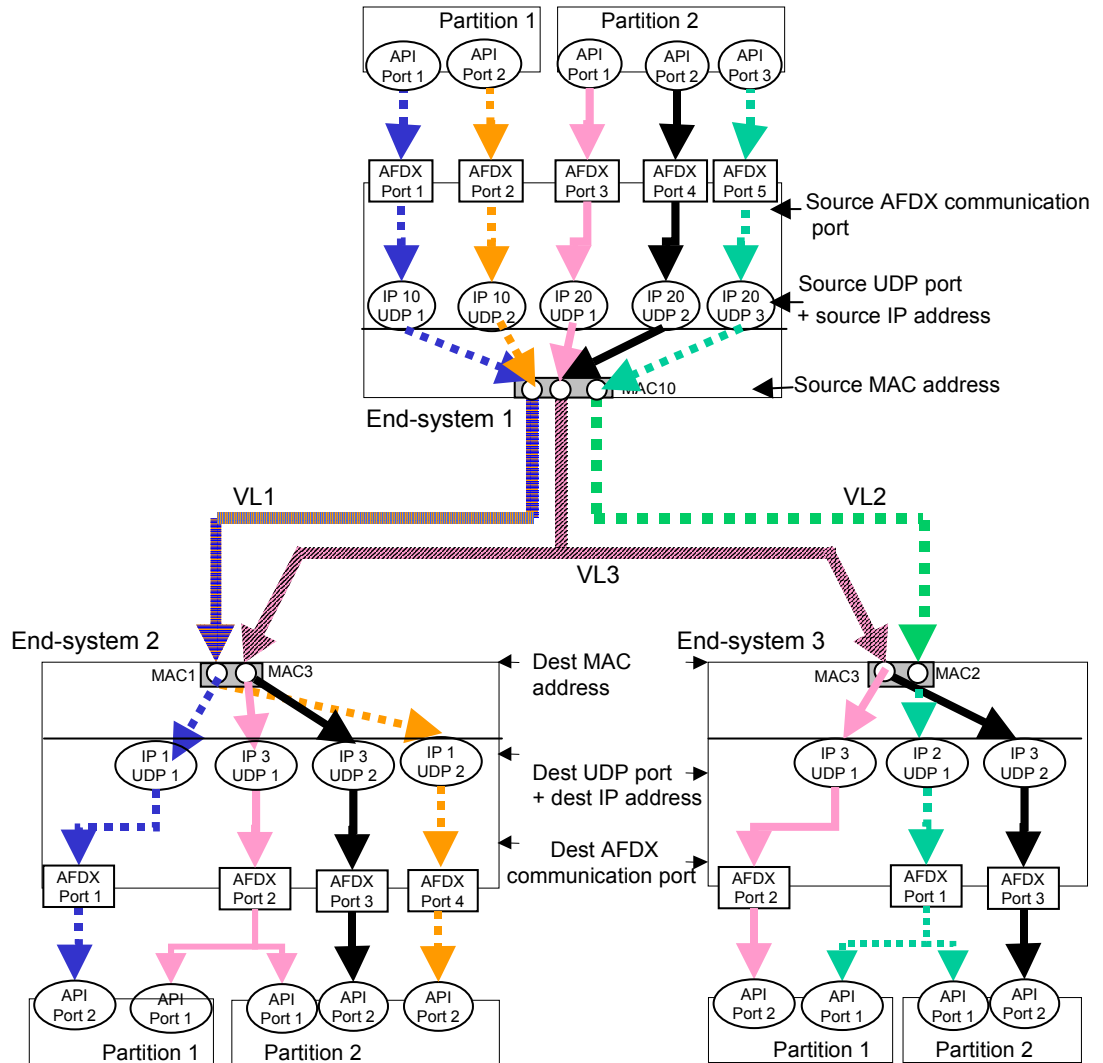


Figure 3-32 – Example of Addressing

3.0 END SYSTEM SPECIFICATION

The following tables show the address tables for each ES:

Transmit Table of End-System 1:

Src AFDX Communication Port	Src partition	Src UDP	Src IP	Src MAC	Dest UDP	Dest IP	Dest MAC
AFDX Port 1	Partition 1	UDP1	IP10	MAC10	UDP1	IP1	MAC1 (VL1)
AFDX Port 2	Partition 1	UDP2	IP10	MAC10	UDP2	IP1	MAC1 (VL1)
AFDX Port 3	Partition 2	UDP1	IP20	MAC10	UDP1	IP3	MAC3 (VL3)
AFDX Port 4	Partition 2	UDP2	IP20	MAC10	UDP2	IP3	MAC3 (VL3)
AFDX Port 5	Partition 2	UDP3	IP20	MAC10	UDP1	IP2	MAC2 (VL2)

Receive Table of End-System 2:

Dest AFDX Communication Port(s)	Dest partition(s)	Src UDP	Src IP	Src MAC	Dest UDP	Dest IP	Dest MAC
AFDX Port 1	Partition 1	UDP1	IP10	MAC10	UDP1	IP1	MAC1 (VL1)
AFDX Port 4	Partition 2	UDP2	IP10	MAC10	UDP2	IP1	MAC1 (VL1)
AFDX Port 2	Partition 1 and 2	UDP1	IP20	MAC10	UDP1	IP3	MAC3 (VL3)
AFDX Port 3	Partition 2	UDP2	IP20	MAC10	UDP2	IP3	MAC3 (VL3)

Receive Table of End-System 3:

Dest AFDX Communication Port(s)	Dest partition(s)	Src UDP	Src IP	Src MAC	Dest UDP	Dest IP	Dest MAC
AFDX Port 2	Partition 1	UDP1	IP20	MAC10	UDP1	IP3	MAC3 (VL3)
AFDX Port 3	Partition 2	UDP2	IP20	MAC10	UDP2	IP3	MAC3 (VL3)
AFDX Port 1	Partition 1 and 2	UDP3	IP20	MAC10	UDP1	IP2	MAC2 (VL2)

COMMENTARY

At the receiver, the demultiplexing is based only on the MAC, IP, UDP destination address.

Figure 3-33 presents the physical topology for this example:

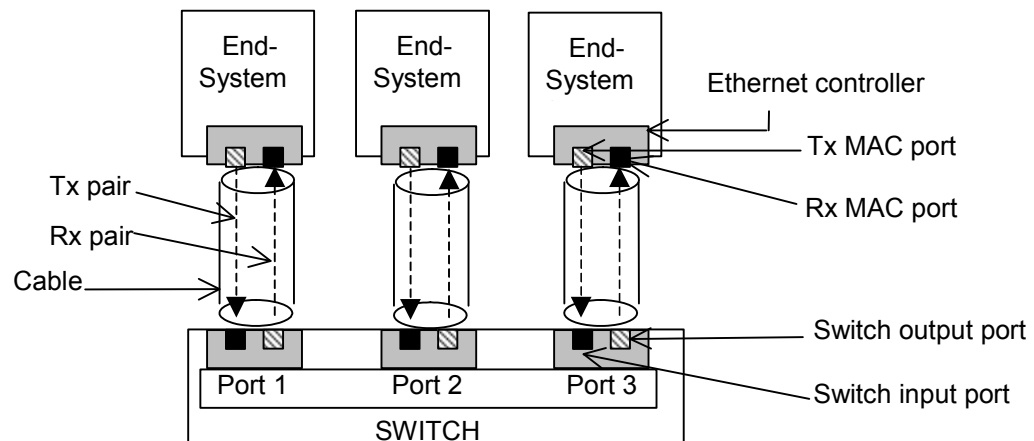


Figure 3-33 – Example Physical Topology

3.0 END SYSTEM SPECIFICATION

Switch Forwarding is defined in Table 3-2.

Table 3-2 – Switch Forwarding

Input port	MAC destination field of the received frames	Output ports
1	MAC1 (VL1)	2
1	MAC2 (VL2)	3
1	MAC3 (VL3)	2 and 3

COMMENTARY

MAC address should be understood as potentially unicast or multicast Ethernet address.

3.4.1.3 Identification for End-to-End Communication

Peer-to-peer communications are identified in each frame by UDP Source Port + Source IP + Destination MAC (VL identification) + Destination IP + UDP Destination Port as illustrated in Figure 3-34.

In the AFDX network, this quintuplet provides a unique identification for each message.

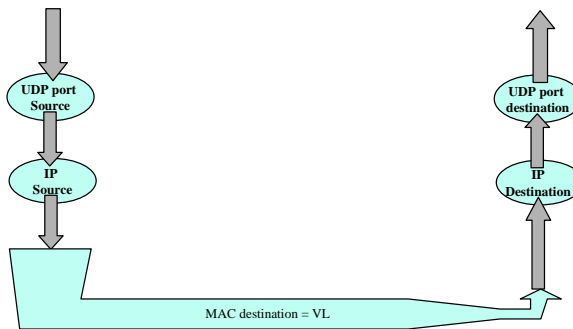


Figure3-34 – Message Identification Concept

For a source IP, there should be several source UDP/TCP ports. For a destination IP there should be several destination UDP/TCP ports.

In Figure 3-35, we have 3 messages identified by 3 quintuplets.

Message 1 => UDP Source Port x + Source IP + destination MAC + destination IP + UDP destination port n

Message 2 => UDP Source Port y + Source IP + destination MAC + destination IP + UDP destination port m

Message 3 => UDP Source Port z + Source IP + destination MAC + destination IP + UDP destination port v

3.0 END SYSTEM SPECIFICATION

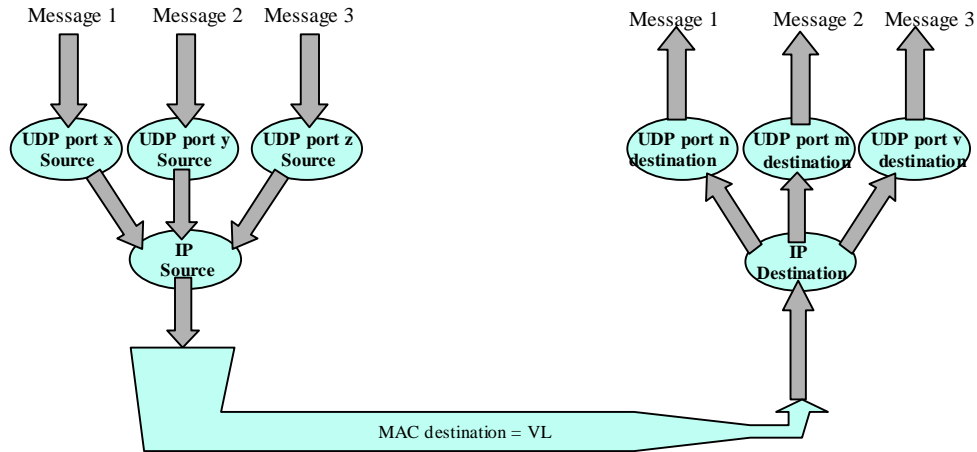


Figure 3-35 – Unique Message Identification within one Virtual Link

3.4.1.3.1 Intra-AFDX Communication

End-to-end communications, which remain within the AFDX network, can be regarded as Intra-AFDX.

The principal characteristic of Intra-AFDX communication is the fact that for each message, the addressing is statically defined.

For unidirectional communications:

- AFDX communication ports are defined through UDP ports. Such ports can be either transmitters or receivers.
- AFDX communication ports are characterized by the Sampling and the Queuing services.

For bi-directional communications:

- Use may be made of the TFTP protocol (or other protocols above UDP/TCP for future evolution). There are two possibilities:
 1. Utilization of the SAP (Service Access Point) ports. These are linked with UDP or TCP ports, and each SAP can be a transmitter or a receiver. To obtain a bi-directional communication, two SAPs should be used (e.g.: SAP 30 000 Tx and SAP 30 000 Rx).

In this case, two quintuplets are defined, one for each direction of communication.

It is also recommended that SAP ports be used for full compliance to Internet protocols: e.g. port 69 is used for TFTP.

2. Utilization of the conventional AFDX communication ports. Bi-directional communication, at a single ES should require two AFDX communication ports : one transmitter and one receiver, (e.g.: AFDX Com port 15 000 Tx and AFDX Com port 15 000 Rx).

For a bi-directional communication, the ports should be used in queuing mode.

3.0 END SYSTEM SPECIFICATION

3.4.1.3.2 Extra-AFDX Communication

This section describes communications between the AFDX network and a compliant network. Two modes of communication are defined as follows:

Unidirectional communications:

This will always be from a transmitting ES to a compliant network, and use can be made of the conventional AFDX communication port with a UDP port link. The implication is that the ES configuration table will contain the destination IP and port number. Hence there will be a statically defined quintuplet for addressing.

Bi-directional communications:

Can utilize TFTP, SNMP, 615A protocols or other protocols using UDP/TCP (future evolution).

SAP ports are used, and as for Intra-AFDX communications, each SAP is either a transmitter or a receiver. Two SAPs would be used to obtain a bi-directional communication.

The receiving SAP can pass to the partition the IP address and the UDP/TCP port identification of the sources in the compliant network.

The transmitting SAP can pass the IP address and the UDP/TCP port identification of the destination in the compliant network.

3.4.1.4 IP Addressing Format

The IP addresses used here should follow the allocations provided in ARINC Specification 664, Part 4. If an IP address translation mechanism is used, then the frames sent by an AFDX End System outside this network (to another Domain) should follow the Part 4 allocations.

3.4.1.4.1 IP Source Address

The IP source address should be used to identify the transmitting partition associated with the End System.

The IP destination address should be used by an End System to forward IP packets to one or more destination End System(s).

The IP address should be Class A and private Internet Unicast Address (First 8 bits should be '0000 1010').

The IP Unicast Addressing Format is shown in Figure 3-37.

COMMENTARY

See ARINC Specification 664, Part 4, for the appropriate address range to use within this “10-dot” range.

3.0 END SYSTEM SPECIFICATION

IP Unicast Addressing Format (source or unicast destination) 32-bits				
Class A 1-bit	Private IP address 7-bits	User_Defined_ID 16-bits	Partition ID	
"0"	"0001010"	"nnnn nnnn nnnn nnnn"	Spare field 3- bits	5-bits

Figure 3-36 – IP Unicast Addressing Format

The User_Defined_ID is a single 16-bit field. It should be used as the system integrator deems appropriate to give each IP addressable host on the network a unique and meaningful IP address.

The Partition_ID is comprised of two fields:

The Spare field is normally not used and set to zero as shown in Figure 3-37. These bits may be used for partition ID if a system has more than 32 partitions.

Spare Field	Meaning
0 0 0	No significance

Figure 3-37 – Spare Field Definition

The IP source address in the IP header of the AFDX frame should be an IP unicast address used to identify the transmitter.

3.4.1.4.2 IP Destination Address

The IP destination address in the IP header of the AFDX frame should be:

- Either the IP Unicast address to identify the target subscriber
- Or an IP Multicast address compliant to the format shown in Figure 3-38

IP Addressing Format 32 bits		
4 bits	28 bits	
Class D "1110"	IP Multicast Identifier	
	Constant field 12 bits = "0000 1110 0000"	Virtual Link Identifier 16 bits

Figure 3-38 – IP Multicast Addressing Format

3.4.1.5 AFDX Communication Port, SAP and UDP/TCP Addressing Format

For Intra and Extra AFDX communications, there are two interfaces between the End System and the partitions: AFDX communication and SAP ports.

The AFDX Communication port, illustrated in Figure 3-39, is characterised by:

- Unidirectional access: Transmission (Tx) or reception (Rx)
- Sampling or queuing mode: sampling and queuing have signification only in reception

3.0 END SYSTEM SPECIFICATION

- In transmission, there is only one link between "AFDX Communication port" and the quintuplet (UDP Source Port, Source IP, destination Mac, destination IP, UDP destination port). The "AFDX Communication port" belongs to an unique partition.
- In reception, there is only one link between "AFDX Communication port" and the quintuplet (UDP Source Port, Source IP, destination Mac, destination IP, UDP destination port). The "AFDX Communication port" may be accessed by different partitions.
- The transmission and the reception path is frozen by configuration, it can be represented by the following figure:

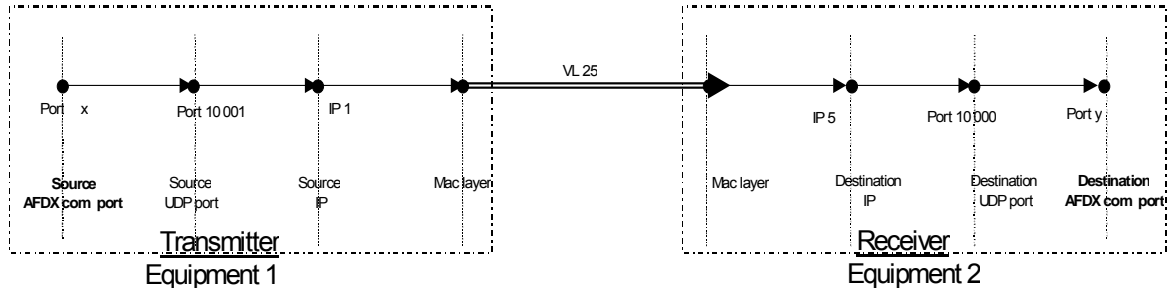


Figure 3-39 – AFDX Communication Port

The SAP port, illustrated in Figure 3-40, is characterized by:

- A SAP port is mapped to a UDP (or TCP for future extension) port, this term is used to differentiate them from the "AFDX communication ports."
- Unidirectional access: Transmission or reception.
- The possible coupling of two SAP ports to identify bi-directional communication e.g. Port 500 TX and 500 Rx.
- In transmission, the SAP port uses by configuration a frozen quadruplet (UDP Port source address, IP source address, Mac Source address, MAC destination address (VL identification)), the destination IP address and destination UDP (or TCP) port are given by the partition.

In reception the SAP is linked only to one destination Port + destination IP + destination Mac (VL identification) + source MAC. The IP source address and UDP (or TCP) source port are delivered by the End System to the partition.

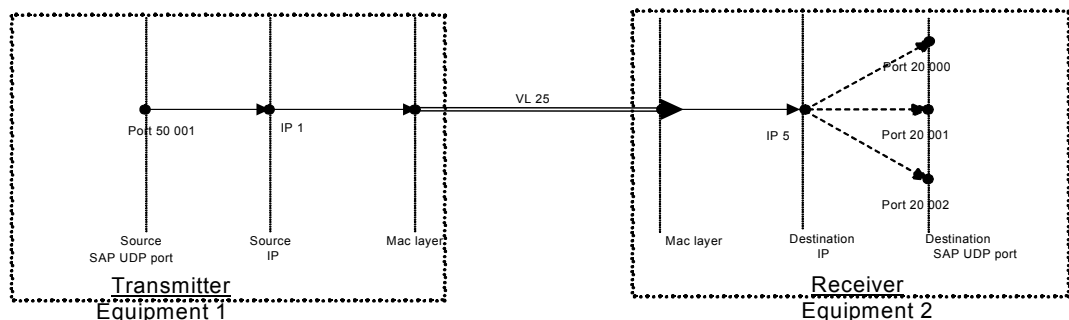


Figure 3-40 – SAP Port Using UDP

3.0 END SYSTEM SPECIFICATION

3.4.1.5.1 AFDX Communication Ports

In transmission, an AFDX Communication Port should only be linked to a single set UDP port source, IP source, VL (MAC destination), IP destination and UDP destination.

In reception, an AFDX Communication Port should only be linked to a single set (unique UDP port destination, IP destination, VL (MAC destination)) and additionally to the Ethernet physical interface if the redundancy management is **disabled**.

3.4.1.5.2 SAP Ports

In transmission, a SAP Port should only be linked to a single set (UDP port source, IP source and VL (MAC destination)).

In reception, a SAP Port should only be linked to a single set (UDP/TCP port destination, IP destination and VL (MAC destination))

The UDP/TCP port number should identify the Service Access Point.

In reception, the Service Access Point should make available the IP source and the UDP/TCP Source to the receiving partition.

In transmission, the Service Access Point should permit the partition to specify the IP address and the UDP/TCP port of the destination partition.

3.4.1.5.3 Allocation of the SAP and AFDX Communication Port Numbers

Port number allocation is defined in **ARINC Specification 664: Aircraft Data Network (ADN), Part 4 - Internet-Based Address Structures and Assigned Numbers**. Figure 3-41 gives this allocation and the choice for the AFDX.

Port range (decimal value)	Allocation range ARINC 664	Allocation range AFDX
0 – 1023	Administered by ICANN "Well-known" port number	Administered by ICANN "Well-known" port number
1 024 – 16 383	Registered by ICANN A664 assigned	Assigned by network manager
16 384 – 32 767	Registered by ICANN System integrator Or User defined	
32 768 – 65 535	Registered by ICANN Recommended for temporary port assignment	

Figure 3-41 – Allocation of SAP and AFDX Port Numbers

3.0 END SYSTEM SPECIFICATION

For each IP unicast or multicast IP, the repartition of the port allocation range is given in Figure 3-42.

Type of port	Type of communication	Port range	Commentary
AFDX Communication port	AFDX ⇔ AFDX AFDX ⇔ Compliant network	1 024 – 65 535	Used for sampling and queuing communications
SAP	AFDX ⇔ AFDX AFDX ⇔ Compliant network	0 – 1023	Used for standard communications e.g Port 69 to open a TFTP, Data loading (ARINC 615A), SNMP, etc..
	AFDX ⇔ AFDX AFDX ⇔ Compliant network	1 024 – 65 535	Used for bi-directional communication: specific TFTP etc..

Figure 3-42 – Port Allocation Range for IP Unicast or Multicast

COMMENTARY

Port number duplication: the only technical limitation regarding port number duplication is that two identical port number cannot be used with in a Virtual Link. Nevertheless, a system integrator may add some rules in order to make a UDP port number unique within the network, or on the contrary to impose the use of the same port number for bi-directional communications.

4.0 SWITCH SPECIFICATION

4.1 Basic Concepts

The switch consists of five functional blocks that interact with each other, as shown in Figure 4-1.

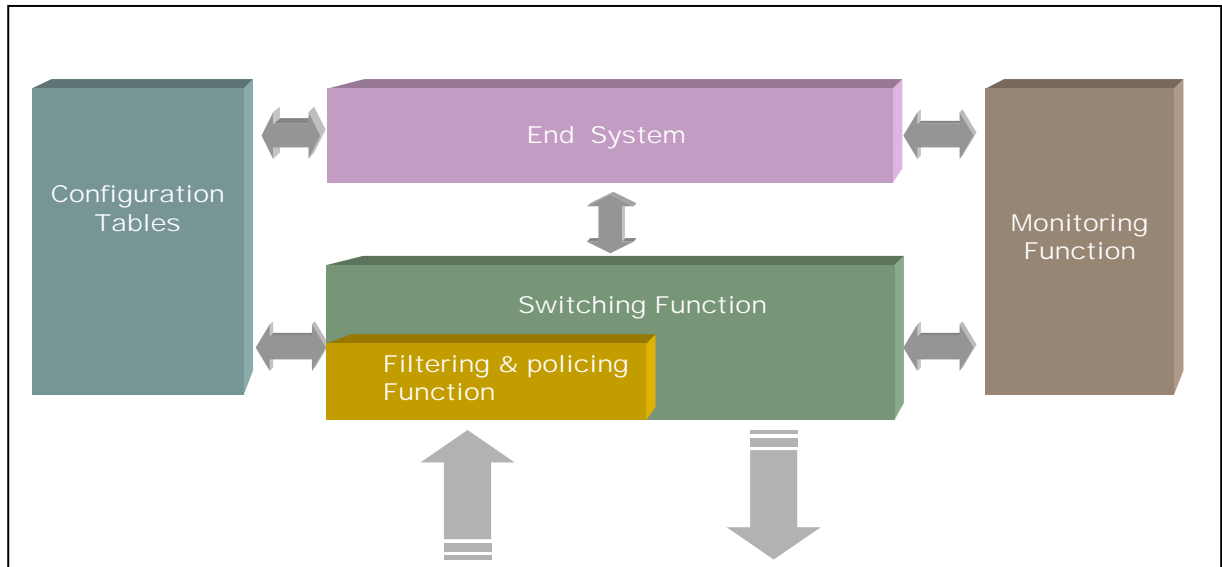


Figure 4-1 – Main Functional Blocks of the AFDX Switch

All frames arrive at the switch in the Filtering & Policing function stage where they are filtered in various steps

that apply rules about frame integrity, frame length, traffic budget and acceptable destination(s).

The core of the switching activity is performed by the Switching function. Frames filtered by the Filtering and Policing function are forwarded to the appropriate physical output ports where they leave the switch again.

These functions are controlled by configuration data contained in static Configuration Tables.

The End System stage provides the means to communicate with the Switch (receives frames dedicated to the Switch and allows the Switch to send frames). This is used for Data Loading and monitoring functions.

All operations are monitored by a Monitoring function that logs events such as the arrival of a frame or a failed CRC check and additionally creates statistics about the internal situation. Since the switch is part of a network, it communicates with the Network Management Function for operational information and for health related information.

4.0 SWITCH SPECIFICATION

4.1.1 Filtering and Policing Function Introduction

4.1.1.1 Policing and Filtering Parameters

In this section it is assumed that the Virtual Link number i (VLI) has the following characteristics:

- The BAG (Bandwidth Allocation Gap) associated **with** VLI is BAG $_i$ (expressed in seconds)
- The Jitter (representing the Window a frame is assured to be located within) associated **with** VLI entering a particular switch is J $_{i,switch}$ (expressed in seconds)
- The maximum frame size for VLI is denoted S_i^{max} (expressed in bytes). S_i^{max} is a configurable parameter that should be in the range [84, 1538]
- The minimum frame size for VLI is denoted S_i^{min} (expressed in bytes). S_i^{min} is a configurable parameter that should be in the range [84, 1538]. S_i^{min} should be less than or equal to S_i^{max}

The frame size value shown in Figure 4-2 corresponds to the actual time a frame occupies the Ethernet line (s). Therefore, all fields have to be taken into account: IFG (12 octets) + Preamble (7 octets) + SFD (1 octet) + MAC Frame size ($L = 64$ to 1518 octets).

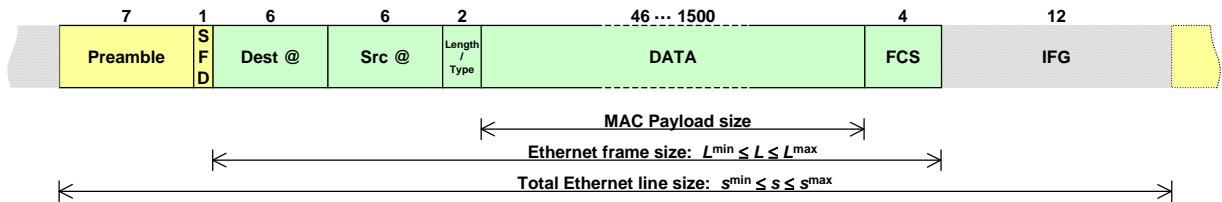


Figure 4-2 – Frame Size Values

4.1.1.2 Frame Filtering

The filtering causes the switch to distribute only valid frames to selected destinations. Upon arrival in a switch, each frame is examined and the contents of certain fields of the frame header (e.g. destination address field, frame check sequence field, etc.) and the construction of the frame itself are monitored:

- The frame size: whether the frame is either longer or shorter than the envelope allows
- The frame integrity: whether the FCS embedded in the frame matches the calculation upon reception
- The frame path: whether the destination requested by the content of the Destination Address field (which in case of the AFDX is the Virtual Link Identifier) of the arriving frame is permitted or not

If the frame properties do not comply with the configuration parameters, the frame is filtered i.e. discarded, and one or more MIB entries updated. The definition of the MIB entries as well as the update conditions and processes are specified in the relevant specification document.

4.0 SWITCH SPECIFICATION

The following aspects of the frame are tested as part of the filtering function:

- Destination address validity (Ethernet Address corresponds to a valid VL, including constant field)
- This VL is valid to be received on that destination port (according to the switch configuration table)
- Frame Check Sequence validity
- Ethernet frame size (L) is an integral number of octets (alignment)
- Ethernet frame size (L) in the range [64 octets, 1518 octets]
- Ethernet frame size (L) less than or equal to Lmax
- Ethernet frame size (S) greater than or equal to Smin, only in case where byte-based traffic policing is used

COMMENTARY

Determinism proof, with byte-based traffic policing algorithm, is based on Smin parameter. In order to insure that this algorithm is always verified, in particular in the case of ES dysfunction, this minimum size has to be respected, and though any frame smaller than Smin for a given VL has to be discarded.

In our case, the length/type field is used as a type field and no frame length consistency test is made based on this field.

The definition of a “valid frame” might be more restrictive in an aeronautical environment than in a commercial one. The term used in the context of networking for enforcement is “policing.”

4.1.1.3 Traffic Policing

This section describes a model of an algorithm that performs traffic based policing on the Destination Address basis. The Destination Address field contains the information that is used to identify a Virtual Link in an AFDX environment. A Virtual Link defines a traffic flow that has certain properties such as a group of recipients, or a minimum allowed gap between two frames. This traffic flow has to be maintained in a segregated way to guarantee its associated properties. In order to use in the context of this specification the same terminology as the one used in commercial documents the Destination Address is used synonymously with the term Virtual Link or “VL.”

The jitter phenomenon is dependent upon Virtual Link and switch characteristics, as it is a function of the traffic of the total of all Virtual Links arriving at a particular switch. If a Virtual Link spans several switches, the actual jitter may be different on each of the intermediate switch. However, maximum values for latency and jitter for any switch provide an upper bound.

The traffic-policing model is described from an End-System point of view because End-Systems are the main traffic generators into the switch, shown in Figure 4-3. Model based description of the switch provides a switch centric view of the properties expected from a proper implementation of the switch.

4.0 SWITCH SPECIFICATION

Traffic policing may be implemented in two different algorithm, byte-based policing or frame-based policing, according to the mathematical method used to prove the determinism.

- The byte-based traffic policing filters out the VL in terms of bandwidth usage expressed in bits per second
- The frame-based traffic policing filters out the VL in terms of bandwidth usage expressed in frames per second

The switch may implement one of the two algorithm, either Byte-based or Frame-based, or both of them. The choice of the algorithm will have an impact on the method used to prove the schedulability of the network.

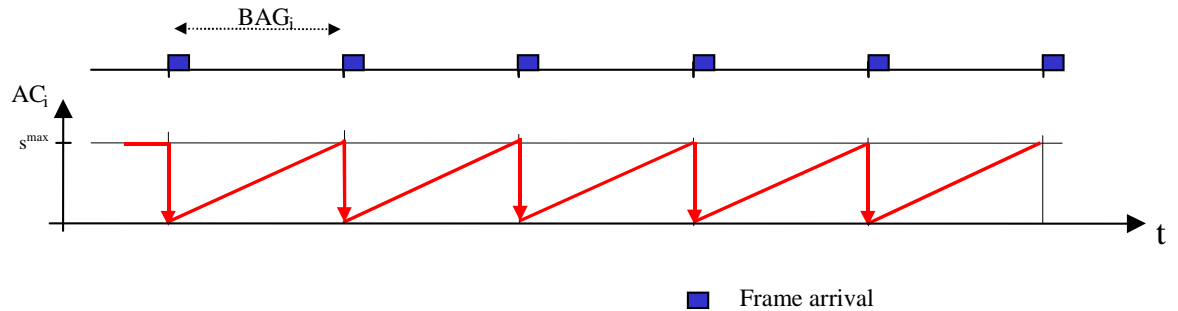


Figure 4-3 – Example of Traffic Without Jitter

Description of the policing algorithm:

Initially, the ACcount for VLi (also called ACi, expressed in bytes) is set to

$$s_i^{\max} \cdot \left(1 + \frac{J_{i,switch}}{BAG_i} \right)$$

ACi is credited as time elapses, proportionally to $\frac{s_i^{\max}}{BAG_i}$ with an upper limit of

$$s_i^{\max} \cdot \left(1 + \frac{J_{i,switch}}{BAG_i} \right)$$

- ACi is checked every time a frame of VLi arrives in the switch.
- Let s be the total Ethernet line size of the received frame (S = Ethernet frame size (L) + 20 octets; the 20 octets correspond to IFG+Preamble+SFD).

COMMENTARY

If $J_{i,switch}=0$, then ACi is set initially to s_i^{\max} (see Figure 4-4).

Two accounts are managed: Byte ACi (Byte based Filtering) and Frame ACi (frame-based filtering).

4.0 SWITCH SPECIFICATION

For byte-based traffic policing:

- If Byte ACi is greater than S, the frame is accepted and Byte ACi is debited by S
- If Byte ACi is less than S, the frame is discarded, the appropriate MIB entry is updated and Byte ACi is not changed

For frame-based traffic policing

- If Frame ACi is greater than S_i^{\max} , the frame is accepted and Frame ACi is debited by S_i^{\max}
- If Frame ACi is less than S_i^{\max} , the frame is discarded, the appropriate MIB entry is updated and Frame ACi is not changed

This traffic policing principle is known in the literature as the "token bucket" algorithm.

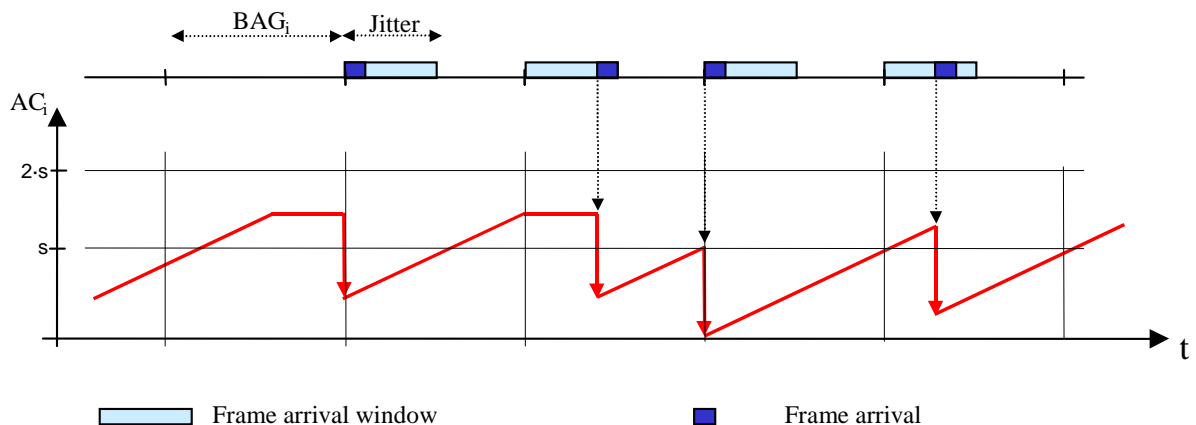


Figure 4-4 – Example of Traffic with Jitter = BAG/2

A traffic policing mechanism should be implemented on the switch in order to ensure fault containment function of the network. Since a failed End System must not disturb the network, any frame belonging to a traffic flow that is not compliant with the network configuration should be discarded.

4.2 Filtering and Policing Function

4.2.1 Frame Filtering

Budget Accounts (Bytes ACi and Frame ACi) should not be consumed by invalid frames, i.e. frames not forwarded by the forwarding function.

No budget should be consumed by frames that are discarded by the filtering mechanisms.

COMMENTARY

If the traffic based filtering process were to consume budget for discarded frames, it would not pass a frame until the minimum time period had elapsed since the previous discarded frame rather than

4.0 SWITCH SPECIFICATION

the previous transmitted frame. This would effectively block a poorly spaced sequence of frames rather than limiting the flow to the assigned budget. Note: The purpose of the filtering is to limit the flow to an assigned budget level. It is not the intent to block the flow entirely in case of budgetary non-compliance.

In order to guarantee total network robustness, it is important that only uncorrupted frames be forwarded. Therefore, upon arrival, the switch should test each frame by use of the Frame Check Sequence (CRC) field according to IEEE Std 802.3. The Switch should discard Frames that do not satisfy this test.

The switch should discard incoming Ethernet frames having an Ethernet frame size (L) greater than 1518 or less than 64 bytes.

The switch should discard incoming frames having an Ethernet frame size that is not an integral number of octets (alignment errors).

The switch should discard incoming frames which total Ethernet line size is greater than the maximum size (Lmax) allowed for the relevant VL, as such a frame would consume more bandwidth than allocated.

In the case where byte-based policing is used, the switch should also discard incoming frames which total Ethernet line size is smaller than the minimum size (Smin) allowed for the relevant VL.

The switch should discard incoming frames with a MAC destination address that has an erroneous 32-bit constant field.

The switch should discard incoming frames for which the VL id is not allowed on the incoming port.

4.2.2 Traffic Policing

The policing algorithm is based on the management of an account assigned to each VL.

The switch should use the VL identifier to obtain the appropriate information of filtering and policing from the configuration table. Each received frame is filtered and policed with respect to the information stored in the configuration. The relevant parameters are accessed through the VL id.

Two types of relationship between MAC destination address and ACi should be supported:

- One ACi for only one MAC destination address (VLi)
- One ACi for a group (several) of MAC destination addresses (VLs): Account Sharing

A shared VL will be part of one and only one account.

COMMENTARY

Partitions may benefit from such an account sharing capability, especially if loss of frames could be accepted. The capability would

4.0 SWITCH SPECIFICATION

compromise the segregation between the respective VLs, as well as eliminate a bandwidth guarantee.

For safety reasons there is a fixed relationship between ONE Destination Address and ONE account preferred. However, it is at the discretion of the network administrator and/or the System designer to use the account sharing features.

For example, several LRU's may use account sharing for VLs used for Data Loading from the LRU to the Data Loader. It is very improbable that all LRU's will be data loaded at the same time. Therefore defining a VL from each LRU to the Data loader would be a waste of bandwidth. Moreover, the recovery mechanisms embedded in ARINC 615A make a limited loss of frames acceptable.

Account sharing is a function used only within the switch and does not impact LRU-ES configuration or functionality.

The Configuration table has a relationship between MAC destination address, AC_i, BAG, Jitter, Smax, and eventually Smin in the case where byte-based traffic policing is used.

Traffic policing should be based on parameters: BAG, Jitter, Smax, and eventually Smin in the case where byte-based traffic policing is used. It should mechanize the algorithm described in paragraph “filtering and policing function introduction.”

For each VL or group of VLs sharing the same account, the Traffic Policing function should authorize one BAG value according to the configuration table.

The Traffic Policing function of the Switch should at least be configurable for BAG values in the range 1ms to 128 ms.

For each VL or group of VLs sharing the same account, the Traffic Policing function should authorize one maximum Jitter value, according to the configuration table.

The traffic policing function should at a minimum be configurable for maximum allowed Jitter values in the range 0 to 10 milliseconds.

This implies that AC_i has a maximum size of at least:

$$s_i^{\max} \cdot \left(1 + \frac{10}{\text{BAG}_i} \right)$$

The traffic policing function should be able to handle at least Ethernet frames sizes (L) in the range [64-1518] octets.

COMMENTARY

This range corresponds to the Ethernet Frame sizes (L): it does not include the 20 octets of IFG+preamble+SFD.

There should be one BAG value per VL, one jitter value per VL, and one maximum size value per VL.

4.0 SWITCH SPECIFICATION

When one ACi is used by a group of VL, all the VL of this group should have the same BAG and the same minimum (smin) and maximum (smax) total Ethernet line size values.

When one ACi is used by a group of VLs, the traffic policing function should use the maximum of the jitter values of all of the VLs in the group as the jitter parameter for the ACi.

Traffic should be policed with a resolution of less than or equal to 100 μ s.

COMMENTARY

This gives a boundary regarding the time granularity of the policing function and thus limits the effect of a not perfectly accurate implementation.

Traffic should be policed with a relative tolerance on the resolution of $\pm 10^{-4}$.

COMMENTARY

This specifies the accuracy of the switch policing rate with respect to the configured value.

COMMENTARY

The two previous requirements have no impact on interoperability. From this perspective, they may be ignored. Nevertheless, the designer should consider these two parameters as they may have an impact on the global network performance (delay, determinism).

4.4 Switching Function

This section addresses the switching function of the Switch.

The order of input and output frames belonging to a Virtual Link should be maintained by the switch.

COMMENTARY

AFDX users expect to receive frames in the same order they are sent (VL ordinal integrity).

The switch should not modify the Frame Check Sequence of incoming frames. Since switches as network elements should be covered by the frame integrity mechanism embedded in the frame structure the CRC may not be regenerated within the switch under any circumstance, including at re-transmission.

For each frame, the content of the Destination Address field should be used to obtain from the configuration table the appropriate port(s) to which the frame has to be forwarded.

If an output port cannot accept a frame due to buffer congestion, this frame should be discarded for this port.

4.0 SWITCH SPECIFICATION

COMMENTARY

This is one of the most critical functions of the switching engine. It has to be guaranteed that the switching engine continues to cycle, no matter what happens to a given output port.

In the case of a link failure of a port, frames forwarded to and held in buffers of this port should be discarded for this port.

An output port should not transmit frames that are older than "max delay." The "max delay" parameter is defined on a per port basis in the configuration table.

The maximum delay parameter of a frame on a given port is defined as the maximum elapsed time between the two following events:

- Arrival of the last bit of a frame on the input port of a switch
- Exit of this last bit of the frame from the given output port of the switch

The switch should have the capability to receive a frame on any port and forward it to any combination of ports (including the receiving port). For example, Data exchanged between partitions hosted on the same equipment can be transmitted on the AFDX network to increase observability and portability.

The Switch should have a traffic prioritization mechanism based on MAC destination address with 2 traffic classes: High Priority and Low Priority. The priority level should be defined in the configuration table on a Virtual Link basis.

For each output port, frames with a priority set to high should be sent prior to the one set to low.

A low priority frame in the process of being transmitted should not be pre-empted by the arrival of a high priority frame.

If frames are present at the output port, switch should send them with a minimum Inter Frame Gap.

4.5 Switch End System Function

4.5.1 Overview

The End System of the Switch should comply with all requirements of Section 3.0, except the one related to network redundancy.

4.5.2 Addressing Policy

The End System of the switch should use its own MAC unicast address as the MAC Source Address when it sends frames.

4.6 Monitoring Function

The AFDX monitoring is based on the following:

- The MIB (Management Information Base) implemented in each AFDX component (equipment, subscriber and Switch) to store information about the components.

4.0 SWITCH SPECIFICATION

- The SNMP agent implemented in each AFDX component (equipment, subscribers and Switch) to communicate with the Network Management Function using SNMP.
- The Network Management Function, that realizes the correlation of information collected from all the components to detect/localize failure and to analyze network performances.

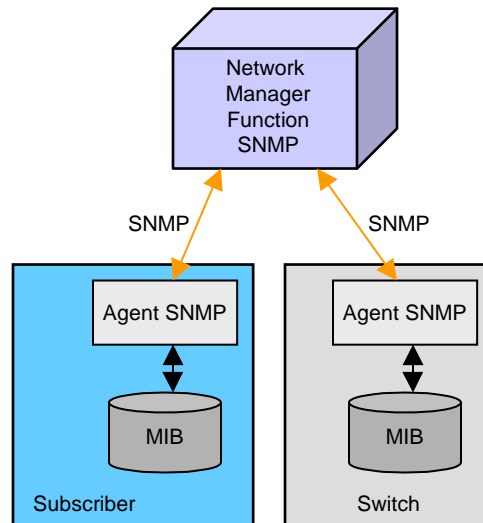


Figure 4-5 – Avionics Data Communication Network Monitoring Overview

The Switch contains an AFDX End System which is an AFDX subscriber. The Switch should implement a MIB and an SNMP agent. Additionally, the MIB of the Switch contains unique MIB objects related to the functions of the Switch.

A fault detection should modify the status object of the MIB managed by the switch.

The Fault/Healthy MIB variable should be updated every 100 ms (like the other MIB variables). This will ensure that when the Fault/Healthy indicator is sent it will always get the current status of the Switch.

COMMENTARY

The data recovered through an SNMP request should reflect the current status of the switch.

The switch should have a Fault/Healthy indicator. The Healthy/Fault indicator should be set to "Fault" when at least one used Ethernet port is in fault. The update of the Fault/Healthy indicator output should be performed at least every 100 ms.

COMMENTARY

This Fault/healthy indicator could be sent to the Flight Warning System. The implementation of this Healthy/fault indicator (e.g. discrete output) should be defined in an ARINC Specification.

4.0 SWITCH SPECIFICATION

4.7 Configuration Files

4.7.1 Introduction

Each switch, illustrated in Figure 4-6, owns at least 2 configuration files: Default and OPS. Depending on the operating mode, the appropriate file is selected in an exclusive mode. All files have to be identical for all switches within a network. Characteristics of the files:

- The Default_Configuration_File, corresponding to the resident configuration, that is used when the Switch is empty (no field loadable software yet loaded) or when the Switch is being dataloaded (in both case, the Switch is in Dataloading mode (DL)).
- The OPS_Configuration_File, corresponding to the loadable configuration, that is used when the Switch is in Operational mode (OPS).

COMMENTARY

Additional configuration files may be defined (example shop file, used for maintenance and production tests).

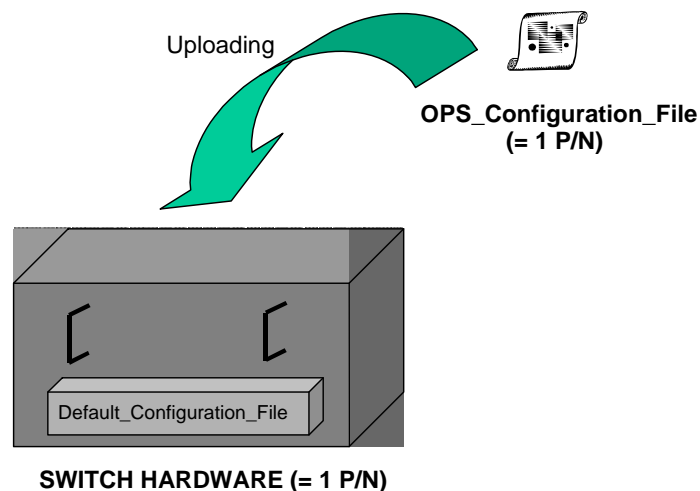


Figure 4-6 – Configuration Files Overview

Each OPS_Configuration_File contains several tables. The switch will extract the relevant table according to switch position. This switch position will be identified by a mean that could be defined in an ARINC Standard. In this way, the addressing to and from a switch is unique. The switch uses parameters contained in Configuration Tables in order to perform:

- Filtering and Policing Function
- Switching Function
- End System Function

In the following paragraphs, it is assumed that the switch position is identified by 12 pins. This should not be considered as normative, but as an example. Nevertheless, if the implementer decides to use 12 pins, the following definitions should be used.

4.0 SWITCH SPECIFICATION

4.7.2 Default_Configuration_Table

The Default_Configuration_Table is inseparable to the Hardware P/N of the Switch and can only be changed in shop, associated with a Hardware P/N modification.

The Default_Configuration_Table is used to define the default behavior of the End System of the Switch: default reception and default transmission. This table should be resident.

4.7.2.1 Default Physical Port

Port # 1 is the default external physical port used to communicate with the End System of the Switch. The default port speed should be set to 100 Mbps, without auto negotiation. This default ports allows the End System of the Switch to receive and to send frames even when it is not yet loaded with its loadable configuration.

4.7.2.2 Default Reception Configuration

Depending of its position in the aircraft network, acquired by pin programming, the Switch subscribes by default to one unique reception VL, further called VL_(0, position).

Justification of the name VL_(0, position):

- "0" indicates that it is a reception VL
- "Position" to indicate that this VL depends on the position of the Switch in the aircraft

The reception part of the Default_Configuration_Table of the Switch should contain at least, in nonvolatile memory, the following information (characteristics of the default reception VL):

- VL identifiers (MAC destination address): VL_(0,position)
- Maximum allowed Total Ethernet Line size (s^{\max})
- Bandwidth Allocation Gap

Default reception VLs are necessary to be able to communicate to the own End System of the Switch, even if it is not loaded with its OPS_Configuration_File (at least to perform dataloading).

This default VL is used to communicate with the Network Management Function and the Data-Loader. The frame size corresponds to the Total Ethernet Line size (including all protocol overheads, IFG, Preamble, and SFD).

The End System Function of the Switch will only take into account the frames whose IP destination field corresponds to its own IP address, deduced from Pin Programming P₁ to P₁₂ (refer to Section 4.10.2.1). The policing function of the switch is activated on the default VL, according to the values defined in the configuration file.

4.7.2.3 Default Transmission Configuration

Each Switch has a unique default transmission VL, further called VL_(1, position). The identifier of this VL depends on the position of the Switch in the aircraft, as defined in the Default_Configuration_File and specified in the default Data Load VL section.

- "1" indicates that it is a transmission VL
- "Position" to indicate that this VL depends on the position of the Switch in the aircraft

- VL identifier (MAC destination address): VL_(1,position) complying with the characteristics shown in Figure 4-7

[illegible]

Where P1 to P12 correspond to the following pin programming values: 1 = GROUND and 0 = OPEN.

- Bandwidth Allocation GAP
- Maximum allowed Total Ethernet Line size (s^{\max})

When the End System of the Switch will send frame, the IP and MAC source addresses fields in the AFDX frame will be the own Switch IP and MAC addresses, deduced from Pin Programming P1 to P12 (refer to Section 4.10.2.1).

The same OPS_Configuration_File (same P/N) is loaded in all the Switches of the Avionics Data Communications Network. The OPS_Configuration_File, illustrated in Figure 4-8, contains the operational configuration for all the Switches. Depending of its current position, acquired by pin programming, the Switch only has access to the dedicated table of the OPS_Configuration_File.

4.0 SWITCH SPECIFICATION

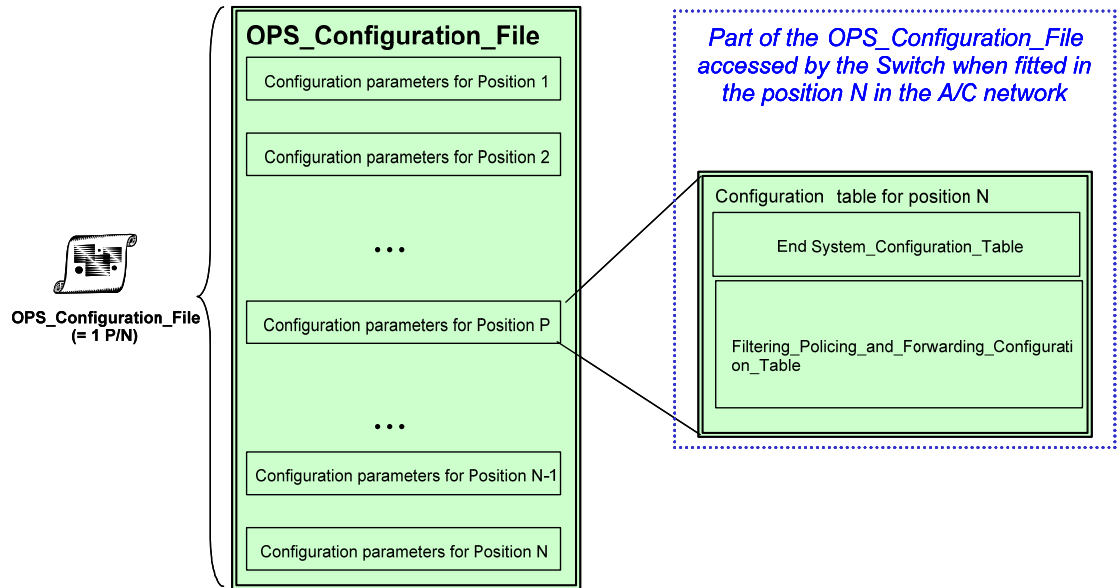


Figure 4-8 – OPS_Configuration_File Overview

Each of the tables of the OPS_Configuration_File contains:

- The EndSystem_Configuration_Table
- The Filtering_Policing_and_Forwarding_Configuration_Table

The OPS_Configuration_File should be a field loadable software file compliant with ARINC 615A and ARINC 665.

4.7.3.1 EndSystem_Configuration_Table

In OPS, the End System function of the Switch uses the EndSystem_Configuration_Table, corresponding to the current position of the Switch, in order to:

- On the receiving side, check the consistency of the received frame
- On the transmitting side, construct the Ethernet frames and perform the flow control mechanism

4.7.3.2 Filtering_Policing and_Forwarding_Configuration_Table

In Operational mode (OPS), the filtering and forwarding functions rely on the parameters contained in the Filtering_Policing_and_Forwarding_Configuration_Table, corresponding to the current position of the Switch.

The Filtering_Policing_and_Forwarding_Configuration_Table should contain the following parameters.

4.0 SWITCH SPECIFICATION

On a per VL basis:

- Input physical port
- List of output physical ports
- MAC destination address (VL identifier VLi)
- Bandwidth Allocation Gap (BAGi)
- Maximum allowed Jitter
- ACcount (ACi) [specifying if this account is shared or not]
- Maximum allowed Total Ethernet Line size (Smax)
- Minimum allowed Total Ethernet Line size (Smin)
- Prioritization (high or low)

On a per port basis:

- Max delay
- Port State (ON/ OFF)
- Speed of the physical port
- Output Buffer Size for Low Priority VLs
- Output Buffer Size for High Priority VLs

Those parameters are required to configure the Filtering, Policing and the Forwarding functions of the Switch.

4.8 Operating Modes

4.8.1 Overview

Figure 4-9 represents different operating modes of the Switch and the conditions of transitions between these modes.

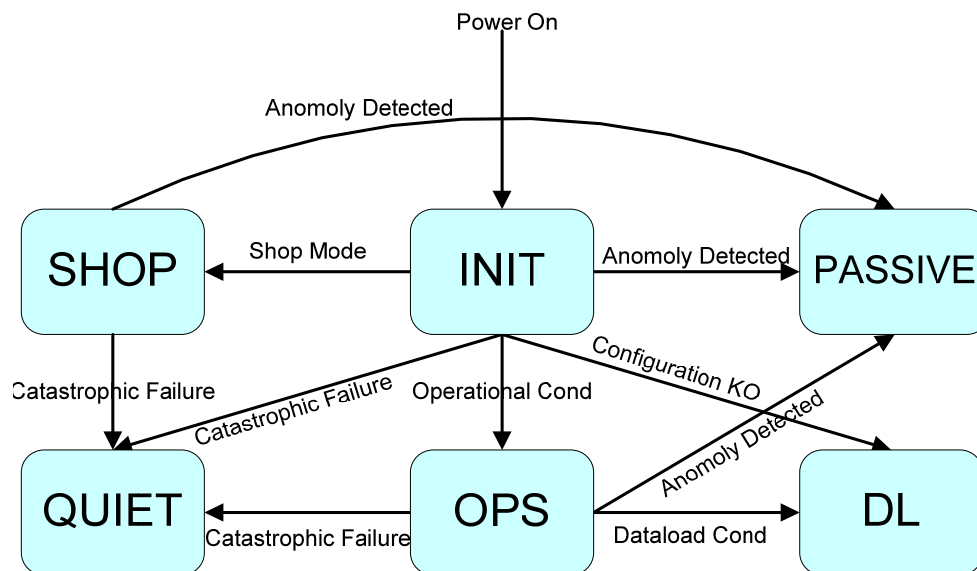


Figure 4-9 – Switch Operating Modes

The Switch enters the INIT mode after power rises or after a reset.

4.0 SWITCH SPECIFICATION

OPS is the operational mode of the Switch. In this mode, the Switch performs operational functions: frame and traffic filtering function, switching function, monitoring function, etc., relying upon the information contained in the loaded configuration tables: OPS_Configuration_File.

DL is the mode in which the Switch is only concentrated on its own dataloading (in particular the uploading operation).

An optional SHOP mode is defined and can be used for performing off aircraft debugging. Shop condition should be defined in an ARINC 700 specification.

Anomaly detected during the INIT mode will lead to the PASSIVE MODE activation. In PASSIVE MODE, the Switch only offers the capability to communicate with the Network Management Function), all others functions should be stopped.

QUIET mode is entered in the event of any sort of catastrophic failure that would cause the functional behavior of the switch to be questionable

The reset activation allows going back to INIT mode, from all the modes.

COMMENTARY

Reset capacity is optional and could be defined in an ARINC Standard or in additional system integrator specification.

4.8.2 INIT

INIT is the default mode in which the Switch enters after power rise or after a reset. In this mode, the Switch prepares and chooses its following mode depending of its context.

The Switch should systematically enter in INIT MODE after power rise following a power interruption longer than the Transparency Time or Power Hold Up time or after a manual reset activation.

COMMENTARY

System integrator or an ARINC Standard should define the value for Transparency Time/Power Hold Up time.

4.0 SWITCH SPECIFICATION

4.8.2.1 Initialization Sequence

In INIT MODE the Switch performs the initialization sequence described in Figure 4-10.

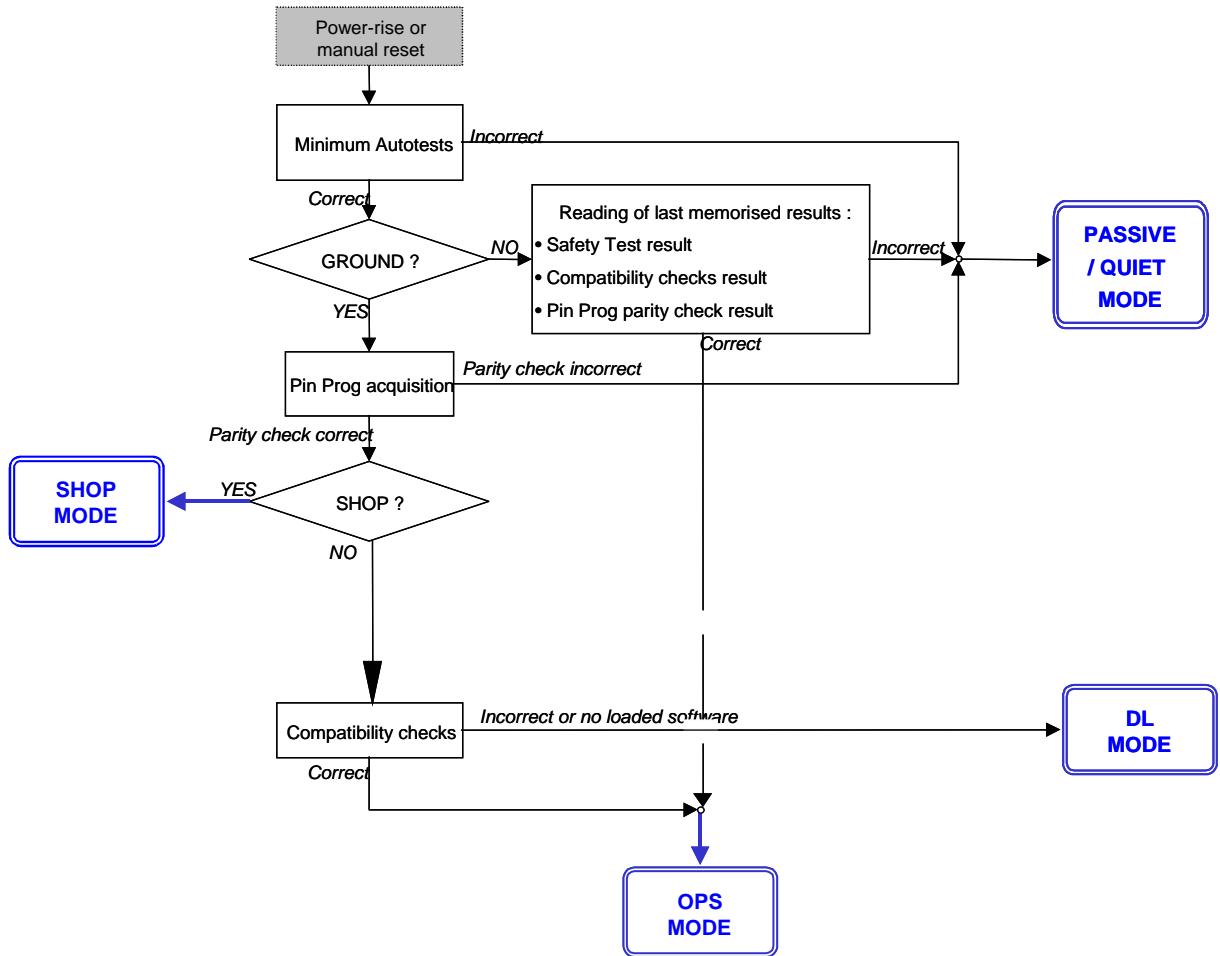


Figure 4-10 – Typical Switch Initialization Sequence

The Switches should be operational before their subscribers (connected LRUs), to allow subscribers to communicate via AFDX during their own initialization phase. Therefore, the initialization sequence should be performed in less than a specified initialization time, from power rise to full OPS MODE (all OPS functions are running).

COMMENTARY

System integrator or an ARINC Standard should define this initialization time both when Switch_Ground_Condition is active or not.

During INIT, the Fault/Healthy indicator should stay in the Fault state.

4.0 SWITCH SPECIFICATION**4.8.2.2 Ground_Condition**

System integrator or an ARINC Standard should define the ground condition value (number of signals, logical condition, confirmation...).

4.8.3 OPS: Operational Mode

After the initialization sequence, the Switch should enter in OPS (operational mode) if the result of compatibility checks is correct and if the shop condition is not active.

In OPS, the Switch should perform its normal functions (frames filtering and Traffic policing functions, Switching function, End System function) exclusively relying upon the data contained in the activated part of the OPS_Configuration_File.

In OPS mode, the Switch End System should be able to perform the following ARINC 615A dataloading operations:

- Information operation
- FIND request

In OPS mode, the Fault/Healthy indicator should be set in the Healthy state.

4.8.4 DL: Dataloading Mode

To allow the uploading of an empty Switch (it means with no loaded software, example: after the installation of an empty Switch in place of a failed one), from the INIT MODE, the Switch should enter in DL MODE only if the following conditions are both true:

- The Switch_Ground_Condition is active
- The results of compatibility checks are incorrect or there is no loaded software

From the OPS MODE, the Switch should enter the DL MODE only if the following conditions are all true:

- The Switch_Ground_Condition is active
- The Switch received an ARINC 615A uploading operation initialization request from the dataloader ([TH_Uploading_Initialization] message) addressed to the own end system of the Switch
- The header file has been received and accepted

Transition from OPS mode to DL mode should not disrupt the ARINC 615A session.

COMMENTARY

This transition is transparent regarding the ARINC 615A protocol.

In DL mode, the Switch should be able to completely perform the following ARINC 615A dataloading operations:

- Information operation
- Uploading operation
- FIND request

4.0 SWITCH SPECIFICATION

Uploading is a particular operation that must be preferably exclusive. During the uploading operation, the Switch is only dedicated to this operation.

In DL Mode, the Switch should perform its End System function exclusively relying upon the data contained in its resident configuration table: Default_Configuration_Table.

COMMENTARY

To allow the uploading of an empty Switch (it means with no loaded software, e.g., after the installation of an empty Switch in place of a failed one).

In DL mode, the Fault/Healthy indicator should be set in the Healthy state.

At the end of the DL mode, the Switch should return to INIT Mode. The end of the DL mode is determined internally by the switch. It corresponds to the end of the data loading function, as specified in ARINC 615A.

4.8.5 SHOP (Optional)

Conditions to enter in SHOP MODE are implementation dependent.

In this SHOP mode, some additional functionality of the Switch may be activated.

ARINC 615A operations (information, uploading, downloading, FIND) could be available in SHOP mode.

4.8.6 PASSIVE

In PASSIVE Mode, the functions of the Switch should be stopped and the switch should keep silent. Nevertheless, the capability to communicate with the Network Management Function should be maintained except if the passive state results from an erroneous parity of the pin programming discretes.

COMMENTARY

If the pin programming is incorrect, the switch has no knowledge of its own MAC address and is unable to exchange information with the Network Management Function.

The default VLs (called VL (0, position) and VL (1, position) in this document) are used to communicate with the Network Management Function.

COMMENTARY

This will be the same VL as the one used for Dataloading. Note that in order to use SNMP and ICMP in this mode, it is necessary that the same LRU within the avionics network sources the SNMP, ICMP and Data Loading requests.

4.0 SWITCH SPECIFICATION

From the INIT mode, the Switch should enter in PASSIVE Mode as soon as one of the following conditions is true:

- Pin programming parity check fails
- Safety Test (if any) result is incorrect

COMMENTARY

The Switch should not enter in OPS mode when one of the above conditions is true. Also, refer to Section 4.10.1.2, Parity Check and Processing, for pin programming parity check definition.

When entering in PASSIVE Mode, the Switch should set the Fault/Healthy indicator in the Fault state.

4.8.7 QUIET

In QUIET mode, the functions of the switch are stopped, the switch does not transmit frames or allow communication with the switch management function (SNMP). The switch software disables all ports when transitioning to QUIET mode.

When entering QUIET Mode, the switch sets the Fault/Healthy indicator to the Fault state.

QUIET mode is entered in the event of any sort of catastrophic failure that would cause the functional behavior of the switch to be questionable. Some examples of these types are failures are:

- Programming pin CRC failure
- Heart Beat monitor test failure
- Boot Code CRC failure
- Processor Instruction test failure
- Address/Data Bus monitor failure
- RAM memory monitoring failure
- Built-In-Self-Test (BIST) failure
- Hardware register test failure
- Default OPS software CRC check failure
- Default Configuration file CRC check failure

These types of failures all suggest some sort of serious hardware problem. These types of failures have the potential to cause erratic and unpredictable behavior on the part of the switch, the switch can't be trusted. The quiet mode ensures that the switch will not forward frames and will remain quiet (not try to generate a message).

4.0 SWITCH SPECIFICATION

4.9 Dataloading

4.9.1 General Dataloading Requirements

ARINC Report 615A and ARINC Report 665 should be used for the process and protocols to upload software and configuration tables into the switch.

4.9.2 Configuration Identification

4.9.2.1 Definition of Switch Configuration

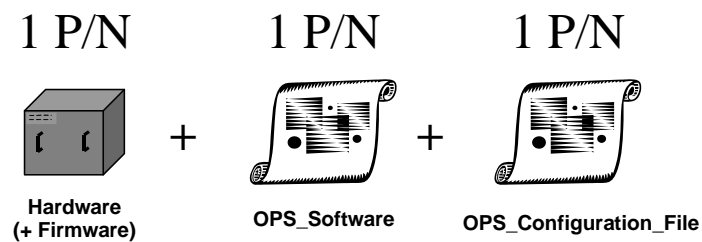


Figure 4-11 – Definition of Switch configuration

Each Switch is identified, at most, with the three following PNs, as shown in Figure 4-11.

- Its hardware PN that includes the resident software (Firmware). The resident software cannot be loaded on board. It is only changed in the repair shop, using specific procedures.
- Two field loadable software PNs:
 - OPS_Software: field loadable operational software
 - OPS_Configuration_File: field loadable configuration tables of the Switch

The Switch should not have more than two field loadable software (OPS_Configuration_File and OPS_Software). Those field loadable configuration file should be the same for all the Switches in the A/C.

COMMENTARY

OPS_Configuration_File and OPS_Software could be combined in the same Load.

4.9.2.2 Switch-On Configuration Identification

When the Switch is powered on, its configuration information should be accessible through the ARINC 615A “information operation.”

4.9.3 Dataloader IP Address

The Switch should learn the IP address of the data loader by reading the IP source address of the data loader.

4.0 SWITCH SPECIFICATION

4.10 Pin Programming

SWITCH hardware pin programming should be used to determine the position of the SWITCH in the network and then to associate default MAC and IP addresses of the End System of the SWITCH.

4.10.1 Pin Programming Processing

4.10.1.1 When Pins Programming Should be Read

Hardware program pins should be read during the initialization (INIT mode) of the SWITCH only if the Switch_Ground_Condition is true and before performing the Safety Test (if any).

When the Switch_Ground_Condition is false, hardware program pins should not be read. The Switch should use the last memorized values.

COMMENTARY

This ensures that a pin programming failure in flight has no operational consequences.

4.10.1.2 Parity Check and Processing

The 12 program pins (P1 to P12) should be checked by a parity bit.

In order to be used in case of power cut in flight condition, when program pins are read and parity check passed, program pins should be memorized in non-volatile memory of the SWITCH.

4.10.2 List of Pin Programming

4.10.2.1 Position Coding

The Switch should acquire its position in the aircraft with 12 pin programming, named P1 to P12. GROUND value of the Pin Programming should be coded as 1. OPEN value of the Pin Programming should be coded as 0.

4.11 Performance Characteristics

4.11.1 General Characteristics

The filtering, policing, and forwarding function of the switch should be able to process at least 4096 VLs.

COMMENTARY

This corresponds to 4096 different MAC destination addresses.

COMMENTARY

System integrator or an ARINC Standard should define the number of physical ports of the switch.

4.0 SWITCH SPECIFICATION**4.11.2 Physical Layer Characteristics**

As a network component, a switch has to conform to some basic physical characteristics of this network. Among these are physical link speed or mode of medium access. Subsequently applicable requirements are listed.

Each port speed should be configurable to operate to speeds of 10 Mbps or 100Mbps in full duplex, without auto negotiation. The speed of each port should be defined in configuration table.

The physical layer should comply with ARINC Specification 664, Part 2.

4.11.3 Processing Capabilities

Given a deterministic configuration (a deterministic configuration is defined by the absence of any output port saturation) the switch must be able to process (i.e. receive, filter, police, relay, transmit) all frames that are incoming at wire speed whatever their size.

COMMENTARY

This allows for maximum performance of the frame header processing and frame switching (for small frames) as well as sufficient memory (to buffer large frames).

The previous requirement implicitly imposes a fair processing of the input ports: no input port should have priority over another.

The MAC layer of each switch input port should be able to receive frames at wire speed.

COMMENTARY

Wire speed is defined as the maximum frame rate achieved when frames are received with a minimum Interframe Gap of 12 octets regardless of frame size.

This corresponds to a maximum frame rate of: 64 bytes (frame) + 12 bytes (IFG) + 7 bytes (Preamble) + 1 byte (SFD) = 84 bytes. When operating at 100 Mb/s, this corresponds to duration of 6.72 μ s per frame (about 148800 frames per second).

Each switch port should be able to filter frames of any valid size that are arriving at wire speed.

Each switch port should be able to police the VLs for frames of any valid size that are arriving at wire speed

Given a deterministic configuration, the switch should be able to relay all incoming frames that are not rejected by filtering and policing, at wire speed and whatever their size.

4.0 SWITCH SPECIFICATION

COMMENTARY

A deterministic configuration is defined by the absence of any output port saturation.

The MAC layer of each switch output port should be able to transmit frames at wire speed.

The technological latency of the switch should be less than 100 μ s.

COMMENTARY

Latency is defined as being the elapsed time between the reception of the last bit of the frame until the transmission of the last bit of the frame. Switch latency is composed of three parts: Technologic latency of switching function, the configuration latency due to switch loading, and the time required to transmit the frame on the medium.

Therefore, technological latency parameter is a contributor to the determinism of the network configuration.

Data contention at the output ports is resolved by buffering. Each output port of the switch should be able to buffer at least 512 frames (balanced between high and low priority).

COMMENTARY

This requirement implies particularly a minimum storage capacity for each output port. Other buffer constraints (input buffer sizes, etc.) may arise from the switch design.

5.0 SYSTEM ISSUES

5.1 Performances

Performance is defined as a percentage of the maximum that the ES is able to handle. The maximum throughput ("wire speed") corresponds to back-to-back frames.

The actual performance is measured by the time necessary to process all frames received during a 1 ms burst of back-to-back frames shown in Figure 5-1.

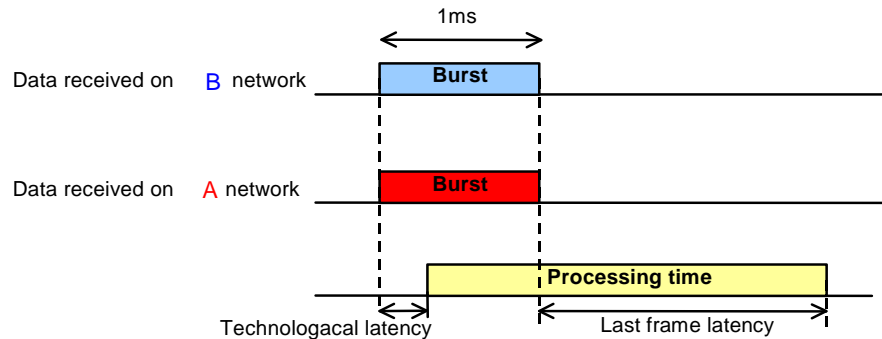


Figure 5-1 – 1 ms Burst of Back-To-Back Frames

The redundancy management algorithm is disabled. It is assumed that data of B and A networks belong to different VLs and thus are both to be provided to the relevant applications.

The performance (as a percentage of wire-speed capability) is given by the formula:

$$\text{Performance} = 100 \times \frac{1}{\text{lastframe latency} - \text{technological latency} + 1} = 100 \times \frac{1}{\text{processing_time}}$$

The "last frame latency" and the "technological latency" are expressed in ms.

The "last frame latency" corresponds to the latency of the last frame of the 1 ms bursts received, whether this frame corresponds to the B or the A network.

The points of measurement are the same as those defined for the latency in reception (see above).

This performance may depend on several parameters such as frame size, queuing or sampling data, end-system activity in transmission, etc.

The End System designer should provide information on the processing capabilities in transmission and reception of the end-system. As a guide, the following invariant parameters should be provided:

- Volume in transmission and reception given by:
 - Number of ports
 - Number of VLs
 - Number of SubVL
 - Frame size
 - Size of IP multicast group per VL

5.0 SYSTEM ISSUES

- Speed in transmission given by:
 - Latency
 - Frame rate
- Speed in reception given by:
 - Latency
 - Traffic profile

A way to measure the performance in reception is proposed at the beginning of this section.

Depending on the design of the end-system, the performance characteristics may vary with the actual configuration (e.g., number of VLS, use of fragmentation or not).

The End System designer should provide information regarding the data storage capacity in reception of the ES.

**ATTACHMENT 1
DATA FORMAT****1-1.0 Introduction**

This attachment provides requirements for formatting messages used on AFDX aircraft data networks in the Avionics domain. These messages are created and received at the application layer (OSI model) and do not include message formats for application layer protocols such as TFTP or ARINC 661. Formats presented here are restricted to non-protocol based data.

This specification is intended to define the message format as it appears on the network medium. Some implementations may use intermediate layers of software to perform format parsing or translation. This could result in applications working with different formats than those specified herein, for both messages and data primitives. The actual formats, with which applications work, is not within the scope of this specification. This specification only deals with data representation from that presented by a data producer as the transport layer payload, across the network medium, and up to the data consumer transport layer payload.

This data formatting approach consists of the following sections:

1. Introduction
2. Formats for primitive data elements (Booleans, integers, floats, strings, etc.)
3. A description of the AFDX message structure.
4. Example formats for grouping aircraft parameters into messages on the network

1-1.1 Bit/Byte Order of Ethernet

An aircraft data network for byte order, like the internet standard, specifies that integers are sent with the most significant byte first (i.e., Big Endian style). If one considers the successive bytes in a packet as it travels from one machine to another, a binary integer in that packet has its most significant bytes nearest the beginning of the packet and its least significant byte nearest the end of the packet.

Ethernet is defined as being Big Endian. This means data is represented in memory with higher bit numbers and bytes first (lower address values) followed by the bytes containing lower bit numbers. Figure 1-1.0 illustrates how primitive data structures are represented in the Ethernet interface buffer memory. It then shows how the data appears on the medium.

1-1.2 Abstract and Transfer Syntaxes

Starting at the lowest level in the data hierarchy are data elements. Data elements, or data primitives, are the smallest pieces of data identified within an aircraft data network. A data element is a specific value, defined by range and resolution that represents a discrete piece of information. Before we discuss the formats of these data elements the concept of abstract and transfer representations should be introduced.

ATTACHMENT 1 DATA FORMAT

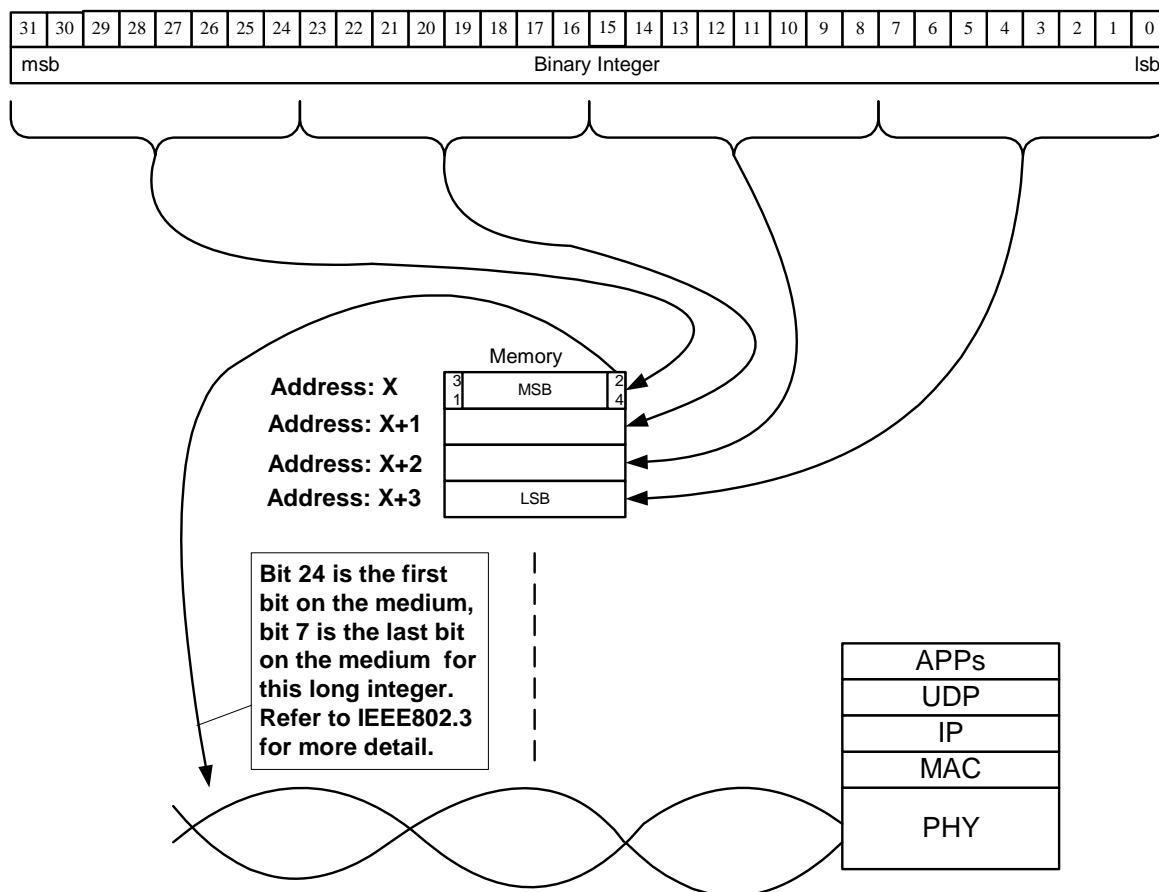


Figure 1-1.0 – Ethernet is Big Endian – Bit and Byte Order on the Medium

1-1.2.1 Abstract Syntax

The abstract syntax is just that, an abstraction away from the underlying computing platform and network. The abstract representation is the format the applications use. An abstract syntax will allow implementers to describe messages in some sort of high level definition language that is a human readable form. This definition will show how each message is made up of a set of data primitives, each data primitive will have an acceptable range of values. It would be very impractical for a designer to attempt to explicitly list all possible messages because of the nearly infinite number of possible combinations of values for each data element in a realistic message. Similarly, the large message size of Ethernet makes a virtually infinite set of different messages possible. Fortunately, tools are available to help with this problem. There are high level definition languages available to help define message formats (define which data primitives are in the message, in what order they appear in the message, and possibly what range of values are acceptable). Some of these languages have translators (compilers) available that can generate programming language data structures to represent the message formats automatically.

**ATTACHMENT 1
DATA FORMAT**

1-1.2.2 Transfer Syntax

The transfer syntax, or physical syntax, is the format of messages as they proceed through the network. Complete messages, made up of combinations of data primitives as defined by the abstract syntax are transmitted by applications. The underlying platform (computer and network interface) may perform a translation of the message to the transfer syntax. The message is forwarded through the network to one or more destination computers where another translation is performed from the transfer syntax to the abstract syntax before the message is passed to the applications.

COMMENTARY

The presentation layer of the OSI model (layer 6) is intended to perform services of this type. This is one possible implementation, to include a thin layer of software at this layer that performs encoding and decoding of messages. This can serve as the translation between the abstract and transfer syntaxes. An alternative to this could be to link in encoder/decoder code with applications. The creation of this code could be built using tools that would make it fairly automated.

A system designers choice of transfer syntax will depend on the requirements of the system. Transfer efficiency might be a high priority for one system, ease of encoding might be a high priority for another, possibly reliability is a priority for yet another. For a system where ease of encoding/decoding is of critical importance, the transfer syntax could be chosen to be identical to the abstract syntax.

Consider an enumerated type parameter, for example. Suppose the enumeration is a mode indicator for a piece of equipment with a definition as shown in Figure 1-1.1. This enumerated type parameter could be an abstract syntax definition. The encoding rules might encode this as a 32-bit signed integer, which is one of the allowed transfer syntax data primitives.

Processor Mode Enum Element Specification		
Labels:	Value	Label
	0	Off
	1	Halt
	2	Invalid Strapping
	3	Dataload
	6	Invalid Platform Configuration
	7	Software Validation
	13	IBIT
	18	Normal

Figure 1-1.1 – Enumerated Type Definition Example

**ATTACHMENT 1
DATA FORMAT**

COMMENTARY

Suppose, for example, that an aircraft was equipped with a network and system of computers, and all LRU/Ms were equipped with Ethernet interfaces except one, the Air Data sensor (ADS). Suppose that the ADS was A429 and provided the system with one label, airspeed. The transfer syntax for this system might be identical to the abstract syntax with the exception of the airspeed message. A transfer syntax could be developed for that 32-bit label, and parser code written to translate it into abstract syntax at each recipient. The applications that use the airspeed parameter use the abstract syntax, consequently when the A429 based ADS is upgraded to an Ethernet based ADS at some point in the future, all that must be done is to run some tools (parser code gets removed), and certain units dataloaded. The applications need not change. This approach helps make them to be more mature (less changes) and portable.

1-2.0 Primitive Data Elements

Each data element is a predefined network-independent data type that is based on a language-neutral Interface Definition Language (IDL). These data types have a well-defined bit and byte order that enables applications to send and receive a data element via the network regardless of the native bit or byte order of the host processors used in the system. The data element types currently defined for use on aircraft data networks are listed in Table 1-1.0.

COMMENTARY

Every effort should be made to use only the data primitives listed in Table 1-1.0 for AFDX networks. When circumstances require that some data structure other than those listed must be used, it should be placed in the opaque data type. Use of the opaque data type, however, should be avoided as much as possible. AFDX messages should be made up of the defined data primitives when ever possible.

Table 1-1.0 – Primitive Data Types

Data Element Type	Size in Bits	Range
Signed_32 Integer	32	$-2^{31} \dots 2^{31} - 1$
Signed_64 Integer	64	$-2^{63} \dots 2^{63} - 1$
Float_32	32	$\pm(2^{-1} / 2^{22}) 2^{-126} \dots 127$
Float_64	64	$\pm(2^{-1} / 2^{51}) 2^{-1022} \dots 1023$
Boolean	32	Not Applicable
String	Varies with definition of data element (see paragraph 2.5)	Not Applicable
Opaque Data	Varies with definition of data element (see paragraph 2.6)	Not Applicable

ATTACHMENT 1 DATA FORMAT

This small set of data primitives serves to keep data formatting simple. Keeping the formatting simple will serve the industry by reducing the complexity of messages and the amount of software required to encode and decode many different data types. Some efficiency is lost to achieve this simplicity. It is not the most efficient approach to store an integer in 32-bits that could be represented with an 8-bit integer. But there is computational benefit in having all integers on 32-bit boundaries. Ethernet has a very large bandwidth compared to past technologies, and message sizes can be significantly larger.

The following paragraphs will present the formats for each primitive data element that will appear on the network.

COMMENTARY

The integer and floating point data types are never scaled in AFDX messages. The scale factor is always one (1), in contrast to ARINC 429 where parameters quite often have scale factors associated with them other than 1.

1-2.1 Signed Long Integer – Signed_32

The 32-bit signed integer uses the two's complement data representation format.

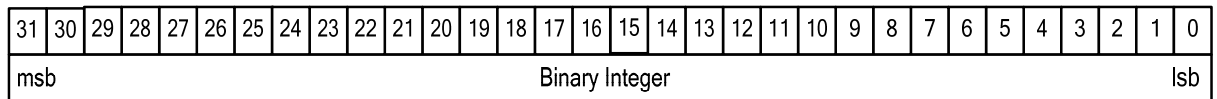


Figure 1-2.1 – Signed 32-Bit Integer Representation

COMMENTARY

When converting 8 or 16-bit integers into this standard 32-bit two's complement signed integer, the shorter integer should be sign-extended if it is a signed value, or it should be zero-extended if it is an unsigned value.

1-2.2 Signed Double Length Integer – Signed 64

The 64-bit signed integer uses the two's complement data representation as shown in Figure 1-2.2.

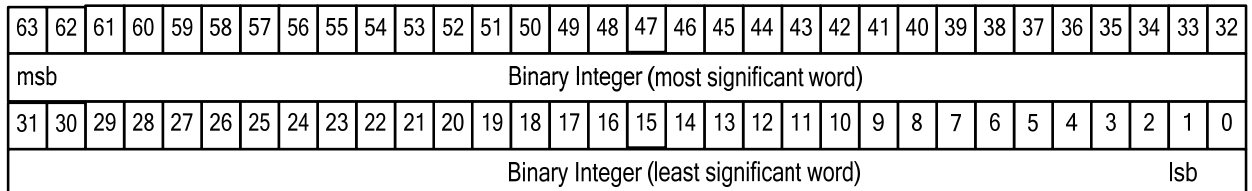


Figure 1-2.2 – 64-Bit Signed Integer Representation

1-2.3 Floating Point

Floating point numbers are represented using the IEEE 754 formats. Details can be found in that standard, some information is provided here for convenience.

ATTACHMENT 1 DATA FORMAT

1-2.3.1 Standard Precision Floating Point – Float_32 - IEEE 754

The single precision floating point format is 32 bits long with 3 fields. These fields are:

1. S: sign bit – 0 indicates a positive number, 1 indicates a negative number.
2. E: exponent – base 2 number, 2 is raised to this power, 8 bits.
3. F: mantissa – the fractional part, base 2, 23 bits in representation.

The real number value for normalized numbers is given by the formula:

$(-1)^S \times (1.F) \times 2^E$. The real number value for de-normalized numbers is given by the formula:

$$(-1)^S \times (0.F) \times 2^E$$

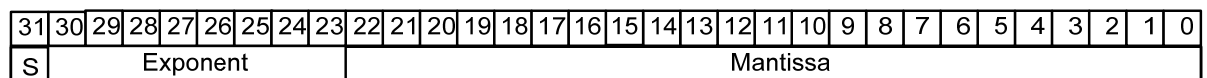


Figure 1-2.3.1 – 32-Bit Floating Point Number Representation

1-2.3.2 Double Precision Floating Point – Float_64 - IEEE 754

The double precision floating point format is 64 bits long with 3 fields. These fields are:

1. S: sign bit – 0 indicates a positive number, 1 indicates a negative number.
2. E: exponent – base 2 number, 2 is raised to this power, 11 bits.
3. F: mantissa – the fractional part, base 2, 52 bits in representation.

The real number value for normalized numbers is given by the formula:

$(-1)^S \times (1.F) \times 2^E$. The real number value for de-normalized numbers is given by the formula:

$$(-1)^S \times (0.F) \times 2^E$$

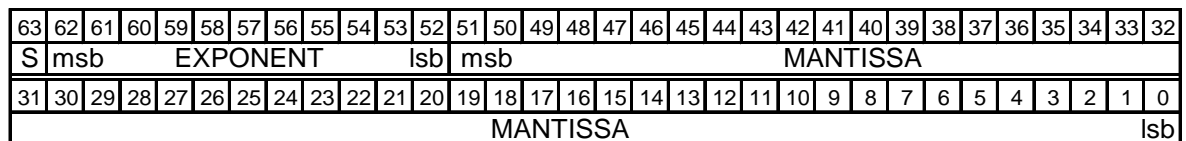


Figure 1-2.3.2 – 64-Bit Floating Point Number Representation

1-2.4 Booleans

1-2.4.1 Standard Boolean

Booleans are represented by a single bit in a 32-bit field. If just one Boolean is represented by a 32-bit field, bit 0 (least significant bit) is used. If bit 0 is a logical “1” the Boolean is said to be true (or yes, or active, or on, etc). If bit 0 is a logical “0” the value of the Boolean is said to be false (or no, or inactive, or off, etc.).

ATTACHMENT 1
DATA FORMAT

1-2.4.2 Bit-Wise Packed Boolean

Booleans may be packed into 32-bit structures (see Figure 1-2.4.2). Each bit will then represent their respective Boolean entities. A single 32-bit structure can represent up to 32 Boolean entities. If 33 Booleans are required, a new 32-bit structure is added to hold the 33rd bit. Booleans are assigned to a structure starting from the least significant bit and filling in toward the most significant bit. Unused bits at the most significant end of the 32-bit structure must be all binary zeros.

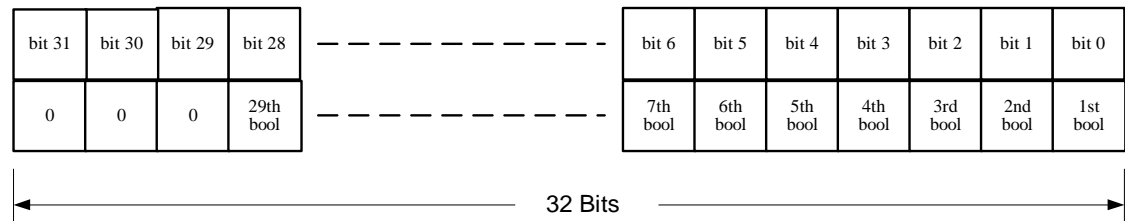


Figure 1-2.4.2 – Bit-Wise Packed Booleans (29 Bools)

1-2.5 Strings

Strings are defined as a sequence of n ASCII characters (see Figure 1-2.5.1). There is one ASCII character per byte, they are numbered 1 through n. The first 2 bytes of the string structure is a 16-bit unsigned integer, this value is the length of the string (number of characters in the string).

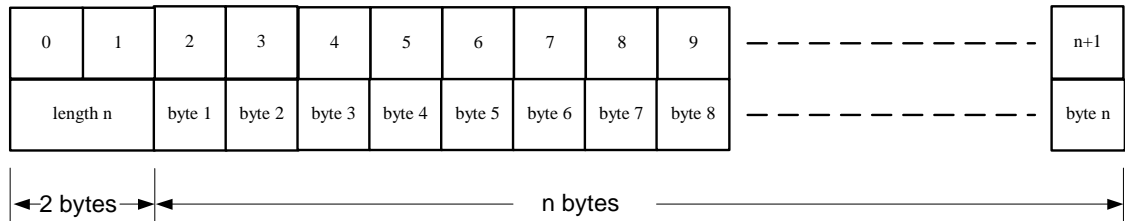


Figure 1-2.5.1 – String Data Structure Format

Strings are fixed-size, data structures that consists of three components; the length field, the data field, and the pad field. The value in the length field defines how many bytes the data field occupies, the pad field places binary zeros in any remaining bytes to fill out the fixed size of the data structure. See Figure 1-2.5.2.

ATTACHMENT 1 DATA FORMAT

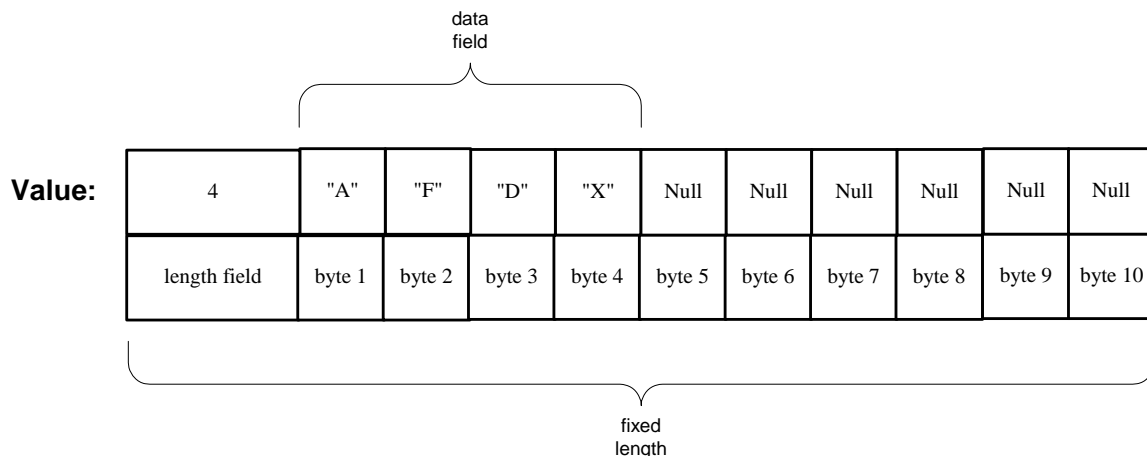


Figure 1-2.5.2 – String Data Structure

1-2.6 Opaque Data

Opaque data is a data primitive used for storing data when the type or structure of the data is not one of those listed in Table 1-1.0 or is not given. The data format might not be relevant to avionics, the fields might be optional, or variant. Maybe the data is not describable. This type of data we call “opaque”. All unused bytes of an opaque data structure should be filled with binary zeros.

1-2.6.1 Fixed Length Opaque Data

The fixed length opaque data structure is defined as a sequence of n bytes, numbered 1 through n . Refer to Figure 1-2.6.1.1.

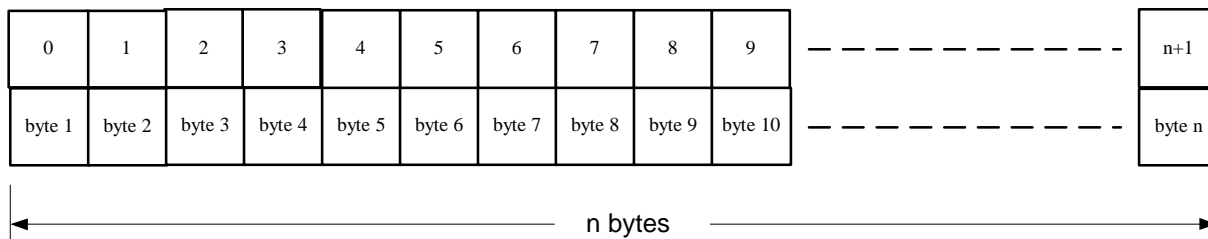


Figure 1-2.6.1.1 – Fixed Length Opaque Data Structure

1-2.6.2 Variable Length Opaque Data

The variable length opaque data structure is defined as a half word (16 bits) length field followed by a sequence of n bytes numbered 1 through n . See Figure 1-2.6.2.1. The length field is encoded as a 16-bit unsigned integer. Byte m in the sequence (any arbitrary byte of the sequence) always follows byte $m - 1$ and the first byte of the sequence always follows the length field.

This structure allows data to be stored without knowing the details of its format. The length field allows knowing the length of a data element that is stored in the structure even though the data element might not use all of the defined storage area. The length n represents the true length of the data element, not necessarily the maximum size of the field.

ATTACHMENT 1 DATA FORMAT

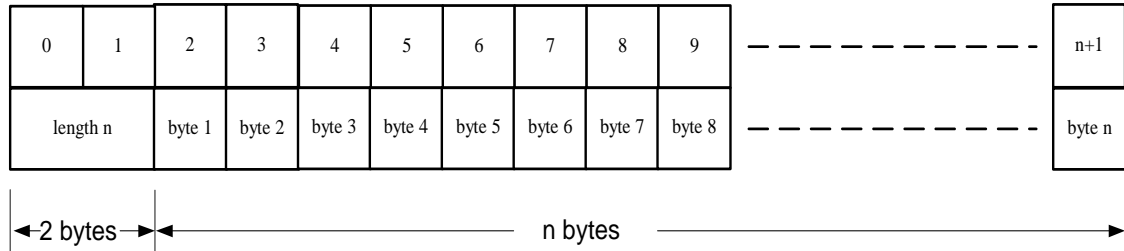


Figure 1-2.6.2.1 – Variable Length Opaque Data Structure

Variable length opaque data structures are fixed-size, data structures that consists of three components; the length field, the data field, and the pad field. The value in the length field defines how many bytes the data field occupies, the pad field places binary zeros in any remaining bytes to fill out the fixed size of the data structure. Refer to Figure 1-2.6.2.2.

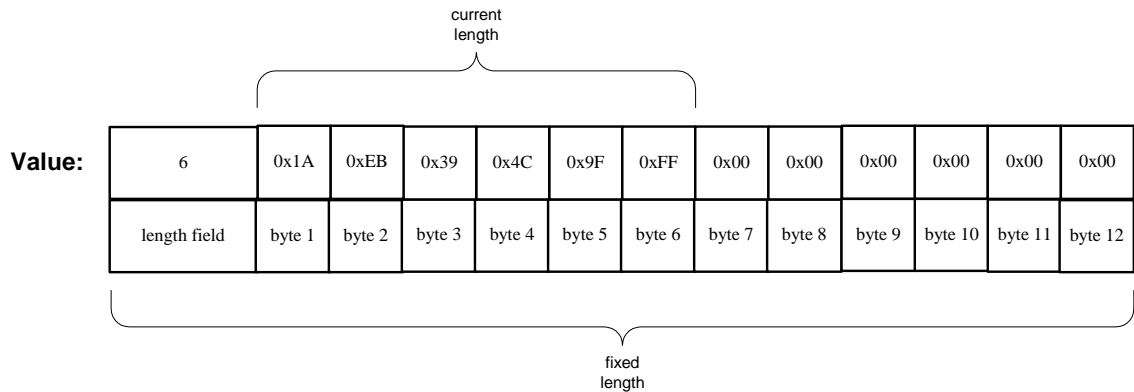


Figure 1-2.6.2.2 – Variable Length Opaque Data

1-3.0 Message Structure

This section deals with the format of messages as viewed from the application layer of the OSI model. These messages are the payload that the network delivers.

1-3.1 Implicit versus Explicit Messages/Port Numbers

Message structures can be defined as Implicit or Explicit. Explicit message structures include format information that allows a recipient of the message to decode the message. Format information is overhead, it might include identifiers to indicate what message structure is encoded in the message and/or length parameters that indicate how long the structure is.

Implicit messages have no format overhead associated with them. The message contains only the data parameters with no means of identifying how the message content is to be interpreted. Implicit messages make for more efficient use of network bandwidth.

The message structure defined in this section is exclusively implicit. An application will know the message format by means of the port number it is received on. This practice is in alignment with the widely accepted concept in the Internet world of a “well known service” (WKS). For

**ATTACHMENT 1
DATA FORMAT**

example, the well known service TFTP (Trivial File Transfer Protocol) is available on port 69. Port numbers and WKS names are assigned for each message format on the aircraft network during integration, the message is referenced then by its well know service name.

There are two types of port numbers used in typical AFDX implementations:

1. The transport layer (UDP or TCP) port numbers
2. The ARINC 653 communications ports (comport)

The A653 comports are not within the scope of this specification, therefore they will not be discussed except to say that their values are typically chosen arbitrarily and can be mapped one for one to the transport layer port number for simplicity.

Transport layer UDP and TCP port numbers are managed by the Internet Assigned Numbers Authority (IANA). They fall into three groups as follows:

1. Assigned port numbers: 0 – 1023
2. Registered port numbers: 1024 – 49151
3. Dynamic/Private port numbers: 49152 – 65535

An AFDX network is a statically configured closed network. Transport layer port assignments are therefore not critical and can be assigned by a system integrator using any values in the range 1 to 65535. There could be some value in using the well-known-service port numbers for those services that exist on the network (such as port 69 for TFTP and port 80 for HTTP), this will help reduce confusion and leverage the experience of the team members familiar with commercial networking.

Because the AFDX network is closed, communication with other networks on the aircraft is performed through an upper layer interconnection device. This device could be a layer 3 router or a gateway. If a gateway is used, it will provide translation of addressing, up to and including transport layer addressing. This means the gateway will communicate to/from the AFDX network using pre-assigned port numbers. These pre-assigned numbers can be carefully chosen to ensure there are not transport layer addressing conflicts in adjacent networks.

In cases where network-to-network communication might be performed with a layer 3 router, and transport layer translation will not exist in this case, system integrators should consider avoiding the use of transport layer port numbers in the range of 0 – 1023 at a minimum. For full assurance of avoiding addressing issues with adjacent networks, the second group of port numbers from 1024 to 49151 should be avoided as well.

1-3.2 Data Alignment

Computers are more efficient at storing and retrieving data if the data is in proper alignment. Data alignment refers to how the data is stored in physical memory. If data is in alignment considerable processing time can be saved. If data elements are aligned in messages, it is easier for

ATTACHMENT 1

DATA FORMAT

applications to ensure their alignment when the message is stored in the application memory space.

A data element is said to be aligned if the address of the element in memory is evenly divisible by the data element length (in bytes). For example, a 4-byte (32-bit) scalar value is aligned if its address is a multiple of 4; an 8-byte scalar value is aligned if its address is a multiple of 8, and so on. For data primitives whose length can vary, such as a string, it would not be acceptable to expect its address to be divisible by its length as its length could get very large. In cases such as this, padding should be used to put the data on a 2-byte boundary (address evenly divisible by 2).

The Reserved word at the beginning of the message should be aligned on a 32-bit boundary. Functional Status Sets should also be aligned on 32-bit boundaries throughout the message. When the message is transferred to the application on receive the message should be placed in a buffer in memory that is positioned such that the Reserved word is aligned. This ensures that the entire message is then aligned.

A consequence of using data alignment is an occasional need for padding. Consider the case where a 4-byte scalar is stored at address 0x0008 – 0x000B (see Figure 1-3.2.1). If an 8-byte scalar is stored after it is in memory, it will not be stored beginning at address 0x000C as it would not be in proper alignment. It must be stored at address 0x0010 leaving addresses 0x000C thru 0x000F unused, or padding. Address 0x0010 is evenly divisible by 8. This discussion assumes that the first byte in the transport layer payload (message) is byte 0.

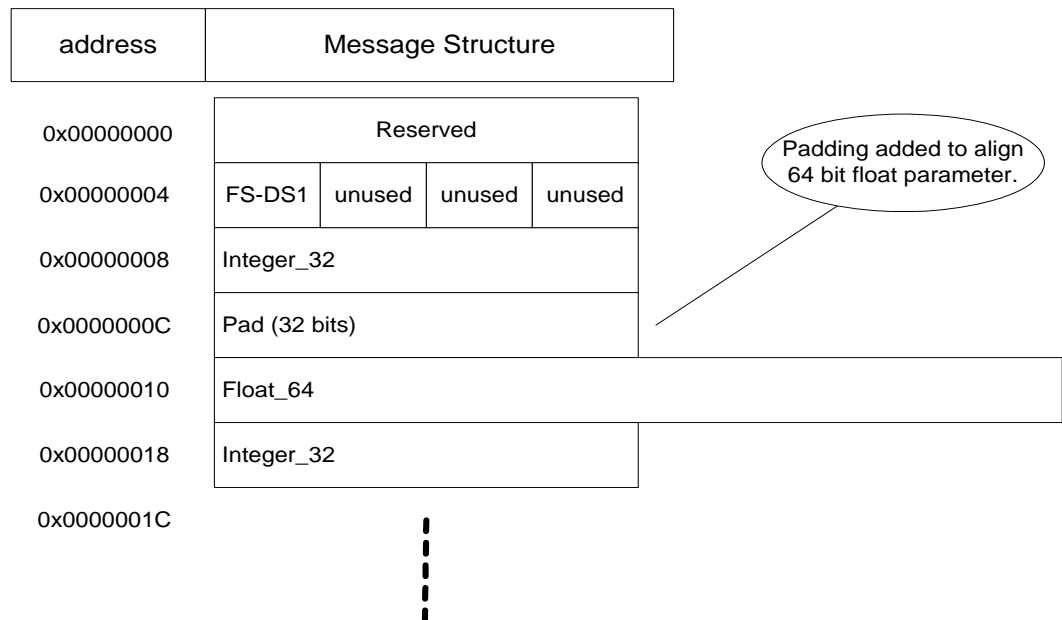


Figure 1-3.2.1 – Example of Data Alignment

**ATTACHMENT 1
DATA FORMAT**

In Figure 1-3.4.2.1 there is a definition of an example message. The fourth field in the message structure is a 32-bit pad field. This pad was placed in the structure for data alignment purposes, to ensure that the next field (Double precision float – 64 bits) is at an address that is divisible by its length (8 bytes).

1-3.3 Spare and Padding

A spare field in a message is an area that is not currently used, but might be reserved for future use. System integrators may choose to include spare fields in message formats as a means of controlling the cost of change in the future. Spare fields could be used to send data parameters in the future without any impact to applications that do not use them.

Padding is an unused area in the message that is the result of specific computer or communications requirements. The most common use of padding is for data alignment.

From a system point of view, both spare fields and padding are unused areas in messages. Both can be used for a new parameter that must be added to a message.

1-3.4 Functional Data Sets

Functional Data sets (FDS) are a means of grouping data primitives together in a message (data primitives as defined in section 2 of this document). Data primitives represent

aircraft parameters or other data. All non-protocol based data is formatted into Functional Data Sets for transmission on an ADN. Functional Data Sets are made up of two types of fields:

1. Functional Status Sets (FS)
2. Data Sets (DS)

The structure of the FDS is as shown in Figure 1-3.4.1. Functional status sets and data sets combine to form Functional Data Sets.

ATTACHMENT 1 DATA FORMAT

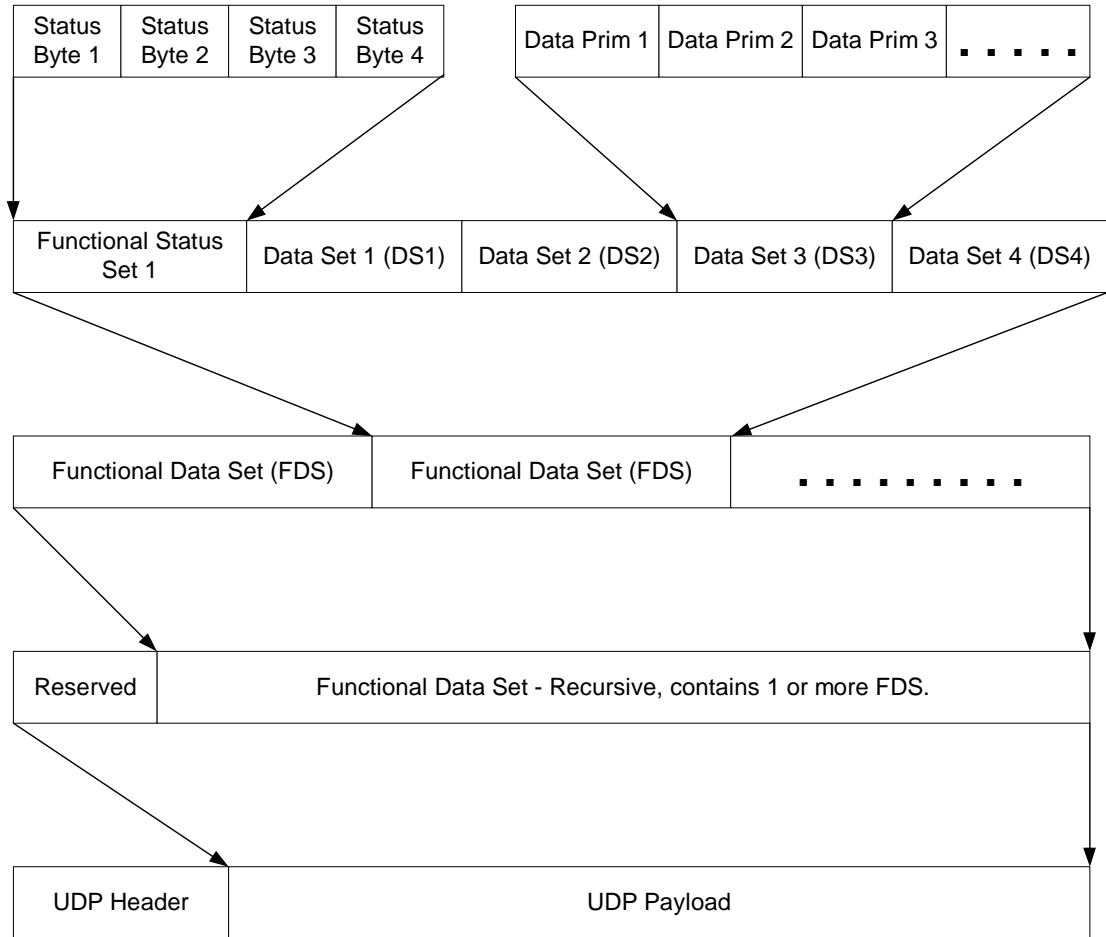


Figure 1-3.4.1 – Structure of a Functional Data Set (FDS)

1-3.4.1 Functional Status Set

The Functional Status Set (FSS) is a 32-bit field made up of four 8 bit status fields (see Figure 3.4.2). The first status field is used to represent the health and status of the first DS. If no other DS are used in a FDS, the remaining 24 bits of the FSS must be zeros. If other DS are used, each of the remaining status fields are used to represent health and status for them, up to 4 DS. If more than four DS are desired, a new FSS is added to the FDS followed by one to four DS, and so on. There is no limit to the number of FDSs (and therefore the number of DSs) that can be placed in a single message except the limitation of the underlying transport mechanisms (payload size).

A DS can have one or more data primitives in it. There is no limit to how many data primitives may be used in a DS other than the limitations of the underlying transport mechanisms. It is important to note that all the data primitives in any given data set are represented by one status field. This means that if one data primitive is invalid, the entire Data Set has to be marked as invalid. For this reason, it is advantageous to group data

**ATTACHMENT 1
DATA FORMAT**

primitives together that originate from the same equipment (such as a sensor). If one piece of this data set is invalid, they possibly are all invalid due to an equipment fault.

Consider a Multi-Mode Radio (MMR) for example. The MMR might have one Ethernet interface to allow all radios to communicate on the network (ILS, DME, etc.). Some data from each radio might be included in one Ethernet message but should be grouped in different DS so they are represented by different FS. If the ILS faults the ILS DS can be marked ND while the DME is still functioning correctly, the DME DS is marked NO (normal operation).

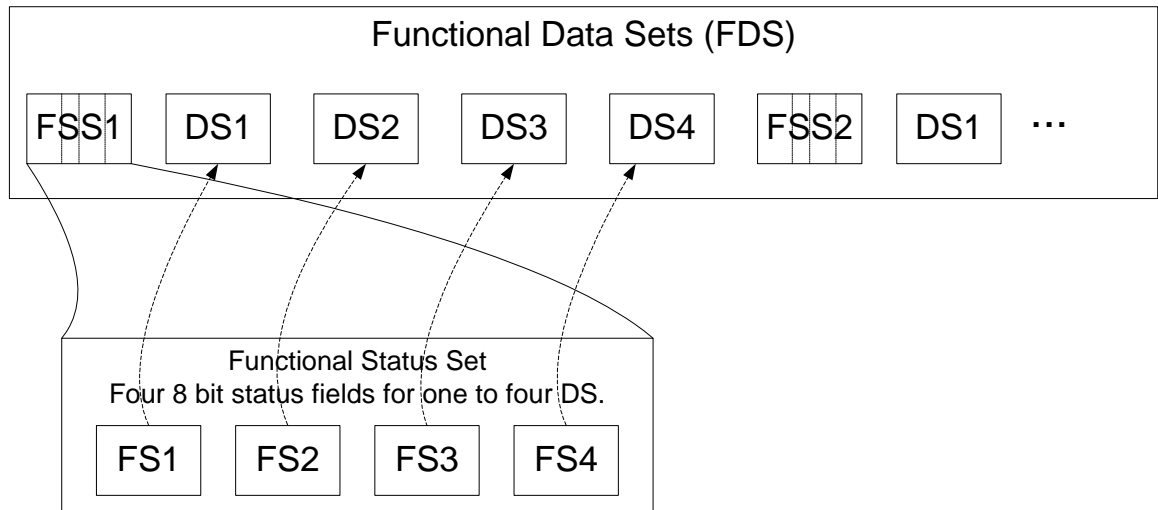


Figure 1-3.4.2 – Format of Functional Status Set

Avionics systems have to know the status of received data. The FS byte provides this indication for each data set in the FDS. The FS is a byte of enumerated type with enumerations as defined in Table 1-2. Refer to Table 1-3 for the enumeration binary and decimal representations.

Table 1-2 – Functional Status Enumerations

Condition	Definition
ND (no data)	No valid data in data set, this would include Fail Warn and other conditions where the contents are meaningless.
NO (normal operation)	Valid data, Normal operating conditions
FT (Functional test)	Equipment test conditions
NCD (no computed data)	Invalid data, equipment is in normal operating conditions but unable to compute reliable data.

This status information is provided by the data producers on the network. A ND (no data) status indication should be used if even one data primitive in the DS is not available. When computing the functional status for a DS it is possible for more than one condition to exist. In that case the condition with the highest priority should be encoded into the status field. Priority 1 is the highest priority, priority 4 is the lowest. For example, a

ATTACHMENT 1 DATA FORMAT

data set might be made up of 10 aircraft data parameters. Suppose 9 of those parameters are NO (normal operation) but the tenth parameter is NCD (no computed data). In this case the FS for this DS will be set to NCD because NCD has the higher priority.

The functional status definitions map reasonably well to ARINC 429 SSM status definitions. When mapping an ARINC 429 SSM “Failure Warning” condition to A664 use the ND (no data) condition. If a receiver reads a functional status value other than 0,3,12, or 48, it will discard the data in the associated DS.

Table 1-3 – Functional Status Enumeration Values

Condition	Binary Value	Decimal Value	Priority
ND (no data)	00000000b	0	1
NO (normal operation)	00000011b	3	4
FT (functional test)	00001100b	12	3
NCD (no computed data)	00110000b	48	2

1-3.4.2 Data Sets

The data set field is a group of one or more data primitives that comprise a data set (DS).

The DS can also include a data spare field at the end of the set if growth is required, or there is risk that parameters will be added later. Using a spare field can help reduce the cost of change by minimizing impact to application developers when data must be added to a set. Assigning data spare is a system designer responsibility. It should be used with care to preserve network bandwidth. Spare data fields can only be placed at the end of the DS. Padding fields can be used for future data as long as the size of the padding field is not changed. Refer to Figure 1-3.4.2.1 for an example of DS.

Data encoding in the data set field is as defined in Section 1-2.0 of Attachment 1. A data set is a group of data primitive structures. Data primitives are placed in the data set one after another with no unused bytes between them except for padding for alignment purposes. Spare data fields should be filled in with binary zeros if unused.

**ATTACHMENT 1
DATA FORMAT**

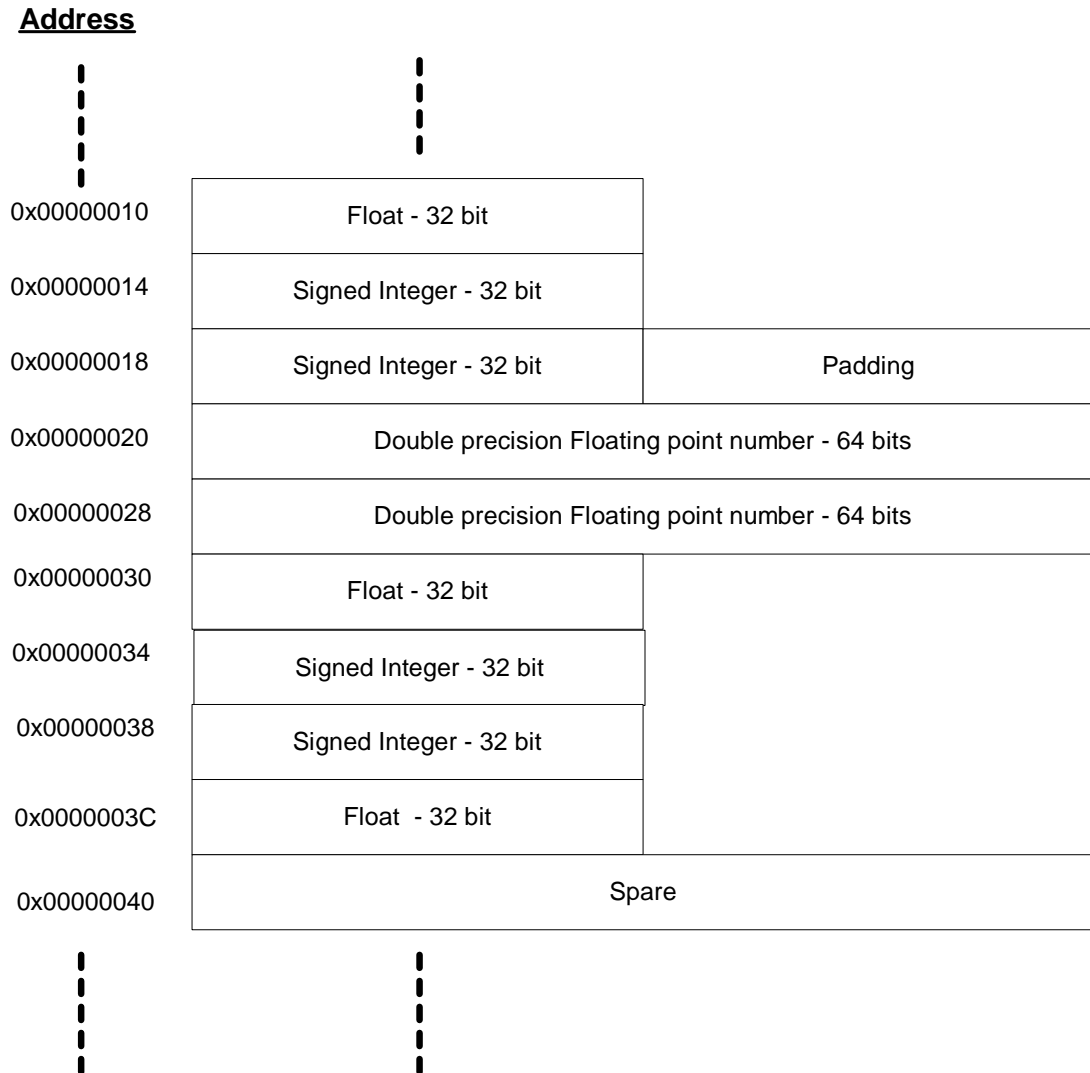


Figure 1-3.4.2.1 – Data Set – Example

COMMENTARY

The padding in the example message of Figure 1-3.4.2.1 is a consequence of placing large data primitives in the middle of messages. The need for padding can be reduced by placing the largest data primitives at the beginning of messages. However, this padding field can be used as spare, so there is some potential value in having padding.

1-3.5 Global Message Structure

Now we need to pull all these data structures together into a complete message. The format for a complete message is presented in Figure 1-3.5.1.

The first field of the global message structure is a 32-bit data field that is “reserved for future use”. This field is followed by one or more FDS where an FDS is made up of a FS and DS as outlined in previous sections. The

ATTACHMENT 1 DATA FORMAT

32-bit reserved field should be filled with all binary zeros. This field is not used, it is included in anticipation of future evolution.

The final field in the message is an optional Global Spare field, used at the discretion of the system designer. This field might have sufficient space in it for a new FDS should it be considered likely that it will become necessary in the future. Making space for growth early on has the potential to minimize the impact of changes in later years. If this field is included in a message, it should be filled with all binary zeros.

Global Aircraft Data Network Message - Non protocol base data

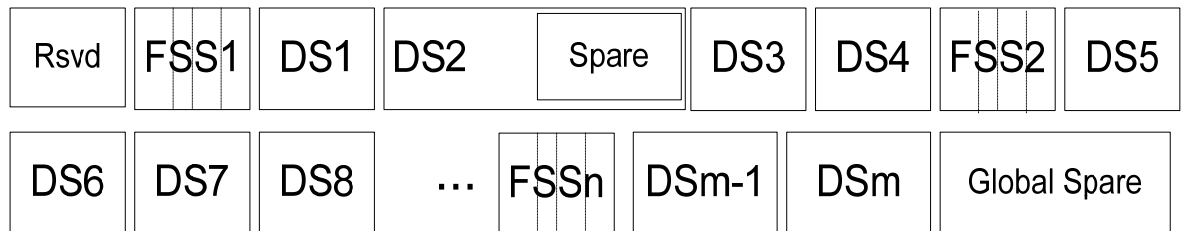


Figure 1-3.5.1 – Global ADN Network Message Structure

Figure 1-3.5.1 shows a complete ADN message that consists of the 32-bit reserved field followed by “n” functional data sets. The first functional data set has the functional status set followed by 4 data sets numbered DS1 – DS4. Data set number 2 has some spare storage at the end of it. The second functional data set starts with the functional status set followed by 4 data sets, numbered DS 5 – DS8. This pattern continues until FDSn. This functional data set starts with the functional status set followed by 2 data sets numbered DSm-1 and DSm. The Global spare field follows the final FDS, it might have space for a future DS that is part of FDSn or it could contain space for 2 additional DS for FDSn plus another complete FDS.

1-3.6 Guidelines for Designing Messages

Care must be taken when the need arises to modify a message structure that has already been defined and is in use. In order to keep the cost of change as low as possible, messages should be modified in a manner that does not shift the position of the data sets in the message. If the data sets are shifted within the message structure by a change, the software that produces and consumes the message will be unnecessarily impacted. The following practices can help avoid these costly changes of message structure:

1. Always define spare fields at the end of every data set.
2. Always define a global spare field at the end of the message.
3. Keep in mind that pad fields that have been added for alignment purposes can be used as spare. Use them to add a parameter when the need arises.

**ATTACHMENT 1
DATA FORMAT**

4. When removing a parameter from a data set, replace it with a pad. Do not shift the remainder of the message up and remove the place the parameter was filling.
5. Minimize the impact to the message by avoiding a shift of position of the existing message parameters, insert new parameters in spare or pad fields.

Global spare can be used to add a complete new data set to the end of a message, or be used to add parameters to the last data set defined in the message. The choice of which one to do is driven by the functional status. If the new parameter(s) is covered by the existing FS, put them in the last data set. If the new parameter(s) are represented by a different FS, create a new data set.

1-4.0 FDS Example Definitions

This section presents examples of how messages can be constructed using AFDX data primitives in the Functional Data Sets. For examples of how A429 to AFDX formatting might be done, refer to Appendix A.

1-4.1 AFDX Message Structure Definitions

Ethernet offers a considerably larger message length than that of A429. This allows for representing data in different formats than that of A429, the data elements can have more precision. The minimum size packet in Ethernet has a payload of 18 bytes. If a message is sent on an AFDX network that is less than 18 bytes long the message will be padded out to 18 bytes before it appears on the medium. Sending 18-byte payloads is not an efficient use of the network since this payload is small compared to the overhead of the packets. The network starts to become more efficient when packet sizes get up into the hundreds of bytes long.

Data primitives need to be grouped according to where they are produced. If a sensor, for example, sends several different ARINC 429 labels at the same rate, these data elements might be candidates for grouping together into a single message when that sensor is equipped with an AFDX network interface.

The rule of thumb used in the following message definitions is to show how the message is made up of a functional status set and a data set, and the data set is made up of a collection of data primitives. One line in the message structure is generally a data primitive, when possible. The boxes that represent a data primitive are intended to represent a group of bytes, the most significant byte on the left, the least significant byte on the right. Also, the bits in the bytes are most significant bit on the left and least significant bit on the right.

1-4.2 Message Format Example

Table 1-4 presents an example message format. It demonstrates how padding and spare can be applied to messages. The padding in this example maintains the alignment of the data elements.

**ATTACHMENT 1
DATA FORMAT**

Table 1-4 – Example Message Format

FDS Name	Aircraft Parameters	Data Primitives	Addr	Rate (Hz)	WKS
GPS PVT Data	Reserved	Reserved Word	0x0000	1	C_GPS_Data
DS1:	FSS	Functional Status Set	0x0004		
	GPS Sensor Mode	Integer_32	0x0008		
	GPS UTC of Time Mark	Float_32	0x000C		
	GPS UTC of Time Mark (Fine / Fractional)	Float_64	0x0010		
	GPS Date (Day)	Integer_32	0x0018		
	GPS Date (Month)	Integer_32	0x001C		
	GPS Date (Year)	Integer_32	0x0020		
	GPS Altitude	Float_32	0x0024		
	GPS Ground Speed	Float_32	0x0028		
	GPS Vertical Velocity	Float_32	0x002C		
	GPS N-S Velocity, True	Float_32	0x0030		
	GPS E-W Velocity, True	Float_32	0x0034		
	GPS Track Angle, True	Float_32	0x0038		
	Pad	Pad – 32	0x003C		
	GPS Present Position - Latitude	Float - 64	0x0040		
	GPS Present Position - Longitude	Float - 64	0x0048		
	GPS Horizontal Dilution Of Precision	Float_32	0x0050		
	GPS Horizontal Figure Of Merit	Float_32	0x0054		
	GPS Horizontal Uncertainty Limit	Float_32	0x0058		
	GPS Horizontal Integrity Limit	Float_32	0x005C		
	GPS RAIM Detected Satellite Fault	Integer_32	0x0060		
	Spare	Pad – 32	0x0064		
	GPS Vertical Dilution Of Precision	Float_32	0x0068		
	GPS Vertical Figure Of Merit	Float_32	0x006C		
	GPS Vertical Integrity Limit	Float_32	0x0070		
	GPS Sensor Mode Status	Integer_32	0x0074		
	GPS UTC of Time Mark Status	Integer_32	0x0078		
	GPS UTC of Time Mark (Fine / Fractional) Status	Integer_32	0x007C		
	GPS Date Status	Integer_32	0x0080		
	GPS Altitude Status	Integer_32	0x0084		
	GPS Ground Speed Status	Integer_32	0x0088		
	GPS Vertical Velocity Status	Integer_32	0x008C		
	GPS N-S Velocity, True Status	Integer_32	0x0090		
	GPS E-W Velocity, True Status	Integer_32	0x0094		
	GPS Track Angle, True Status	Integer_32	0x0098		
	GPS Present Position - Latitude Status	Integer_32	0x009C		
	GPS Present Position - Longitude Status	Integer_32	0x00A0		
	GPS Horizontal Dilution Of Precision Status	Integer_32	0x00A4		

**ATTACHMENT 1
DATA FORMAT**

FDS Name	Aircraft Parameters	Data Primitives	Addr	Rate (Hz)	WKS
	GPS Horizontal Figure Of Merit Status	Integer_32	0x00A8		
	GPS Horizontal Uncertainty Limit Status	Integer_32	0x00AC		
	GPS Horizontal Integrity Limit Status	Integer_32	0x00B0		
	GPS Vertical Dilution Of Precision Status	Integer_32	0x00B4		
	GPS Vertical Figure Of Merit Status	Integer_32	0x00B8		
	GPS Vertical Integrity Limit Status	Integer_32	0x00BC		
	Discretes	Boolean_32	0x00C0		
	Position 0	Boolean	0x00C0:0		
	Position 1	Boolean	0x00C0:1		
	Position 2	Boolean	0x00C0:2		
	Position 3	Boolean	0x00C0:3		
	Spare	Spare – 28	0x00C0:4-31		
	Spare	Spare – 32	0x00C4		
	Spare	Spare – 32	0x00C8		

In some cases a Boolean data primitive may be used to represent discretes. In Table 1-4 the example message has a packed Boolean sitting at address offset 0x00C0. Table 1-4 shows the definition of each bit in the Boolean that are being used for position discretes. Figure 1-4.2.1 shows the detail of how this Boolean data primitive is actually represented in memory. Packed Booleans are right justified, they begin filling in bits from right to left with spare on the left.

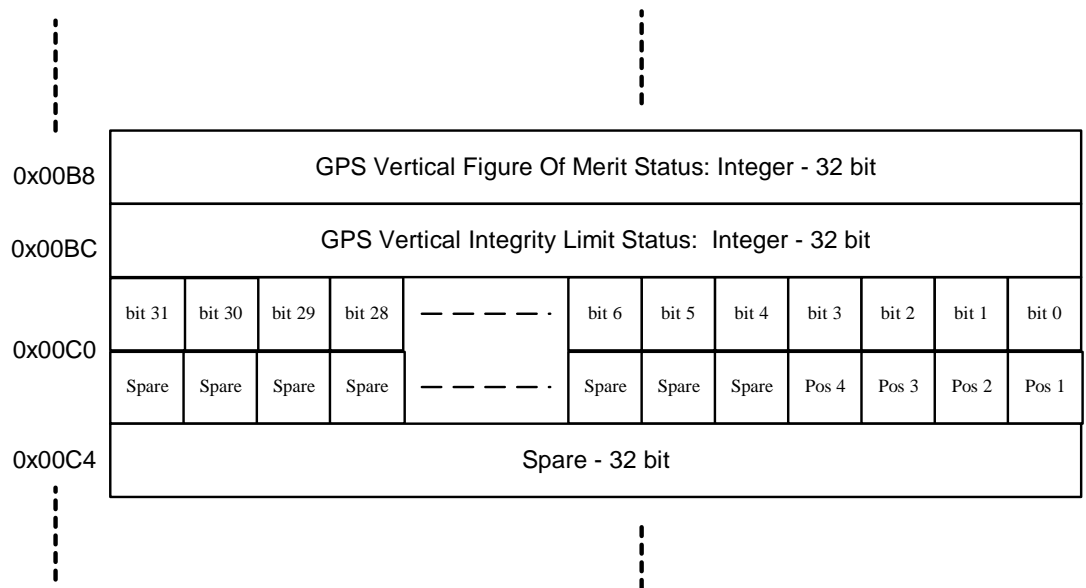


Figure 1-4.2.1 – Detail of Boolean From Message of Table 1-4

ATTACHMENT 2
IP/ICMP, UDP, AND TCP PROFILE PROVISIONS

Tables 2-1, 2-2, and 2-3 are duplicates of the tables presented in RFC 1122. These tables list the primary services or features available in each of the protocols. RFC-1122 provides, for each protocol, an explicit set of requirements, recommendations, and options, on a feature by feature basis. These indications are for commercial Ethernet networking. The following tables provide the same information for AFDX.

COMMENTARY

For AFDX, the column "OPTIONAL" is not used. It is replaced by "NOT APPLICABLE" in order to identify the requirements that do not apply to an AFDX End System, e.g., IP options and router requirements.

NOT APPLICABLE should be interpreted as MUST NOT.

Table 2-1 – Transport Layer TCP Requirements Summary (RFC-1122 Paragraph 4.2)

Transport Layer TCP Requirements Summary (RFC-1122 Paragraph 4.2)		INTERNET RFC1122	AFDX			COMMENTS
			MUST	NOT APPLICABLE	MUST NOT	
PUSH FLAG						
Aggregate or queue un-pushed data		4.2.2.2	X			
When an application issues a series of SEND calls without setting the PUSH flag, the TCP MAY aggregate the data internally without sending it. Similarly, when a series of Segments is received without the PSH bit, a TCP MAY queue the data internally without passing it to the receiving application.						
Sender collapse successive PUSH flags		4.2.2.2			X	
The transmitter SHOULD collapse Successive PSH bits when it packetizes data, to send the largest possible segment.						
SEND call can specify PUSH		4.2.2.2	X			
A TCP MAY implement PUSH flags on SEND calls...						
If cannot, sender buffer indefinitely		4.2.2.2			X	
...then the sending TCP: (1) must not buffer data indefinitely...						
If cannot, PUSH last segment		4.2.2.2	X			
...and (2) MUST set the PSH bit in the last buffered segment (i.e., when there is no more queued data to be sent).						
Notify receiving application of PUSH		4.2.2.2	X			
Passing a received PSH flag to the application layer is now OPTIONAL.						
Send maximum size segment when possible		4.2.2.2	X			
However, a TCP SHOULD send a maximum-sized segment whenever possible, to improve performance						
WINDOW						

ATTACHMENT 2
IP/ICMP, UDP, AND TCP PROFILE PROVISIONS

Transport Layer TCP Requirements Summary (RFC-1122 Paragraph 4.2)	INTERNET RFC1122	AFDX			COMMENTS
		MUST	NOT APPLICABLE	MUST NOT	
Treat as unsigned number	4.2.2.3	X			
The window size MUST be treated as an unsigned number.					
Handle as 32-bit number	4.2.2.3	X			
In anticipation of the adoption of such an extension, TCP implementors should treat windows as 32 bits.					
Shrink window from right	4.2.2.16			X	
A TCP receiver SHOULD NOT shrink the window, i.e., move the right window edge to the left.					
Robust against shrinking window	4.2.2.16	X			
However, a sending TCP MUST be robust against window shrinking, which may cause "useable window" (see Section 4.2.3.4) to become negative					
Receiver window closed indefinitely	4.2.2.17			X	
A TCP MAY keep its offered receive window closed indefinitely.					
Sender probe zero window	4.2.2.17	X			
Probing of zero (offered) windows MUST be supported.					
First probe after RTO (Note 1)	4.2.2.17	X			
The transmitting host SHOULD send the first zero-window probe when a zero window has existed for the retransmission timeout period.					
Exponential backoff	4.2.2.17	X			
The transmitting host SHOULD increase exponentially the interval between successive probes.					
Allow window stay zero indefinitely	4.2.2.17	X			
The sending TCP MUST allow the connection to stay open.					
Sender timeout OK connection with zero window	4.2.2.17			X	
The sending TCP MUST allow the connection to stay open.					
URGENT DATA					
Pointer points to last octet	4.2.2.4		X		
The urgent pointer points to the sequence number of the LAST octet (not LAST+1)					
Arbitrary length urgent data sequence	4.2.2.4		X		
A TCP MUST support a sequence of urgent data of any length.					
Inform application asynchronously of urgent data	4.2.2.4		X		
A TCP MUST inform the application layer asynchronously whenever it receives an Urgent pointer and there was previously no pending urgent data.					
Application can learn if/how much urgent data is queued			X		

ATTACHMENT 2
IP/ICMP, UDP, AND TCP PROFILE PROVISIONS

Transport Layer TCP Requirements Summary (RFC-1122 Paragraph 4.2)		INTERNET RFC1122	AFDX			COMMENTS
			MUST	NOT APPLICABLE	MUST NOT	
	There MUST be a way for the application to learn how much urgent data remains to be read from the connection, or at least to determine whether or not more urgent data remains to be read.					
TCP OPTIONS						
	Receive TCP Option in any segment	4.2.2.5	X			
	A TCP MUST be able to receive a TCP option in any segment.					
	Ignore unsupported options	4.2.2.5	X			
	A TCP MUST ignore without error any TCP option it does not implement, assuming that the option has a length field.					
	Cope with illegal option length	4.2.2.5	X			
	TCP MUST be prepared to handle an illegal option length.					
	Implement sending and receiving MSS option (Note 2)	4.2.2.6	X			
	TCP MUST implement both sending and receiving the Maximum Segment Size option.					
	Send MSS option unless 536	4.2.2.6	X			
	TCP SHOULD send an MSS (Maximum Segment Size) option in Every SYN segment when its receive MSS differs from the default 536,...					
	Send MSS option always	4.2.2.6			X	
	...and MAY send it always.					
	Send MSS default is 536	4.2.2.6	X			
	If an MSS option is not received at connection setup, TCP MUST assume a default send MSS of 536.					
	Calculate effective send segment size	4.2.2.6	X			
	The maximum size of a segment that TCP really sends, the "effective send MSS," MUST be the smaller of the send MSS (which reflects the available reassembly buffer size at the remote host) and the largest size permitted by the IP layer.					
TCP CHECKSUMS						
	Sender compute checksum	4.2.2.7	X			
	The sender MUST generate.					
	Receiver check checksum	4.2.2.7	X			
	The receiver MUST check it.					
	Use clock-driven ISN selection (Note 3)	4.2.2.9	X			
	A TCP MUST use the specified clock-driven selection of initial sequence numbers.					

ATTACHMENT 2
IP/ICMP, UDP, AND TCP PROFILE PROVISIONS

Transport Layer TCP Requirements Summary (RFC-1122 Paragraph 4.2)	INTERNET RFC1122	AFDX			COMMENTS
		MUST	NOT APPLICABLE	MUST NOT	
OPENING CONNECTIONS					
Support simultaneous open attempts	4.2.2.10	X			
A TCP MUST support simultaneous open attempts.					
SYN-RCVD remember last state	4.2.2.11	X			
TCP implementation MUST keep track of whether a connection has reached SYN_RCVD state as the result of a passive OPEN or an active OPEN.					
Passive open call interfere with others	4.2.2.18			X	
Every passive OPEN call it MUST NOT affect any previously created connection record.					
Function simulate LISTENs for same port	4.2.2.18	X			
A TCP that supports multiple concurrent users MUST provide an OPEN call that will functionally allow an application to LISTEN on a port while a connection block with the same local port is in SYN-SENT or SYN-RECEIVED state.					
Ask IP for source address for SYN if necessary	4.2.3.7	X			
The TCP MUST ask the IP layer to select a local IP.					
Otherwise, use local address of connection	4.2.3.7	X			
At all other times, a previous segment has either been sent Or received on this connection, and TCP MUST use the same local address is used that was used in those previous segments.					
OPEN to broadcast/multicast IP address	4.2.3.10	X			It is possible to receive a multicast OPEN. In this case, it must be rejected.
A TCP implementation MUST reject as an error a local OPEN call for an invalid remote IP address (e.g., a broadcast or multicast address).					
Silently discard segment to broadcast/multicast address	4.2.3.10	X			
A TCP implementation MUST silently discard an incoming SYN segment that is addressed to a broadcast or multicast address.					
CLOSING CONNECTIONS					
RST can contain data	4.2.2.12	X			
A TCP SHOULD allow a received RST segment to include data.					
Inform applications of aborted connection	4.2.2.13	X			
If a TCP connection is closed by the remote site, the local application MUST be informed whether it closed normally or was aborted.					
Half-duplex close connections	4.2.2.13			X	

ATTACHMENT 2
IP/ICMP, UDP, AND TCP PROFILE PROVISIONS

Transport Layer TCP Requirements Summary (RFC-1122 Paragraph 4.2)		INTERNET RFC1122	AFDX			COMMENTS
			MUST	NOT APPLICABLE	MUST NOT	
	A host MAY implement a "half-duplex" TCP close sequence, so that an application that has called CLOSE cannot continue to read data from the connection.					
	Send RST to indicate data lost	4.2.2.13	X			
	If such a host issues a CLOSE call while received data is still pending in TCP, or if new data is received after CLOSE is called, its TCP SHOULD send a RST to show that data was lost.					
	In TIME-WAIT state for 2xMSL seconds	4.2.2.13	X			
	When a connection is closed actively, it MUST linger in TIME-WAIT state for a time 2xMSL.					
	Accept SYN from TIME-WAIT state	4.2.2.13			X	
	It MAY accept a new SYN from the remote TCP to reopen the connection directly from TIME-WAIT state.					
RETRANSMISSIONS						
	Jacobson Slow Start algorithm	4.2.2.15			X	
	Recent work by Jacobson [TCP:7] on Internet congestion and TCP retransmission stability has produced a transmission algorithm combining "slow start" with "congestion avoidance". A TCP MUST implement this algorithm.					
	Jacobson Congestion-Avoidance algorithm	4.2.2.15			X	
	Recent work by Jacobson [TCP:7] on Internet congestion and TCP retransmission stability has produced a transmission algorithm combining "slow start" with "congestion avoidance". A TCP MUST implement this algorithm.					
	Retransmit with same IP ident	4.2.2.15			X	
	If a retransmitted packet is identical to the original packet (which implies not only that the data boundaries have not changed, but also that the window and acknowledgment fields of the header have not changed), then the same IP Identification field MAY be used.					
	Karn's algorithm	4.2.3.1			X	
	A host TCP MUST implement Karn's algorithm and Jacobson's algorithm for computing the retransmission timeout ("RTO").					
	Jacobson Retransmission Timeout estimation algorithm	4.2.3.1			X	
	A host TCP MUST implement Karn's algorithm and Jacobson's algorithm for computing the retransmission timeout ("RTO").					
	Exponential backoff	4.2.3.1			X	
	This implementation also MUST include "exponential backoff" for successive RTO values for the same segment.					
	SYN Retransmission Timeout calculation same as data	4.2.3.1	X			

ATTACHMENT 2
IP/ICMP, UDP, AND TCP PROFILE PROVISIONS

Transport Layer TCP Requirements Summary (RFC-1122 Paragraph 4.2)		INTERNET RFC1122	AFDX			COMMENTS
			MUST	NOT APPLICABLE	MUST NOT	
	Retransmission of SYN segments SHOULD use the same algorithm as data segments.					
	Recommended initial values and bounds	4.2.3.1	X			
	The following values SHOULD be used to initialize the estimation parameters for a new connection:					
	(a) RTT = 0 seconds.					
	(b) RTO = 3 seconds.					
	The recommended upper and lower bounds on the RTO are known to be inadequate on large internets. The lower bound SHOULD be measured in fractions of a second and the upper bound should be 2*MSL, i.e., 240 seconds.					
GENERATING ACKNOWLEDGMENTS						
	Queue out-of-order segments	4.2.2.20			X	
	A TCP SHOULD be capable of queueing out-of-order TCP segments.					
	Process all queued data before sending ACK	4.2.2.20	X			
	In general, the processing of received segments MUST be implemented to aggregate ACK segments whenever possible.					
	Send ACK for out-of-order segment	4.2.2.21			X	
	A TCP MAY send an ACK segment acknowledging RCV.NXT when a valid segment arrives that is in the window but not at the left window edge.					
	Delayed ACKs	4.2.3.2	X			
	A TCP SHOULD implement a delayed ACK, but an ACK should not be excessively delayed.					
	Delay less than 0.5 seconds	4.2.3.2	X			
	The delay MUST be less than 0.5 seconds.					
	Every second full-size segment ACK'd	4.2.3.2	X			
	In a stream of full-sized segments there SHOULD be an ACK for at least every second segment.					
	Receiver SWS-Avoidance algorithm (Note 4)	4.2.3.3	X			
	A TCP MUST include a SWS avoidance algorithm in the receiver.					
SENDING DATA						
	Configurable Time to Live	4.2.2.19		X		Not used in AFDX, default value is 1
	The TTL value used to send TCP segments MUST be configurable.					
	Sender SWS-Avoidance algorithm	4.2.3.6			X	
	A TCP MUST include a SWS avoidance algorithm in the sender.					
	Nagle algorithm	4.2.3.4			X	

ATTACHMENT 2
IP/ICMP, UDP, AND TCP PROFILE PROVISIONS

Transport Layer TCP Requirements Summary (RFC-1122 Paragraph 4.2)		INTERNET RFC1122	AFDX			COMMENTS
			MUST	NOT APPLICABLE	MUST NOT	
	A TCP SHOULD implement the Nagle Algorithm [TCP:9] to coalesce short segments.					
	Application can disable Nagle algorithm	4.2.3.4		X		
	There MUST be a way for an application to disable the Nagle algorithm on an individual connection.					
CONNECTION FAILURE						
	Negative advice to IP on R1 retransmissions (Note 5)	4.2.3.5	X			
	When the number of transmissions of the same segment reaches or exceeds threshold R1, pass negative advice to the IP layer, to trigger dead-gateway diagnosis.					
	Close connection on R2 retransmissions (Note 6)	4.2.3.5	X			
	When the number of transmissions of the same segment reaches a threshold R2 greater than R1, close the connection.					
	Application can set R2	4.2.3.5	X			
	An application MUST be able to set the value for R2 for a particular connection.					
	Inform application of $R1 \leq \text{retransmissions} < R2$	4.2.3.5			X	
	TCP SHOULD inform the application of the delivery Problem (unless such information has been disabled by the application; see Section 4.2.4.1), when R1 is reached and before R2.					
	Recommended values for R1 and R2	4.2.3.5	X			
	The value of R1 SHOULD correspond to at least 3 retransmissions, at the current RTO. The value of R2 SHOULD correspond to at least 100 seconds.					
	Same mechanism for SYN	4.2.3.5	X			
	SYN Retransmissions MUST be handled in the general way just described for data retransmissions, including notification of the application layer.					
	R2 at least 3 minutes for SYN	4.2.3.5			X	
	R2 for a SYN segment MUST be set large enough to provide retransmission of the segment for at least 3 minutes.					
SEND KEEP-ALIVE PACKETS		4.2.3.6			X	Connections are statically defined in AFDX. Therefore, keep alive packets are useless.
	Application can request	4.2.3.6		X		
	Implementors MAY include "Keep live" in their TCP implementation.					
	Default is "off"	4.2.3.6		X		

ATTACHMENT 2
IP/ICMP, UDP, AND TCP PROFILE PROVISIONS

Transport Layer TCP Requirements Summary (RFC-1122 Paragraph 4.2)		INTERNET RFC1122	AFDX			COMMENTS
			MUST	NOT APPLICABLE	MUST NOT	
	If keep-alives are included, they MUST default to off.					
	Only send if idle for interval	4.2.3.6		X		
	Keep-alive packets MUST only be sent when no data or Acknowledgement packets have been received for the connection within an interval.					
	Interval configurable	4.2.3.6		X		
	This interval MUST be configurable.					
	Default is at least 2 hours	4.2.3.6		X		
	If a keep-alive mechanism is implemented it MUST NOT interpret failure to respond to any specific probe as a dead connection.					
IP OPTIONS						
	Ignore options TCP doesn't understand	4.2.3.8	X			
	When received options are passed up to TCP from the IP layer, TCP MUST ignore options that it does not understand.					
	Time Stamp support	4.2.3.8			X	
	A TCP MAY support the Time Stamp.					
	Record Route support	4.2.3.8			X	
	A TCP MAY support the Record Route options.					
Source Route:						
	Application can specify Source Route	4.2.3.8			X	
	An application MUST be able to specify a source route when it actively opens a TCP connection.					
	Override Source Route in datagram	4.2.3.8			X	
	This MUST take precedence over a source route received in a datagram.					
	Build return route from Source Route	4.2.3.8			X	
	When a TCP connection is OPENed passively and a packet arrives with a completed IP Source Route option (containing a return route), TCP MUST save the return route and use it for all segments sent on this connection.					
	Later Source Route override	4.2.3.8			X	
	If a different source route arrives in a later segment, the later definition SHOULD override the earlier one.					
RECEIVING ICMP MESSAGES FROM IP						
	Destination Unreachable (0,1,5) sent to application	4.2.3.9		X		Only ICMP "echo request" and "echo reply" are allowed.
	TCP MUST react to a Source Quench by slowing transmission on the connection. The RECOMMENDED procedure is for a Source Quench to trigger a "slow start,"					

ATTACHMENT 2
IP/ICMP, UDP, AND TCP PROFILE PROVISIONS

Transport Layer TCP Requirements Summary (RFC-1122 Paragraph 4.2)		INTERNET RFC1122	AFDX			COMMENTS
			MUST	NOT APPLICABLE	MUST NOT	
	as if a retransmission timeout had occurred. Destination Unreachable -- codes 0, 1, 5.					
	Destination Unreachable (0,1,5) aborts connection	4.2.3.9		X		Only ICMP "echo request" and "echo reply" are allowed.
	Since these Unreachable messages indicate soft error conditions, TCP MUST NOT abort the connection.					
	Destination Unreachable (2,4) aborts connection	4.2.3.9		X		Only ICMP "echo request" and "echo reply" are allowed.
	Destination Unreachable -- codes 2-4 These are hard error conditions, so TCP SHOULD abort the connection.					
	Source Quench => slow star	4.2.3.9		X		Only ICMP "echo request" and "echo reply" are allowed.
	TCP MUST react to a Source Quench by slowing transmission on the connection. The RECOMMENDED procedure is for a Source Quench to trigger a "slow start," as if a retransmission timeout had occurred.					
	Time Exceeded => tell application, don't abort	4.2.3.9		X		Only ICMP "echo request" and "echo reply" are allowed.
	This should be handled the same way as Destination Unreachable codes 0, 1, 5.					
	Parameter Problem => tell application, don't abort	4.2.3.9		X		Only ICMP "echo request" and "echo reply" are allowed.
	This should be handled the same way as Destination Unreachable codes 0, 1, 5.					
ADDRESS VALIDATION						
	Reject OPEN call to invalid IP address	4.2.3.10	X			
	A TCP implementation MUST reject as an error a local OPEN call for an invalid remote IP address.					
	Reject SYN from invalid IP address	4.2.3.10	X			
	An incoming SYN with an invalid source address must be ignored either by TCP or by the IP layer.					
	Silently discard SYN to broadcast/multicast address	4.2.3.10	X			
	A TCP implementation MUST silently discard an incoming SYN segment that is addressed to a broadcast or multicast address.					
TCP/APPLICATION INTERFACE SERVICES						

ATTACHMENT 2
IP/ICMP, UDP, AND TCP PROFILE PROVISIONS

Transport Layer TCP Requirements Summary (RFC-1122 Paragraph 4.2)		INTERNET RFC1122	AFDX			COMMENTS
			MUST	NOT APPLICABLE	MUST NOT	
Error Report mechanisms		4.2.4.1	X			
	There MUST be a mechanism for reporting soft TCP error conditions to the application.					
Application can disable Error Report Routine		4.2.4.1			X	The application can ignore the error report.
	An application program that does not want to receive such ERROR_REPORT calls SHOULD be able to effectively disable these calls.					
Application can specify TOS for sending		4.2.4.2			X	TOS is not used by AFDX
	The application layer MUST be able to specify the Type-of-Service (TOS) for segments that are sent on a connection.					
Passed unchanged to IP		4.2.4.2		X		TOS is not used by AFDX
	TCP SHOULD pass the current TOS value without change to the IP layer, when it sends segments on the connection.					
Application can change TOS during connection		4.2.4.2			X	
	The application SHOULD be able to change the TOS during the connection lifetime.					
Pass received TOS up to application		4.2.4.2		X		TOS is not used by AFDX
	TCP MAY pass the most recently received TOS up to the Application.					
FLUSH call		4.2.4.2			X	
	Some TCP implementations have included a FLUSH call.					
Optional local IP address parameter in OPEN		4.2.4.4			X	
	The OPEN call MUST have an optional parameter.					
NOTES						
1 RTO: Retransmit timeout						
2 MSS: Maximum Segment Size						
3 ISN: Initial sequence number.						
4 SWS: Silly Window Syndrome						
5 R1: First retransmission threshold						
6 R2: Second retransmission threshold						

ATTACHMENT 2
IP/ICMP, UDP, AND TCP PROFILE PROVISIONS

Table 2-2 – Transport Layer UDP Requirements Summary (RFC-1122 Paragraph 4.1)

Transport Layer UDP Requirements Summary (RFC-1122 Paragraph 4.1)		INTERNET RFC1122	AFDX			COMMENTS
			MUST	NOT APPLICABLE	MUST NOT	
UDP send Port Unreachable		4.1.3.1		X		Only ICMP "echo request" and "echo reply" are allowed.
	If a datagram arrives addressed to a UDP port for which there is no pending LISTEN call, UDP SHOULD send an ICMP Port Unreachable message.					
IP Options in UDP						
Pass received IP options to application layer		4.1.3.2		X		IP options are not used in AFDX
	UDP MUST pass any IP option that it receives from the IP layer transparently to the application layer.					
Application layer can specify IP options in Send		4.1.3.2		X		IP options are not used in AFDX
	An application MUST be able to specify IP options to be sent in its UDP datagrams.					
UDP passes IP options down to IP layer		4.1.3.2		X		IP options are not used in AFDX
	UDP MUST pass these options to the IP layer.					
Pass ICMP messages up to application layer		4.1.3.3	X			Only for ICMP "echo request".
	UDP MUST pass to the application layer all ICMP error messages that it receives from the IP layer.					
UDP Checksums:						
Able to generate/check checksum		4.1.3.4		X		Checksum is not used in AFDX
	The host MUST implement the facility to generate and validate UDP checksums.					
Silently discard bad checksum		4.2.3.4		X		Checksum is not used in AFDX
	If a UDP datagram is received with a checksum that is non-zero and invalid, UDP MUST silently discard the datagram.					

ATTACHMENT 2
IP/ICMP, UDP, AND TCP PROFILE PROVISIONS

Transport Layer UDP Requirements Summary (RFC-1122 Paragraph 4.1)		INTERNET RFC1122	AFDX			COMMENTS
			MUST	NOT APPLICABLE	MUST NOT	
Sender option to not generate checksum		4.1.3.4		X		Checksum is not used in AFDX
	An application MAY optionally be able to control whether a UDP checksum will be generated...					
Default is to checksum		4.1.3.4		X		Checksum is not used in AFDX
	...but it MUST default to checksumming on.					
Receiver option is to require checksum		4.1.3.4		X		Checksum is not used in AFDX
	An application MAY optionally be able to control whether UDP datagrams without checksums should be discarded or passed to the application.					
						kkk
UDP Multihoming						
Pass spec-destination address to application		4.1.3.5			X	
	When a UDP datagram is received, its specific-destination address MUST be passed up to the application layer.					
Application layer can specify local IP address		4.1.3.5			X	
	An application program MUST be able to specify the IP source address to be used for sending a UDP datagram...					
Application layer specify wildcard local IP address		4.1.3.5			X	
	...or to leave it unspecified (in which case the networking software will choose an appropriate source address).					
Application layer notified of local IP address used		4.1.3.5			X	
	There SHOULD be a way to communicate the chosen source address up to the application layer (e.g, so that the application can later receive a reply datagram only from the corresponding interface).					
Bad IP source address silently discarded by UDP/IP		4.1.3.6			X	
	A UDP datagram received with an invalid IP source address (e.g., a broadcast or multicast address) must be discarded by UDP or by the IP layer.					
Only send valid IP source address		4.1.3.6			X	
	When a host sends a UDP datagram, the source address MUST be (one of) the IP address(es) of the host.					
UDP Application Interface Services						

ATTACHMENT 2
IP/ICMP, UDP, AND TCP PROFILE PROVISIONS

Transport Layer UDP Requirements Summary (RFC-1122 Paragraph 4.1)		INTERNET RFC1122	AFDX			COMMENTS
			MUST	NOT APPLICABLE	MUST NOT	
Full IP interface of 3.4 for application		4.1.4		X		These parameters are statically configured. Therefore Not Applicable
	The application interface to UDP MUST provide the full services of the IP/transport interface described in Section 3.4 of this document. Thus, an application using UDP needs the functions of the GET_SRCADDR(), GET_MAXSIZES(), ADVISE_DELIVPROB(), and RECV_ICMP() calls described in Section 3.4. For example, GET_MAXSIZES() can be used to learn the effective maximum UDP maximum datagram size for a particular {interface,remote host,TOS} triplet.					
Able to specify TTL, TOS and IP options when sending datagrams		4.1.4		X		These parameters are statically configured. Therefore Not Applicable
	An application-layer program MUST be able to set the TTL and TOS values as well as IP options for sending a UDP datagram, and these values must be passed transparently to the IP layer.					
Pass received TOS up to application layer		4.1.4			X	
	UDP MAY pass the received TOS up to the application layer.					

ATTACHMENT 2
IP/ICMP, UDP, AND TCP PROFILE PROVISIONS

Table 2-3 – Transport Layer IP Requirements Summary (RFC-1122 Paragraph 3.5)

Transport Layer IP Requirements Summary (RFC-1122 Paragraph 3.5)		INTERNET RFC 1122	AFDX			COMMENTS
			MUST	NOT APPLICABLE	MUST NOT	
Implement IP		3.1	X			
	The Internet layer of host software MUST implement IP. See Section 3.3.7 for the requirements on support of IGMP					
Implement ICMP		3.1	X			
	The Internet layer of host software MUST implement ICMP.					
Handle remote multihoming in application layer		3.1			X	
	At present, remote multihoming MUST be handled at the application Layer					
Support local multihoming		3.1	X			
	A host MAY support local multihoming,					
Meet gateway specifications if capable of forwarding datagrams		3.1		X		
	Any host that forwards datagrams generated by another host is acting as a gateway and MUST also meet the specifications laid out in the gateway requirements RFC [INTRO:2]					
Configuration switch for embedded gateway		3.1		X		
	An Internet host that includes embedded gateway code MUST have a configuration switch to disable the gateway function...					
Configuration default is non-gateway		3.1		X		There is no default configuration but A configuration. Therefore, No change
	...and this switch MUST default to the non-gateway mode.					
Autoconfiguration based on number of interfaces		3.1				
	The host software MUST NOT automatically move into gateway mode if the host has more than one interface.					
Able to log discarded datagrams		3.1	X			logged in MIB
	However, for diagnosis of problems a host SHOULD provide the capability of logging the error (see Section 1.2.3), including the contents of the silently-discarded datagram...					
Record in counter		3.1	X			logged in MIB
	...and SHOULD record the event in a statistics counter.					
Silently discard if IP version is not equal to 4		3.2.1.1	X			
	A datagram whose version number is not 4 MUST be					

**ATTACHMENT 2
IP/ICMP, UDP, AND TCP PROFILE PROVISIONS**

Transport Layer IP Requirements Summary (RFC-1122 Paragraph 3.5)	INTERNET RFC 1122	AFDX			COMMENTS
		MUST	NOT APPLICABLE	MUST NOT	
silently discarded.					
Verify IP Checksum, silently discard bad datagram	3.2.1.2	X			
A host MUST verify the IP header checksum on every received datagram and silently discard every datagram that has a bad checksum.					
Addressing:					
Subnet addressing (RFC-950)	3.2.1.3	X			
A host MUST support the subnet extensions to IP [IP:3].					
Source address must be host's own IP address	3.2.1.3	X			
When a host sends any datagram, the IP source address MUST be one of its own IP addresses (but not a broadcast or multicast address).					
Silently discard datagram with bad destination address	3.2.1.3	X			
A host MUST silently discard an incoming datagram that is not destined for the host.					
Silently discard datagram with bad source address	3.2.1.3	X			
A host MUST silently discard an incoming datagram containing an IP source address that is invalid by the rules of this section.					
Support reassembly	3.2.1.4	X			
The Internet model requires that every host support Reassembly.					
Retain same ID field in identical datagrams	3.2.1.5			X	RFC 1122 section 3.2.1.5 says this is an application issue
When sending an identical copy of an earlier datagram, a host MAY optionally retain the same Identification field in the copy.					
TOS:					
Allow transport layer to set TOS	3.2.1.6			X	
The IP layer MUST provide a means for the transport layer to set the TOS field of every datagram that is sent; the default is all zero bits.					
Pass received TOS up to transport layer	3.2.1.6			X	
The IP layer SHOULD pass received TOS values up to the transport layer.					
Use RFC-795 link-layer mappings for TOS	3.2.1.6			X	
The particular link-layer mappings of TOS contained in RFC- 795 SHOULD NOT be implemented.					
TTL:					
Send packet with TTL of 0	3.2.1.7			X	Must be set to 1
A host MUST NOT send a datagram with a Time-to-Live (TTL) value of zero .					

ATTACHMENT 2
IP/ICMP, UDP, AND TCP PROFILE PROVISIONS

Transport Layer IP Requirements Summary (RFC-1122 Paragraph 3.5)	INTERNET RFC 1122	AFDX			COMMENTS
		MUST	NOT APPLICABLE	MUST NOT	
Discard received packets with TTL < 2	3.2.1.7			X	
A host MUST NOT discard a datagram just because it was received with TTL less than 2.					
Allow transport layer to set TTL	3.2.1.7			X	
The IP layer MUST provide a means for the transport layer to set the TTL field of every datagram that is sent.					
Fixed TTL is configurable	3.2.1.7			X	
When a fixed TTL value is used, it MUST be configurable .					
IP Options:					
Allow transport layer to send IP options	3.2.1.8			X	
There MUST be a means for the transport layer to specify IP options to be included in transmitted IP datagrams.					
Pass all IP options received to higher layer	3.2.1.8			X	
All IP options (except NOP or END-OF-LIST) received in datagrams MUST be passed to the transport layer (or to ICMP processing when the datagram is an ICMP message).					
IP layer silently ignore unknown options	3.2.1.8	X			
The IP and transport layer MUST each interpret those IP options that they understand and silently ignore the others.					
Security option	3.2.1.8a			X	
Some environments require the Security option in every datagram.					
Send Stream Identifier option	3.2.1.8a			X	
This option is obsolete; it SHOULD NOT be sent...					
Silently ignore Stream Identifier option	3.2.1.8b	X			
...and it MUST be silently ignored if received.					
	3.2.1.8d		X		This spec applies to an ES only
Implementation of originating and processing the Record Route option is OPTIONAL.					
Timestamp option	3.2.1.8e		X		
Implementation of originating and processing the Timestamp option is OPTIONAL.					
Source Route Option:					
Originate and terminate Source Route Options	3.2.1.8c		X		No options
A host MUST support originating a source route and MUST be able to act as the final destination of a source route.					
Datagram with completed Source Route padded up to	3.2.1.8c		X		No options

ATTACHMENT 2
IP/ICMP, UDP, AND TCP PROFILE PROVISIONS

Transport Layer IP Requirements Summary (RFC-1122 Paragraph 3.5)		INTERNET RFC 1122	AFDX			COMMENTS
			MUST	NOT APPLICABLE	MUST NOT	
Transport Layer						
	If host receives a datagram containing a completed source route (i.e., the pointer points beyond the last field), the datagram has reached its final destination; the option as received (the recorded route) MUST be passed up to the transport layer (or to ICMP message processing).					
	Build correct (non-redundant) return route	3.2.1.8c		X		No options
	When a Return source route is built, it MUST be correctly formed even if the recorded route included the source host.					
	Send multiple Source Route options in one header	3.2.1.8c		X		No options
	An IP header containing more than one Source Route option MUST NOT be sent.					
Internet Control Message Protocol						
	Silently discard ICMP messages with unknown type	3.2.2	X			
	If an ICMP message of unknown type is received, it MUST be silently discarded.					
	Include more than 8 octets of original datagram	3.2.2			X	
	Include octets same as received	3.2.2			X	
	Every ICMP error message includes the Internet header and at least the first 8 data octets of the datagram that triggered the error; more than 8 octets MAY be sent.					
	Demultiplex ICMP Errors to transport protocol	3.2.2		X		No ICMP errors
	In those cases where the Internet layer is required to pass an ICMP error message to the transport layer, the IP protocol number MUST be extracted from the original header and used to select the appropriate transport protocol entity to handle the error.					
	Send ICMP error message with TOS=0	3.2.2		X		No ICMP errors
	An ICMP error message SHOULD be sent with normal (i.e., zero) TOS bits.					
	Send ICMP error message for:					
	- ICMP error message	3.2.2		X		No ICMP errors
	- IP broadcast or IP multicast	3.2.2		X		No ICMP errors
	- Link layer broadcast	3.2.2		X		No ICMP errors
	- Datagram with non-unique source address	3.2.2		X		No ICMP errors
	An ICMP error message MUST NOT be sent as the result of receiving:					
	* an ICMP error message, or					
	* a datagram destined to an IP broadcast or IP multicast address, or					
	* a datagram sent as a link-layer broadcast, or					

ATTACHMENT 2
IP/ICMP, UDP, AND TCP PROFILE PROVISIONS

Transport Layer IP Requirements Summary (RFC-1122 Paragraph 3.5)		INTERNET RFC 1122	AFDX			COMMENTS
			MUST	NOT APPLICABLE	MUST NOT	
	* a non-initial fragment, or					
	* a datagram whose source address does not define a single host -- e.g., a zero address, a loopback address, a broadcast address, a multicast address, or a Class address.					
Return ICMP error messages (when not prohibited)		3.2.2		X		No ICMP errors
	Wherever practical, hosts MUST return ICMP error datagrams on detection of an error, except in those cases where returning an ICMP error message is specifically prohibited.					
Destination unreachable:						
Generate /destination Unreachable (code 2/3)		3.2.2.1		X		No ICMP errors
	A host SHOULD generate Destination Unreachable messages with Code 2 and 3.					
Pass ICMP Destination Unreachable to higher layer		3.2.2.1		X		No ICMP errors
	A Destination Unreachable message that is received MUST be reported to the transport layer.					
Higher layer acts on Destination Unreachable		3.2.2.1		X		No ICMP errors
	The transport layer SHOULD use the information appropriately.					
Interpret Destination Unreachable as only a hint		3.2.2.1		X		No ICMP errors
	A Destination Unreachable message that is received with code 0 (Net), 1 (Host), or 5 (Bad Source Route) may result from a routing transient and MUST therefore be interpreted as only a hint.					
Redirect:						
Host send Redirect		3.2.2.2		X		No ICMP redirect from ES
	A host SHOULD NOT send an ICMP Redirect message; Redirects are to be sent only by gateways.					
Update route cache when receiving Redirect		3.2.2.2		X		No ICMP redirect from ES
	A host receiving a Redirect message MUST update its routing information accordingly.					
Handle both Host and Net Redirects		3.2.2.2		X		No ICMP redirect from ES
	Every host MUST be prepared to accept both Host and Network Redirects and to process them as described in Section 3.3.1.2 below.					
Discard illegal Redirect		3.2.2.2		X		No ICMP redirect from ES

ATTACHMENT 2
IP/ICMP, UDP, AND TCP PROFILE PROVISIONS

Transport Layer IP Requirements Summary (RFC-1122 Paragraph 3.5)		INTERNET RFC 1122	AFDX			COMMENTS
			MUST	NOT APPLICABLE	MUST NOT	
	A Redirect message SHOULD be silently discarded if the new gateway address it specifies is not on the same connected (sub-) net through which the Redirect arrived, or if the source of the Redirect is not the current first-hop gateway for the specified destination.					
Source Quench:						
Send Source Quench if buffering exceeded		3.2.2.3			X	
	A host MAY send a Source Quench message if it is approaching, or has reached, the point at which it is forced to discard incoming datagrams due to a shortage of reassembly buffers or other resources.					
Pass Source Quench to higher layer		3.2.2.3		X		Only ICMP "echo request" and "echo reply" are allowed.
	If a Source Quench message is received, the IP layer MUST report it to the transport layer (or ICMP processing).					
Higher layer act on Source Quench		3.2.2.3		X		Only ICMP "echo request" and "echo reply" are allowed.
	The transport or application layer SHOULD implement a mechanism to respond to Source Quench for any protocol that can send a sequence of datagrams to the same destination and which can reasonably be expected to maintain enough state information to make this feasible.					
Time Exceeded; pass to higher layer		3.2.2.4		X		Only ICMP "echo request" and "echo reply" are allowed.
	An incoming Time Exceeded message MUST be passed to the transport layer.					
Parameter Problem:						Only ICMP "echo request" and "echo reply" are allowed.
	A host SHOULD generate Parameter Problem messages. An incoming Parameter Problem message MUST be passed to the transport layer, and it MAY be reported to the user.					
Send parameter Problem messages		3.2.2.5		X		Only ICMP "echo request" and "echo reply" are allowed.
Pass parameter Problem to higher layer		3.2.2.5		X		Only ICMP "echo request" and "echo reply" are allowed.

ATTACHMENT 2
IP/ICMP, UDP, AND TCP PROFILE PROVISIONS

Transport Layer IP Requirements Summary (RFC-1122 Paragraph 3.5)	INTERNET RFC 1122	AFDX			COMMENTS
		MUST	NOT APPLICABLE	MUST NOT	
Report Parameter Problem to user	3.2.2.5		X		Only ICMP "echo request" and "echo reply" are allowed.
ICMP Echo Request or Reply					
Echo server and Echo client	3.2.2.6	X			
Every host MUST implement an ICMP Echo server function that receives Echo Requests and sends corresponding Echo Replies.					
Echo client	3.2.2.6	X			
A host SHOULD also implement an application-layer interface for sending an Echo Request and receiving an Echo Reply, for diagnostic purposes.					
Discard Echo Request to broadcast address	3.2.2.6	X			
An ICMP Echo Request destined to an IP broadcast or IP multicast address MAY be silently discarded.					
Discard Echo Request to multicast address	3.2.2.6	X			
An ICMP Echo Request destined to an IP broadcast or IP multicast address MAY be silently discarded.					
Use specific destination address as Echo Reply source	3.2.2.6	X			
The IP source address in an ICMP Echo Reply MUST be the same as the specific-destination address (defined in Section 3.2.1.3) of the corresponding ICMP Echo Request message.					
Send same data to Echo Reply	3.2.2.6	X			
Data received in an ICMP Echo Request MUST be entirely included in the resulting Echo Reply.					
Pass Echo Reply to higher layer	3.2.2.6	X			
Echo Reply messages MUST be passed to the ICMP user interface, unless the corresponding Echo Request originated in the IP layer.					
Reflect record Rout, Time Stamp options	3.2.2.6		X		Only ICMP "echo request" and "echo reply" are allowed.
If a Record Route and/or Time Stamp option is received in an ICMP Echo Request, this option (these options) SHOULD be updated to include the current host and included in the IP header of the Echo Reply message, without "truncation" Thus, the recorded route will be for the entire round trip.					
Reverse and reflect Source Route options	3.2.2.6		X		Only ICMP "echo request" and "echo reply" are allowed.

**ATTACHMENT 2
IP/ICMP, UDP, AND TCP PROFILE PROVISIONS**

Transport Layer IP Requirements Summary (RFC-1122 Paragraph 3.5)		INTERNET RFC 1122	AFDX			COMMENTS
			MUST	NOT APPLICABLE	MUST NOT	
	If a Source Route option is received in an ICMP Echo Request, the return route MUST be reversed and used as a Source Route option for the Echo Reply message.					
	ICMP Information Request or Reply	3.2.2.7		X		Only ICMP "echo request" and "echo reply" are allowed.
	A host SHOULD NOT implement these messages.					
	ICMP Timestamp and Timestamp Reply:	3.2.2.8		X		Only ICMP "echo request" and "echo reply" are allowed.
	A host MAY implement Timestamp and Timestamp Reply.					
	Minimize delay variability	3.2.2.8		X		Only ICMP "echo request" and "echo reply" are allowed.
	If this function is implemented, it SHOULD be designed for minimum variability in delay.					
	Silently discard multicast Timestamp	3.2.2.8		X		Only ICMP "echo request" and "echo reply" are allowed.
	An ICMP Timestamp Request message to an IP broadcast or IP multicast address MAY be silently discarded.					
	Silently discard multicast Timestamp	3.2.2.8		X		Only ICMP "echo request" and "echo reply" are allowed.
	An ICMP Timestamp Request message to an IP broadcast or IP multicast address MAY be silently discarded.					
	Use specific destination as Timestamp Reply source	3.2.2.8		X		Only ICMP "echo request" and "echo reply" are allowed.
	The IP source address in an ICMP Timestamp Reply MUST be the same as the specific-destination address of the corresponding Timestamp Request message.					
	Reflect Record Route, Time Stamp options	3.2.2.6		X		Only ICMP "echo request" and "echo reply" are allowed.
	If a Source-route option is received in an ICMP Echo Request, the return route MUST be reversed and used as a Source Route option for the Timestamp Reply message.					
	Reverse and reflect Source Route option	3.2.2.8		X		Only ICMP "echo request" and "echo

ATTACHMENT 2
IP/ICMP, UDP, AND TCP PROFILE PROVISIONS

Transport Layer IP Requirements Summary (RFC-1122 Paragraph 3.5)	INTERNET RFC 1122	AFDX			COMMENTS
		MUST	NOT APPLICABLE	MUST NOT	
					reply" are allowed.
If a Record Route and/or Timestamp option is received in a Timestamp Request, this (these) option(s) SHOULD be updated to include the current host and included in the IP header of the Timestamp Reply message.					
Pass Timestamp Reply to higher layer	3.2.2.8		X		Only ICMP "echo request" and "echo reply" are allowed.
Incoming Timestamp Reply messages MUST be passed up to the ICMP user interface.					
Obey rules for "standard value"	3.2.2.8		X		Only ICMP "echo request" and "echo reply" are allowed.
The preferred form for a timestamp value (the "standard value") is in units of milliseconds since midnight Universal Time.					
ICMP Address Mask Request and Reply					
Address Mask source configurable	3.2.2.9		X		Only ICMP "echo request" and "echo reply" are allowed.
The choice of method to be used in a particular host MUST be configurable...					
Support static configuration of address mask	3.2.2.9		X		Only ICMP "echo request" and "echo reply" are allowed.
(1) static configuration information...					
Get address mask dynamically during booting	3.2.2.9		X		Only ICMP "echo request" and "echo reply" are allowed.
(2) obtaining the address mask(s) dynamically as a side-effect of the system initialisation process...					
Get address via ICMP Address Mast Request/Reply	3.2.2.9		X		Only ICMP "echo request" and "echo reply" are allowed.
(3) sending ICMP Address Mask Request(s) and receiving ICMP Address Mask Reply(s).					
Retransmit Address Mask Request if no Reply	3.2.2.9		X		Only ICMP "echo request" and "echo reply" are allowed.
When method 3, it MUST retransmit this message a small number of times if it does not receive an immediate Address Mask Reply.					

**ATTACHMENT 2
IP/ICMP, UDP, AND TCP PROFILE PROVISIONS**

Transport Layer IP Requirements Summary (RFC-1122 Paragraph 3.5)	INTERNET RFC 1122	AFDX			COMMENTS
		MUST	NOT APPLICABLE	MUST NOT	
Assume default mask if no Reply	3.2.2.9		X		Only ICMP "echo request" and "echo reply" are allowed.
When method 3, until it has received an Address Mask Reply, the host SHOULD assume a mask appropriate for the address class of the IP address.					
Update Address Mask from first Reply only	3.2.2.9		X		Only ICMP "echo request" and "echo reply" are allowed.
When method 3, the first Address Mask Reply message received MUST be used to set the address mask corresponding to the particular local IP address.					
Reasonableness check on Address Mask	3.2.2.9		X		Only ICMP "echo request" and "echo reply" are allowed.
A host SHOULD make some reasonableness check on any address mask it installs.					
Send unauthorized Address Mask Reply message	3.2.2.9		X		Only ICMP "echo request" and "echo reply" are allowed.
A system MUST NOT send an Address Mask Reply unless it is an authoritative agent for address masks.					
Explicitly configured to be agent	3.2.2.9		X		Only ICMP "echo request" and "echo reply" are allowed.
An authoritative agent may be a host or a gateway, but it MUST be explicitly configured as a address mask agent.					
Static configuration-> Address Mask Authorization flag	3.2.2.9		X		Only ICMP "echo request" and "echo reply" are allowed.
With a statically configured address mask, there SHOULD be an additional configuration flag that determines whether the host is to act as an authoritative agent for this mask.					
Broadcast Address Mask Reply when initiated	3.2.2.9		X		Only ICMP "echo request" and "echo reply" are allowed.
If it is configured as an agent, the host MUST broadcast an Address Mask Reply for the mask on the appropriate interface when it initializes.					
ROUTING OUTBOUND DATAGRAMS					
Use address mask in local/remote decision	3.3.1.1.		X		The IP address of the gateway (if any) will be

ATTACHMENT 2
IP/ICMP, UDP, AND TCP PROFILE PROVISIONS

Transport Layer IP Requirements Summary (RFC-1122 Paragraph 3.5)	INTERNET RFC 1122	AFDX			COMMENTS
		MUST	NOT APPLICABLE	MUST NOT	
					in the configuration table.
To decide if the destination is on a connected network, the following algorithm MUST be used.					
Operate with no gateways on connected network	3.3.1.1	X			
The host IP layer MUST operate correctly in a minimal network environment, and in particular, when there are no gateways.					
Maintain "route cache" of next-hop gateway	3.3.1.2		X		AFDX is statically defined
To efficiently route a series of datagrams to the same destination, the source host MUST keep a "route cache" of mappings to next-hop gateways.					
Treat Host and Net Redirect the same	3.3.1.2		X		Only ICMP "echo request" and "echo reply" are allowed.
Since the subnet mask appropriate to the destination address is generally not known, a Network Redirect message SHOULD be treated identically to a Host Redirect message.					
If no cache entry, use default gateway	3.3.1.2		X		AFDX is statically defined
The IP layer MUST pick a gateway from its list of "default" gateways.					
Support multiple default gateways	3.3.1.2		X		AFDX is statically defined
The IP layer MUST support multiple default gateways.					
Provide table of static routes	3.3.1.2		X		AFDX is statically defined
As an extra feature, a host IP layer MAY implement a table of "static routes".					
Flag: route over-rideable by Redirect	3.3.1.2		X		Only ICMP "echo request" and "echo reply" are allowed.
Each such static route MAY include a flag specifying whether it may be overridden by ICMP Redirects.					
Key route case on host, not network address	3.3.1.3		X		AFDX is statically defined, no default gateways
Each route cache entry needs to include the following fields:					
(1) Local IP address (for a multihomed host)					

ATTACHMENT 2
IP/ICMP, UDP, AND TCP PROFILE PROVISIONS

Transport Layer IP Requirements Summary (RFC-1122 Paragraph 3.5)	INTERNET RFC 1122	AFDX			COMMENTS
		MUST	NOT APPLICABLE	MUST NOT	
(2) Destination IP address					
(3) Type(s)-of-Service					
(4) Next-hop gateway IP address					
Include TOS in route cache	3.3.1.3		X		AFDX is statically defined, no default gateways
The TOS SHOULD be included.					
Able to detect failure of next-hop gateway	3.3.1.4		X		AFDX is statically defined, no default gateways
The IP layer MUST be able to detect the failure of a "next-hop" gateway that is listed in its route cache and to choose an alternate gateway.					
Assume route is good forever	3.3.1.4		X		AFDX is statically defined, no default gateways
A particular gateway SHOULD NOT be used indefinitely in the absence of positive indications that it is functioning.					
Ping gateways continuously	3.3.1.4		X		AFDX is statically defined, no default gateways
In particular, hosts MUST NOT actively check the status of a first-hop gateway by simply pinging the gateway continuously.					
Ping only when traffic being sent	3.3.1.4		X		AFDX is statically defined, no default gateways
Pinging MUST be used only when traffic is being sent to the gateway...					
Ping only when no positive indication	3.3.1.4		X		AFDX is statically defined, no default gateways
...and when there is no other positive indication to suggest that the gateway is functioning.					
Higher and lower layers give advice	3.3.1.4		X		AFDX is statically defined, no default gateways
To avoid pinging, the layers above and/or below the Internet layer SHOULD be able to give "advice" on the status of route cache entries when either positive (gateway OK) or negative (gateway dead) information is available.					
Switch from failed default gateway to another	3.3.1.5		X		AFDX is statically defined, no default

ATTACHMENT 2
IP/ICMP, UDP, AND TCP PROFILE PROVISIONS

Transport Layer IP Requirements Summary (RFC-1122 Paragraph 3.5)		INTERNET RFC 1122	AFDX			COMMENTS
			MUST	NOT APPLICABLE	MUST NOT	
						gateways
	If the failed gateway is not the current default, the IP layer can immediately switch to a default gateway. If it is the current default that failed, the IP layer MUST select a different default gateway.					
	Manual method of entering configuration information	3.3.1.6		X		AFDX is statically defined, no default gateways
	A manual method of entering this configuration data MUST be provided.					
REASSEMBLY AND FRAGMENTATION:						
	Able to reassemble incoming datagrams	3.3.2	X			
	The IP layer MUST implement reassembly of IP datagrams.					
	At least 576 bytes datagrams	3.3.2	X			
	EMTU_R configurable or indefinite (Note 7)	3.3.2	X			Defined by system integrator
	We designate the largest datagram size that can be reassembled by EMTU_R ("Effective MTU to receive"); this is sometimes called the "reassembly buffer size". EMTU_R MUST be greater than or equal to 576, SHOULD be either configurable or indefinite, and SHOULD be greater than or equal to the MTU of the connected network(s).					
	Transport layer able to learn MMS_R	3.3.2		X		Defined by system integrator
	There MUST be a mechanism by which the transport layer can learn MMS_R.					
	Send ICMP Time Exceeded on reassembly timeout	3.3.2		X		Only ICMP "echo request" and "echo reply" are allowed.
	If this timeout expires, the partially-reassembled datagram MUST be discarded and an ICMP Time Exceeded message sent to the source host (if fragment zero has been received).					
	Fixed reassembly timeout value	3.3.2		X		Time out is not used for reassembly
	The TCP specification [TCP:1] arbitrarily assumes a value of 2 minutes for MSL. This sets an upper limit on a reasonable reassembly timeout value.					
	Pass MMS_S to higher layers (Note 8)	3.3.3			X	
	A host MUST implement a mechanism to allow the transport layer to learn MMS_S.					
	Local fragmentation of outgoing packets	3.3.3	X			

**ATTACHMENT 2
IP/ICMP, UDP, AND TCP PROFILE PROVISIONS**

Transport Layer IP Requirements Summary (RFC-1122 Paragraph 3.5)	INTERNET RFC 1122	AFDX			COMMENTS
		MUST	NOT APPLICABLE	MUST NOT	
Optionally, the IP layer MAY implement a mechanism to fragment outgoing datagrams intentionally.					
Else don't send bigger than MMS_S	3.3.3			X	
A host that does not implement local fragmentation MUST ensure that the transport layer (for TCP) or the application layer (for UDP) obtains MMS_S from the IP layer and does not send a datagram exceeding MMS_S in size.					
Send max 576 to off-net destination	3.3.3			X	Defined by system integrator
In the absence of actual knowledge of the minimum MTU along the path, the IP layer SHOULD use EMTU_S <= 576 whenever the destination address is not on a connected network, and otherwise use the connected network's.					
All-Subnets-MTU configuration flag	3.3.3			X	
A host IP layer implementation MAY have a configuration flag "All-Subnets-MTU", indicating that the MTU of the connected network is to be used for destinations on different subnets within the same network, but not for other networks.					
MULTIHOMING:					
Reply with same address as specified destination address	3.3.4.2	X			
(1) If the datagram is sent in response to a received datagram, the source address for the response SHOULD be the specific-destination address of the request.					
Allow application to choose local IP address	3.3.4.2	X			
(2) An application MUST be able to explicitly specify the source address for initiating a connection or a request.					
Silently discard datagram in "wrong" interface	3.3.4.2	X			
(A) A host MAY silently discard an incoming datagram whose destination address does not correspond to the physical interface through which it is received.					
Only send datagram through "right" interface	3.3.4.2	X			
(B) A host MAY restrict itself to sending (non-source-routed) IP datagrams only through the physical interface that corresponds to the IP source address of the datagrams.					
SOURCE-ROUTE FORWARDING:					
Forward datagram with Source Route Option	3.3.5		X		Source routing not allowed

ATTACHMENT 2
IP/ICMP, UDP, AND TCP PROFILE PROVISIONS

Transport Layer IP Requirements Summary (RFC-1122 Paragraph 3.5)		INTERNET RFC 1122	AFDX			COMMENTS
			MUST	NOT APPLICABLE	MUST NOT	
	Subject to restrictions given below, a host MAY be able to act as an intermediate hop in a source route, forwarding a source-routed datagram to the next specified hop.					
	Obey corresponding gateway rules	3.3.5		X		This spec applies to an ES only
	However, in performing this gateway-like function, the host MUST obey all the relevant rules for a gateway forwarding source-routed datagrams.					
	Update TTL by gateway rules	3.3.5		X		This spec applies to an ES only
	The TTL field MUST be decremented and the datagram.					
	Able to generate ICMP error codes 4 and 5	3.3.5		X		Only ICMP "echo request" and "echo reply" are allowed.
	A host MUST be able to generate Destination Unreachable messages with the following codes:					
	4 (Fragmentation Required but DF Set) when a source-routed datagram cannot be fragmented to fit into the target network					
	5 (Source Route Failed) when a source-routed datagram cannot be forwarded, e.g., because of a routing problem or because the next hop of a strict source route is not on a connected network.					
	IP source address not local host	3.3.5		X		This spec applies to an ES only
	A source-routed datagram being forwarded MAY (and normally will) have a source address that is not one of the IP addresses of the forwarding host.					
	Update Timestamp, Record Route options	3.3.5		X		Options are not used.
	A host that is forwarding a source-routed datagram containing a Record Route option MUST update that option, if it has room.					
	A host that is forwarding a source-routed datagram containing a Timestamp Option MUST add the current timestamp to that option, according to the rules for this option.					
	Configurable switch for non-local Source Routing	3.3.5		X		This spec applies to an ES only
	A host that supports non-local source-routing MUST have a configurable switch to disable forwarding...					
	Default to OFF	3.3.5		X		This spec applies to an ES only

ATTACHMENT 2
IP/ICMP, UDP, AND TCP PROFILE PROVISIONS

Transport Layer IP Requirements Summary (RFC-1122 Paragraph 3.5)	INTERNET RFC 1122	AFDX			COMMENTS
		MUST	NOT APPLICABLE	MUST NOT	
...and this switch MUST default to disabled.					
Satisfy gateway access rules for non-local Source Routing	3.3.5		X		This spec applies to an ES only
The host MUST satisfy all gateway requirements for configurable policy filters restricting non-local forwarding.					
If not forward, send Destination Unreachable (code 5)	3.3.5		X		Only ICMP "echo request" and "echo reply" are allowed.
If a host receives a datagram with an incomplete source route but does not forward it for some reason, the host SHOULD return an ICMP Destination Unreachable (code 5, Source Route Failed) message, unless the datagram was itself an ICMP error message.					
BROADCAST:					
Broadcast address as IP source address	3.2.1.3			X	
Section 3.2.1.3 defined the four standard IP broadcast address forms:					
* Limited Broadcast: {-1, -1}					
* Directed Broadcast: {<Network-number>,-1}					
* Subnet Directed Broadcast: {<Network-number>,<Subnet-number>,-1}					
* All-Subnets Directed Broadcast: {<Network-number>,-1,-1}					
A host MUST recognize any of these forms in the destination address of an incoming datagram.					
Receive 0 or 1 broadcast formats OK	3.3.6			X	
There is a class of hosts that use non-standard broadcast address forms, substituting 0 for -1. All hosts SHOULD recognize and accept any of these non-standard broadcast addresses as the destination address of an incoming datagram.					
Configurable option to send 0 or 1 broadcasts	3.3.6			X	
A host MAY optionally have a configuration option to choose the 0 or the -1 form of broadcast address, for each physical interface...					
Default to 1 broadcast	3.3.6			X	
...but this option SHOULD default to the standard (-1) form.					
Recognize all broadcast address formats	3.3.6			X	
A host MUST recognize any of these forms in the destination address of an incoming datagram.					
Use IP broadcast/multicast address in link-layer broadcast	3.3.6			X	

ATTACHMENT 2
IP/ICMP, UDP, AND TCP PROFILE PROVISIONS

Transport Layer IP Requirements Summary (RFC-1122 Paragraph 3.5)		INTERNET RFC 1122	AFDX			COMMENTS
			MUST	NOT APPLICABLE	MUST NOT	
	When a host sends a datagram to a link-layer broadcast address, the IP destination address MUST be a legal IP broadcast or IP multicast address.					
	Silently discard link-layer-only broadcast datagrams	3.3.6			X	
	A host SHOULD silently discard a datagram that is received via a link-layer broadcast (see Section 2.4) but does not specify an IP multicast or broadcast destination address.					
	Use Limited Broadcast address for connected network	3.3.6			X	
	Hosts SHOULD use the Limited Broadcast address to broadcast to a connected network.					
MULTICAST:						
	Support local IP multicasting (RFC-1112)	3.3.7	X			
	A host SHOULD support local IP multicasting on all connected networks for which a mapping from Class D IP addresses to link-layer addresses has been specified.					
	Support IGMP (RFC-1112)	3.3.7			X	
	Support for local IP multicasting includes sending multicast datagrams, joining multicast groups and receiving multicast datagrams, and leaving multicast groups. This implies support for all of except the IGMP protocol itself, which is OPTIONAL .					
	Join all-hosts group at startup	3.3.7			X	
	If IGMP is not implemented, a host SHOULD still join the "all-hosts" group (224.0.0.1) when the IP layer is initialized and remain a member for as long as the IP layer is active.					
	Higher layers learn interface multicast capability	3.3.7			X	
	A host SHOULD provide a way for higher-layer protocols or applications to determine which of the host's connected network(s) support IP multicast addressing.					
INTERFACES:						
	Allow transport layer to use all IP mechanisms	3.4			X	
	The interface between the IP layer and the transport layer MUST provide full access to all the mechanisms of the IP layer, including options, Type-of-Service, and Time-to-Live.					
	Pass interface identification up to transport layer	3.4			X	
	The transport layer MUST either have mechanisms to set these interface parameters, or provide a path to pass them through from an application, or both.					
	Pass all IP options up to transport layer	3.4			X	

ATTACHMENT 2
IP/ICMP, UDP, AND TCP PROFILE PROVISIONS

Transport Layer IP Requirements Summary (RFC-1122 Paragraph 3.5)		INTERNET RFC 1122	AFDX			COMMENTS
			MUST	NOT APPLICABLE	MUST NOT	
	The parameter opt contains all the IP options received in the datagram; these MUST also be passed to the transport layer.					
	Transport layer can send certain ICMP messages	3.4			X	
	The transport layer MUST be able to send certain ICMP Messages.					
	Pass specified ICMP messages up to transport layer	3.4			X	
	The IP layer MUST pass certain ICMP messages up to the appropriate transport-layer routine.					
	Include IP header + 8-octets or more from original	3.4			X	
	For an ICMP error message, the data that is passed up MUST include the original Internet header plus all the octets of the original message that are included in the ICMP message.					
	Able to leap tall buildings in a single bound	3.4		X		
	7 EMTU_R - Effective Maximum Transfer Unit to Receive. RFC-1122 paragraph 3.3.2					
	8 MMS_R - Maximum Message Size that can be received and reassembled. RFC-1122 paragraph 3.3.2					
	Comment: The TCP profile uses "Not Applicable" which is less confusing than "NO" [gvb]					

APPENDIX A AN EXAMPLE OF ES IDENTIFICATION

The equipment which hosts one or several End System(s) is identified in the network by:

- Domain ID (functional grouping)
- Side ID
- Location ID

The Domain ID indicates the domain (functional grouping) to which the equipment belongs.

The Side ID indicates the side of the equipment within the Domain.

The Location ID indicates the position of the equipment relatively to the side in the domain.

Domain ID, Side ID and the Location ID are used to build the IP and MAC addresses, based upon a 12-bit user ID, as in Figure A-1.

Ethernet MAC Controller Identification (48-bits)						
Constant field: 24-bits	User Defined ID 16 bits				Interface_ID	Constant field: 5-bits
	Constant field: 4-bits	Domain_ID	Side_ID	Location_ID		
"0000 0010 0000 0000 0000 0000"	"0000"	4-bits	3-bits	5-bits	3-bits	"0 0000"

IP Unicast Addressing Format (source or unicast destination) 32-bits							
Class A 1-bit	Private IP address 7-bits	User Defined ID 16-bits				Partition ID 8-bits	
		Constant field: 4-bits	Domain_ID	Side_ID	Location_ID		
"0"	"0001010"	"0000"	4-bits	3-bits	5-bits	"000"	5-bits

Figure A-1 – Construction of MAC and IP Addresses

The Domain ID should be coded with 4 bits.

The 0000 and 1111 are forbidden values. Therefore, 14 Domains ID are possible.

The Side ID format should be coded with 3 bits.

The 000 and 111 are forbidden values.

The Location ID format should be coded with 5 bits.

The 00000 and 11111 are forbidden values.

The Domain ID, Side ID and Location ID will be specified for each hosting equipment.

APPENDIX B
GUIDELINES FOR ARINC 429 TO AFDX FORMATTING

ARINC 429 devices will be used on aircraft for some time to come. AFDX networks must have a means of transporting data from/to these devices. This section provides examples of message definitions for packaging ARINC 429 labels into AFDX messages.

The preferred way to put ARINC 429 data on an AFDX network is to convert the individual parameters into one of the permitted AFDX data elements of Table 1-1.0. Every effort should be made to do this at the gateway where ARINC 429 and AFDX meet. This helps ensure that as much as possible, all data on the network is being represented consistently. When this is not possible for some reason, at least some level of commonality can be achieved if the following approaches are used.

Two general approaches are presented:

1. Place each label in a fixed length 32-bit opaque data primitive
2. Place a varying number of labels in a variable length opaque structure

Approach 1:

Figure B-1.1 is an example of how a single label would be placed in a message. This structure should be used for single label messages. Note that the FS represents the status of the gateway that is introducing the ARINC 429 data to the AFDX network. The status of the ARINC 429 device is presented in the label.

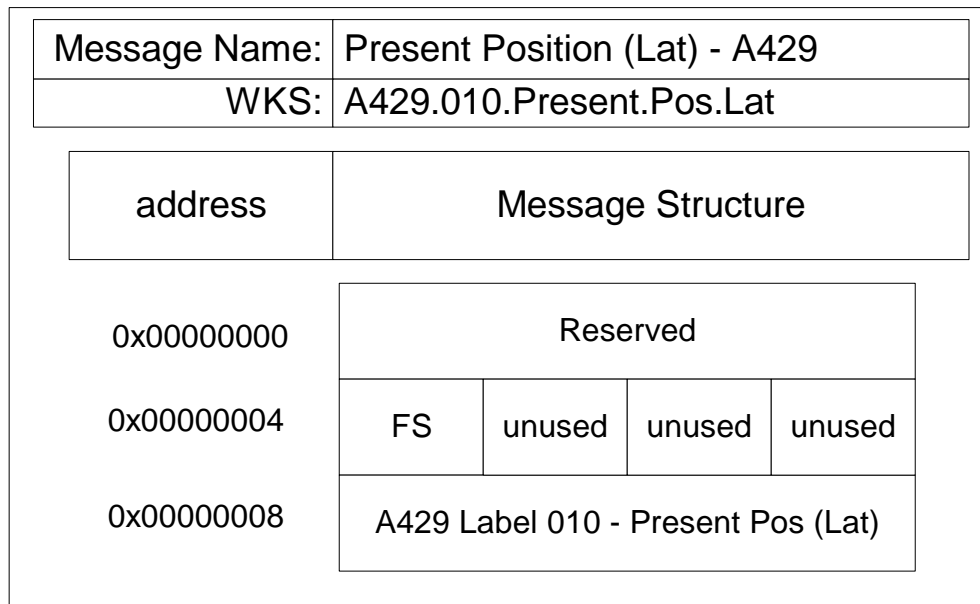


Figure B-1.1 – Message Format for Single Label Messages

APPENDIX B
GUIDELINES FOR ARINC 429 TO AFDX FORMATTING

Figures B-1.2 and B-1.3 show an example of a message format with multiple labels. ARINC 429 labels are placed on the network by means of concentrator devices such as:

- RDC (Remote Data Concentrator) – Concentrates discretes/ARINC 429/etc
- IOC (Input/Output Concentrator) – Concentrates analogs/discretes/ARINC 429/etc
- RIU (Radio Interface Unit) – Communicates ARINC 429 to/from radios
- P5 (Cockpit overhead panel) – Uses ARINC 429 to communicate with control panels

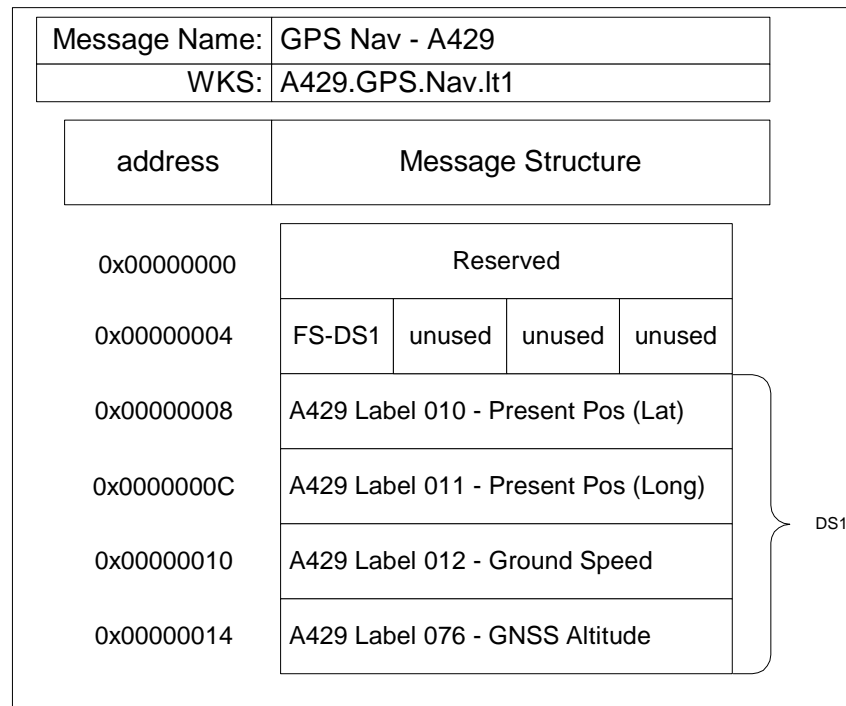


Figure B-1.2 – Message Format for Multiple Label Using 1 DS

When these devices pack ARINC 429 labels into an Ethernet frame, the status of the devices that generated the ARINC 429 labels is indicated in the individual labels by means of the SSM field. In this manner, the status of each ARINC 429 device is indicated in the frame. The FS is used to indicate the status of the concentrator.

If all labels are concentrated into a single Ethernet frame by a single threaded device, a single DS should be used, and the DS status is reflected by a single FS. If the concentrator device is designed in such a way that a portion of it could fail in some way that causes some labels to be invalid, but a completely legal Ethernet frame is still generated with some valid labels, more than one DS should be considered. This way each DS is represented by its own FS. This allows some good data to be received and used even in the presence of a fault and some invalid data.

APPENDIX B
GUIDELINES FOR ARINC 429 TO AFDX FORMATTING

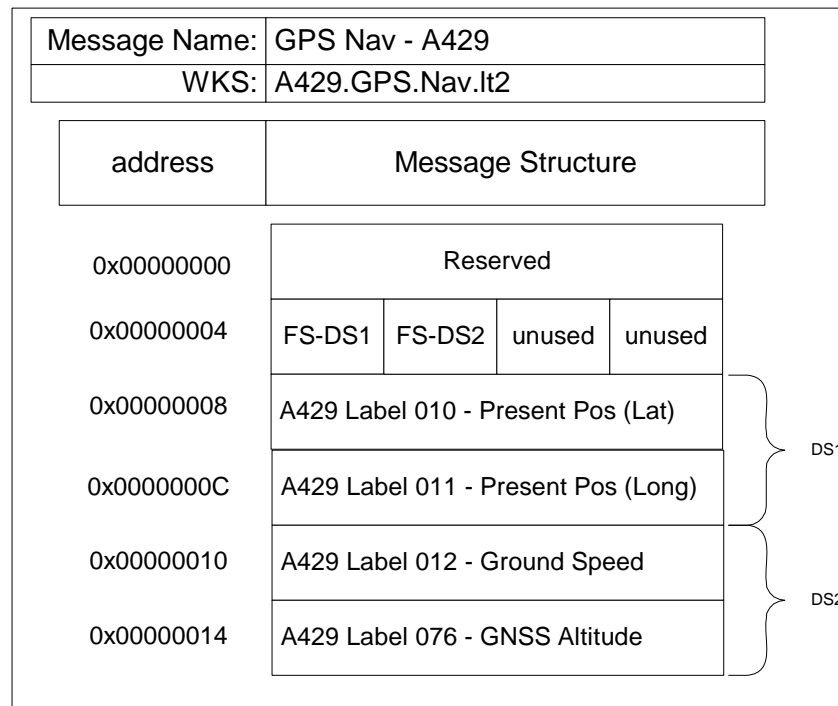


Figure B-1.3 – Message Format for Multiple Label using Multiple DS

Approach 2:

This approach assumes that a gateway is running on some sort of schedule and periodically sending messages with ARINC 429 labels in them. Labels may arrive at the gateway at different rates. This can cause there to be varying number of labels to concentrate onto AFDX for any given cycle. This approach presents the labels to a data consumer as though the consumer was directly connected to the ARINC 429 device. The labels are presented in the order received by the gateway so the gateway and AFDX network are essentially transparent. Since any one message the gateway sends could have a varying number of labels, a variable length data structure is required (see Figure B-1.4). The variable length opaque structure is provided to do just that. The system designer must figure out what the highest or maximum number of labels is that could possibly arrive at the gateway between transmissions. From this maximum number the maximum size of the variable length opaque structure is established. This structure is then statically configured into the system.

As labels arrive at the gateway, they are placed one after another in the opaque structure beginning with the byte after the length field and pad field. When it is time for transmission, the gateway will compute how many bytes of the static length have good data in them and load this number into the length field of the structure. The unused portion of the opaque field is filled with binary zeros. The message is then sent, with the entire static structure, not just the used portion.

The data consumer that receives this structure can read the length field and compute how many labels are there to process. In the case where legacy code is being reused to service the ARINC 429 devices, the code most likely already exists to parse through each label and determine what the data is.

APPENDIX B
GUIDELINES FOR ARINC 429 TO AFDX FORMATTING

AFDX (Wrapped A429)

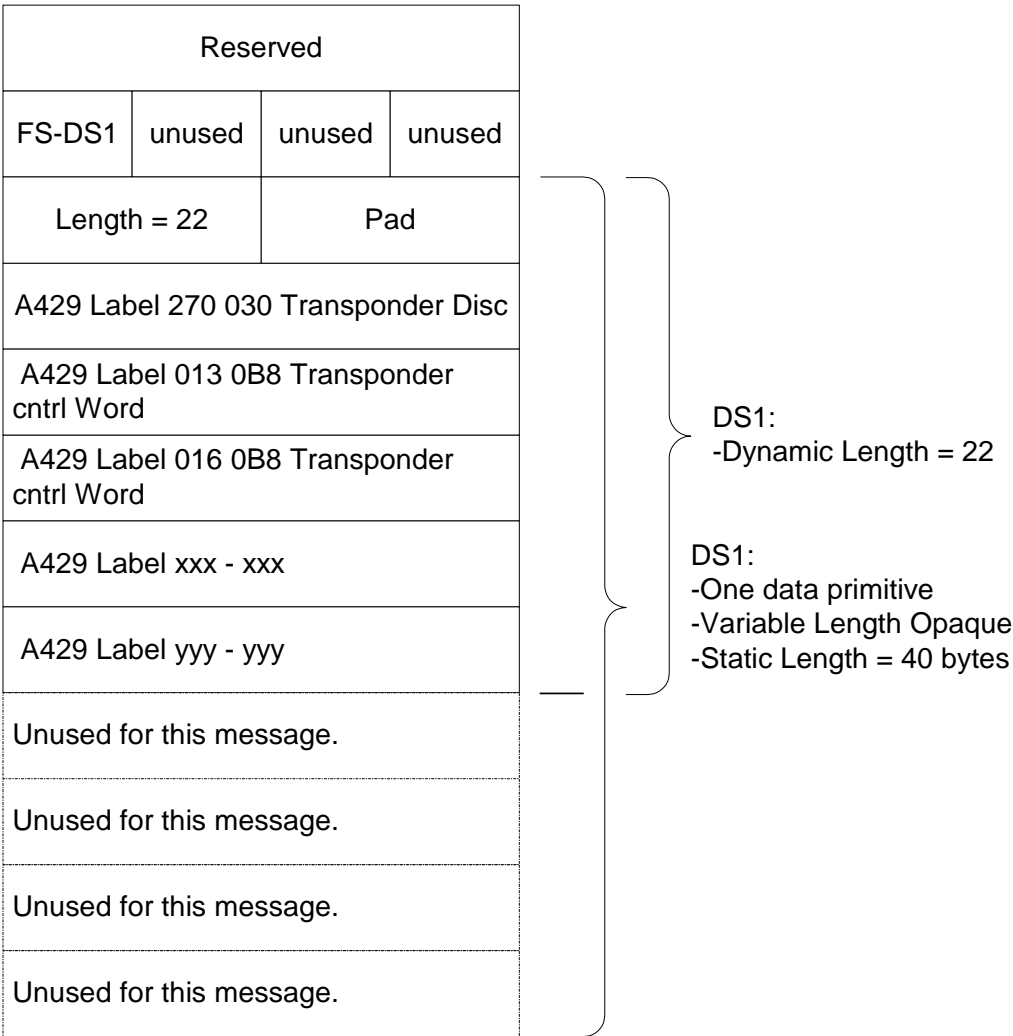


Figure B-1.4 – Message Format Using Variable Length Opaque Structure for ARINC 429

Sending the complete, statically defined structure, helps keep the jitter in the system lower. This makes every message the same length, even though the data in the message is not all valid. The bandwidth has been allocated to this data flow for the maximum size the message can be, and is not useable by any other flow. It is therefore not wasted.

APPENDIX C NETWORK TERMINOLOGY

Application Data

Application Data is the unit of data passed across the interface between an application and the Communication Ports or SAP ports implemented in the End-System. Application data consist of only the data sent or received by this application. Application data are transmitted by encapsulation inside an UDP datagram or a TCP segment.

The term “Application Message” may also be used sometimes in place of “Application Data.”

Segment

A segment is the unit of data passed across the interface between the TCP layer and the IP layer. A segment consists of a TCP header followed by an application message. A segment is transmitted by encapsulation inside an IP datagram.

Message

In this document, a message is the unit of transmission at TCP or UDP level. A message consists of a transport protocol header followed by application data. To be transmitted end-to-end through the AFDX, a message must be encapsulated inside an IP datagram.

IP Datagram

An IP datagram is the unit of end-to-end transmission at IP level. An IP datagram consists of an IP header followed by transport layer data, i.e., of an IP header followed by an UDP datagram or a segment.

In this document, the unqualified term "datagram" should be understood to refer to an IP datagram.

Fragment

Maximum size of a packet is limited by the MTU (Max Transfer Unit) of the network. Whenever an UDP datagram is too long to fit in a packet, it is break in an almost arbitrary number of pieces that are placed in fragments.

Packet

A packet is the unit of data passed across the interface between the IP layer and the data link layer (MAC). It includes an IP header and data. A packet is either a complete IP datagram or a fragment.

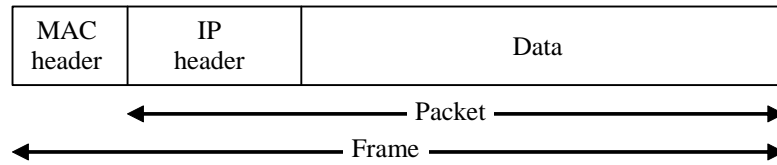
Frame

A frame is the unit of transmission at Ethernet level, and consists of a MAC header followed by a packet.

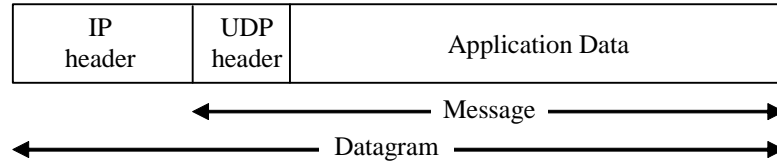
The terms frame, packet, datagram, message, segment and Application data are illustrated by the following schematic diagrams:

APPENDIX C NETWORK TERMINOLOGY

Transmission on connected network:



Before IP fragmentation or after IP reassembly (for UDP and TCP):



**APPENDIX D
SERVICES TO PROTOCOL MAPPING**

Services to Protocol Mapping

Table D-1 presents the mapping of network services to the network protocols. This provides insight into what protocols are used to provide an given service. The rows of the table list the services and the columns represent the protocols. When an "X" appears at the junction point of a row and column, this indicates that the protocol listed in that column is used in providing the network service listed for the row, in an AFDX network.

Services	Protocols																		
		VIP	IEEE-802.3	IPv4	UDP	TCP	ARP	RARP	ICMP	IGMP	FTP	DNS	DL-TFTP	TFTP	BOOTP	DHCP	HTTP	Telnet	SNMP
	Physical & Data Link Layer		X																
	Broadcast																		
	Multicast		X	X															
	Network Layer			X					X										
	Data Transport (simple)				X														
	Data Transport (sophisticated)																		
	Map IP Address to 802.3 MAC		X																
	Map Host Name to IP Address																		
	Network Diagnostics																		X
	Network Configuration		X	X	X				X				X						X
	Routing																		
	File Transfer													X					
	Remote User Access																		
	Network Management																		X
	Simple Remote Data Access																		
	ARINC 615A Data Loader		X	X	X								X						
		Note: Many of the protocols listed in this table are profiled for use in AFDX. Read the body of ARINC 664 part 7 for details.																	

Table D-1 – Services to Protocol Mapping

AERONAUTICAL RADIO, INC.
2551 Riva Road
Annapolis, Maryland 24101-7435

SUPPLEMENT 1
TO
ARINC SPECIFICATION 664P7
AIRCRAFT DATA NETWORK, PART 7, AVIONICS FULL-DUPLEX SWITCHED ETHERNET
NETWORK

Published: September 22, 2009

Prepared by the AEEC

A. PURPOSE OF THIS DOCUMENT

This Supplement makes some editorial corrections and adds descriptive paragraphs some features.

B. ORGANIZATION OF THIS SUPPLEMENT

In the past, changes introduced by a Supplement to an ARINC Standard were identified by vertical change bars with an annotation indicating the change number. Electronic publication of ARINC Standards has made this mechanism impractical.

In this document **blue bold** text is used to indicate those areas of text changed by the current Supplement only.

C. CHANGES TO ARINC SPECIFICATION 664P7 INTRODUCED BY THIS SUPPLEMENT

This section presents a complete listing of the changes to the document introduced by this Supplement. Each change is identified by the section number and the title as it will appear in the complete document. Where necessary, a brief description of the change is included.

3.1.1 ES Identification

The words “available only to” are replaced by “available to” was deleted to clarify the meaning.

3.2.4.1 Latency

The word ‘frame delay’ has been changed to ‘contention delay’ to be more accurate.

3.2.6.2.2 Redundancy Management

The second paragraph was deleted. This section has been modified to add more detail to the redundancy management process.

3.3 Interoperability at the IP Layer and Above

Editorial correction.

3.3.3.2 ES IP Profile

This section has been restructured into two sub-paragraphs.

3.3.3.2.1 IP Packet Structure

This section heading was added to existing text.

3.3.3.2.2 IP Fragmentation/Reassembly

This new section was added to better describe this process.

3.4.1.4 IP Addressing Format

This section was rewritten to conform to current practice as far as using IP addresses which go beyond the AFDX network.

3.4.1.4.1 IP Source Address

A clarifying commentary was added.

3.4.1.5.1 AFDX Communication Ports

Editorial correction.

4.1.1.2 Frame Filtering

Editorial correction, bullets were added to separate three 'frame' items.

ARINC Standard – Errata Report

1. Document Title

ARINC Specification 664P7: *Aircraft Data Network, Part 7, Avionics Full-Duplex Switched Ethernet Network*

Published: September 15, 2009

2. Reference

Page Number: _____ Section Number: _____ Date of Submission: _____

3. Error

(Reproduce the material in error, as it appears in the standard.)

4. Recommended Correction

(Reproduce the correction as it would appear in the corrected version of the material.)

5. Reason for Correction (Optional)

(State why the correction is necessary.)

6. Submitter (Optional)

(Name, organization, contact information, e.g., phone, email address.)

Please return comments to fax +1 410-266-2047 or standards@arinc.com

Note: Items 2-5 may be repeated for additional errata. All recommendations will be evaluated by the staff. Any substantive changes will require submission to the relevant subcommittee for incorporation into a subsequent Supplement.

[To be completed by IA Staff]

Errata Report Identifier: _____ **Engineer Assigned:** _____

Review Status: _____

ARINC IA Project Initiation/Modification (APIM)

- 1.0 Name of Proposed Project** **APIM #:** _____
(Insert name of proposed project.)
- 2.0 Subcommittee Assignment and Project Support**
- 2.1 Identify AEEC Group
(Identify an existing or new AEEC group.)
- 2.2 Support for the activity
Airlines: *(Identify each company by name.)*
Airframe Manufacturers:
Suppliers:
Others:
- 2.3 Commitment for resources *(Identify each company by name.)*
Airlines:
Airframe Manufacturers:
Suppliers:
Others:
- 2.4 Chairman: *(Recommended name of Chairman.)*
- 2.5 Recommended Coordination with other groups
(List other AEEC subcommittees or other groups.)
- 3.0 Project Scope** *(why and when standard is needed)*
- 3.1 Description
(Insert description of the scope of the project. Use the following symbol to check yes or no below. ☒)
- 3.2 Planned usage of the envisioned specification
- New aircraft developments planned to use this specification yes ☐ no ☐
- Airbus: *(aircraft & date)*
- Boeing: *(aircraft & date)*
- Other: *(manufacturer, aircraft & date)*
- Modification/retrofit requirement yes ☐ no ☐
- Specify: *(aircraft & date)*
- Needed for airframe manufacturer or airline project yes ☐ no ☐
- Specify: *(aircraft & date)*

Mandate/regulatory requirement yes ☐ no ☐

Program and date: *(program & date)*

Is the activity defining/changing an infrastructure standard? yes ☐ no ☐

Specify *(e.g., ARINC 429)*

When is the ARINC standard required?
(month/year)

What is driving this date? *(state reason)*

Are 18 months (min) available for standardization work? yes ☐ no ☐

If NO please specify solution: _____

Are Patent(s) involved? yes ☐

If YES please describe, identify patent holder: _____

3.3 Issues to be worked

(Describe the major issues to be addressed.)

4.0 Benefits

4.1 Basic benefits

Operational enhancements yes ☐ no ☐

For equipment standards:

a. Is this a hardware characteristic? yes ☐ no ☐

b. Is this a softwareware characteristic? yes ☐ no ☐

c. Interchangeable interface definition? yes ☐ no ☐

d. Interchangeable function definition? yes ☐ no ☐

If not fully interchangeable, please explain: _____

Is this a software interface and protocol standard? yes ☐ no ☐

Specify: _____

Product offered by more than one supplier yes ☐ no ☐

Identify: *(company name)*

4.2 Specific project benefits

(Describe overall project benefits.)

4.2.1 Benefits for Airlines

(Describe any benefits unique to the airline point of view.)

4.2.2 Benefits for Airframe Manufacturers

(Describe any benefits unique to the airframe manufacturer's point of view.)

4.2.3 Benefits for Avionics Equipment Suppliers

(Describe any benefit unique to the equipment supplier's point of view.)

5.0 Documents to be Produced and Date of Expected Result

5.1 Meetings and Expected Document Completion

The following table identifies the number of meetings and proposed meeting days needed to produce the documents described above.

Activity	Mtgs	Mtg-Days (Total)	Expected Start Date	Expected Completion Date
<i>Document a</i>	# of mtgs	# of mtg days	mm/yyyy	mm/yyyy
	# of mtgs *	# of mtg days *		
<i>Document b</i>	# of mtgs	# of mtg days	mm/yyyy	mm/yyyy
	# of mtgs *	# of mtg days *		

* Indicate unsupported meetings and meeting days, i.e., technical working group or other ad hoc meetings that do not requiring IA staff support.

6.0 Comments

(Insert any other information deemed useful to the committee for managing this work.)

For IA Staff use

Date Received: _____

IA Staff Assigned: _____

Estimated Cost: _____

Potential impact: _____

(**A. Safety** **B. Regulatory** **C. New aircraft/system** **D. Other**)

Forward to committee(s) (AEEC, AMC, FSEMC): _____ Date Forwarded: _____

Committee resolution: _____

(**0 Withdrawn** **1 Authorized** **2 Deferred** **3 More detail needed** **4 Rejected**)

Assigned Priority: _____ Date of Resolution: _____

(**A High - execute first** **B Normal - may be deferred.**)

Assigned to SC/WG: _____

