EUROPEAN ORGANISATION FOR THE SAFETY OF AIR NAVIGATION

EUROCONTROL

EUROCONTROL GUIDELINES FOR IMPLEMENTATION SUPPORT (EGIS)
Part 5
COMMUNICATION & NAVIGATION SPECIFICATIONS
CHAPTER 13
FLIGHT MESSAGE TRANSFER PROTOCOL (FMTP)

EGIS.COM.FMTP

Edition	**:**	**2.0**
Edition Date	**:**	**12/12/08**
Status	**:**	**Released Issue**
Class	**:**	**General Public**

EUROPEAN AIR TRAFFIC CONTROL HARMONISATION AND INTEGRATION PROGRAMME

DOCUMENT CHARACTERISTICS

TITLE		
EUROCONTROL GUIDELINES FOR IMPLEMENTATION SUPPORT (EGIS) **Part 5** **COMMUNICATION & NAVIGATION SPECIFICATIONS** **Chapter 13** **FLIGHT MESSAGE TRANSFER PROTOCOL (FMTP)**		
	EATMP Infocentre Reference:	05/06/10-1
Document Identifier	**Edition Number:**	2.0
	Edition Date:	12/12/08

Abstract

Flight Message Transfer Protocol is a communications profile designed to support the exchange of data between flight data or flight plan processing systems. It specifies a Message Transfer Protocol that operates over TCP/IP and includes a system identification phase.

The Flight Message Transfer Protocol (FMTP) is published as a European Union Implementing Rule No. 633/2008. The Implementing Rule is further supported by EUROCONTROL Specifications for FMTP recognised as Community Specifications and means of compliance to the Implementing Rule.

This EUROCONTROL Guideline further supports implementation of the EUROCONTROL Specifications for FMTP.

Keywords

Flight data	Interface Control	IP	FMTP
Data exchange	OLDI	FDPS	Protocol
Networks	TCP	Implementing Rule	SES

Contact Person(s)	Tel	Unit
E. CERASI	+32 2 7293791	DAP/CSP

STATUS, AUDIENCE AND ACCESSIBILITY					
Status		**Intended for**		**Accessible via**	
Working Draft	☐	General Public	☑	Intranet	☐
Draft	☐	EATMP Stakeholders	☐	Extranet	☐
Proposed Issue	☐	Restricted Audience	☐	Internet (www.eurocontrol.int)	☐
Released Issue	☑	*Printed & electronic copies of the document can be obtained from the EATMP Infocentre (see page iii)*			

ELECTRONIC SOURCE		
Path:		
Host System	Software	Size
Windows_NT	Microsoft Word 8.0b	198 Kb

DOCUMENT APPROVAL

The following table identifies all management authorities who have successively approved the present issue of this document.

AUTHORITY	NAME AND SIGNATURE	DATE
WP LEADER	E. CERASI	19/12/08
HEAD OF COM DOMAIN	J. POUZET	19/12/08.
HEAD OF DAP/CSP	R. STEWART	19/12/08
SIS/EIS/EGIS REVIEW AND DOC CONTROL	P. KORFIATIS	19/12/08
SIS/ASSIST MANAGER	T. PAPAVRAMIDIS	19/12/08
HEAD OF DAP/SIS	D. APSOURIS	7/12/08
DIRECTOR EATM	B. REDEBORN	16/1/09

DOCUMENT CHANGE RECORD

The following table records the complete history of the successive editions of the present document.

EDITION	DATE	REASON FOR CHANGE	SECTIONS / PAGES AFFECTED
1.0	22 April 2005	First draft edition	All
2.0	12 Dec. 2008	Second edition following the release of publication of the FMTP Implementing Rule and EUROCONTROL Specifications	All

© 2008 European Organisation for the Safety of Air Navigation (EUROCONTROL). All rights reserved.

The copyright and all other rights relating to the data/information contained herein are vested in EUROCONTROL and no part may be reproduced or used except as authorised by contract or other written permission. The copyright and the foregoing restriction and use extend to all media in which the data/information is embodied.

The present EUROCONTROL Guidelines for Implementation Support (EGIS) are made available free of charge for the sole purposes of review by industry or reference for Air Navigation Service Providers (ANSPs) to prepare Calls For Tender or associated development.

DISCLAIMER

THE DATA/INFORMATION CONTAINED IN THE EGIS IS PROVIDED ON AN AS-SEEN/AS IS-BASIS AND EUROCONTROL PROVIDES AND ACCEPTS NO EXPRESS OR IMPLIED WARRANTY OF ANY KIND, INCLUDING BUT NOT LIMITED TO THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ACCEPTS NO LIABILITY WHATSOEVER FOR OR IN CONNECTION WITH THE USE OF THE DATA/INFORMATION CONTAINED IN THE EGIS.

TABLE OF CONTENTS

FIGURES

Intentionally blank

FOREWORD

The Single European Sky (SES) Interoperability Implementing Rule, Flight Message Transfer Protocol, has been published 7 June 2007 as European Union Commission Regulation No.633/2007 [Reference 1].

The EUROCONTROL Specification of Interoperability and Performance Requirements for the Flight Message Transfer Protocol [Reference 2] have been published in June 2007. The Specifications are based on the previous Edition 1.0 of this Guideline [Reference 7]. Edition 2.0 of this Guideline has been prepared to take into account the fore-mentioned publications.

The advent of the internet and its associated suite of protocols have radically changed the way in which information is exchanged. In view of the decline of X.25 communications, the Flight Message Transfer Protocol has been designed on the basis of the internet protocols thereby allowing applications such as On-Line Data Interchange (OLDI) [Reference 3], to take full advantage of modern communications.

The replacement of X.25 by the internet protocol (IP) for Flight Data Exchange is not straight-forward. As a result, FMTP is not backwards compatible with the former EUROCONTROL Standard Flight Data Exchange Interface Control Document (FDE ICD) Part 1 [Reference 5]. A gateway function providing seem-less communication between FMTP and the FDE ICD Part 1 is technically feasible; or, existing FDE ICD Part 1 implementations may wish to integrate FMTP in a dual-stack fashion. It is assumed that the migration from FDE ICD Part 1 to FMTP modifies the communication layers and does not affect operational applications and procedures as defined by OLDI.

The intention of this edition of the EUROCONTROL Guideline is to remove technical provisions that have been integrated within the Implementing Rule and the EUROCONTROL Specifications.

Intentionally blank

1. GENERAL CONSIDERATIONS

1.1 Scope of the this document

1.1.1 The Flight Message Transfer Protocol (FMTP) provides a data communications interface for the exchange of flight-related data messages between Air Traffic Control Units (ATCUs) for the purpose of notification, co-ordination and transfer (COTR).

1.1.2 This EUROCONTROL Guideline supports the implementation of the EUROCONTROL FMTP specification [Reference 2].

1.1.3 The EUROCONTROL Specification for FMTP and this Guideline are applicable for the support of EUROCONTROL Specification On-Line Data Interchange (OLDI) as described in [Reference 3].

1.1.4 FMTP is applicable for connection using either:

- Internet Protocol (IP) over point-to-point connections,
- Internet Protocol (IP) over Public Switched Telephone Network (PSTN), Integrated Services Digital Network (ISDN) connections,
- Internet Protocol (IP) over public data networks that provide standard internet protocol (IP) access, or
- Internet Protocol (IP) over private data networks that provide standard internet protocol (IP) access.

NOTES

1. *As a general rule, OLDI implementation connections make use of private networks for increased reliability, security and managed services.*
2. *The arrangement between Flight Data Processing Systems (FDPSs) is represented in Figure 1, in which the operational application is also referred to as the MT-User (User of the Message Transfer Protocol).*
3. *Figure 1 does not illustrate potential backup connections or all possible interface arrangements.*

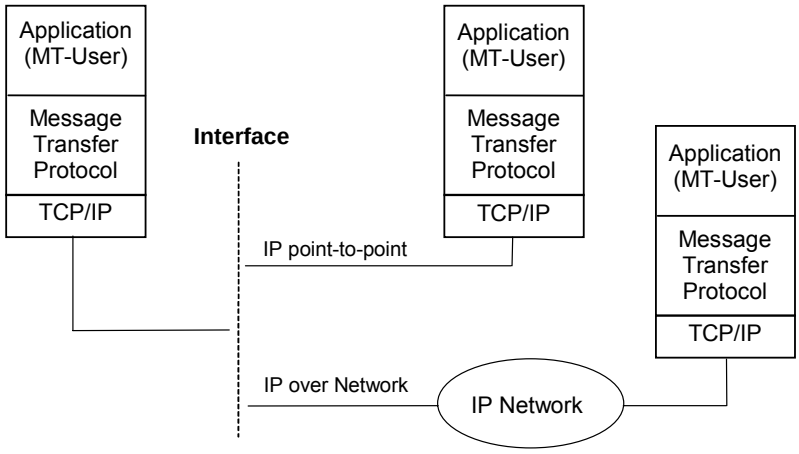

Figure 1 - Possible Interface Arrangements (non-exhaustive)

1.1.5 This Guideline also considers the re-use of the FMTP protocol for other applications than OLDI.

1.2 Use of the Document

This document can be used for a different purposes, including the following:

a) As guidelines supporting the understanding of the EUROCONTROL FMTP Specification;

b) As a complementary document source to define further requirements for operational system implementation;

1.3 Organisation of the Document

1.3.1 This document is composed of the following sections.

Section 1: provides an introduction including information about the document structure, conformance with the ECAC Strategy and reference documents.

Section 2: provides a short technical overview of the FMTP protocol stack, relation with previous versions of specifications, implementation services and arrangements.

Section 3: provides general guidelines on the FMTP protocol.

Section 4: provides detailed guidelines on the FMTP protocol.

Section 5: indicates possible enhancements when using FMTP for other applications than OLDI

Annex A provides information about the system management

Annex B provides information about the service availability and reliability

Annex C provides information about the conformance testing methodology

1.4 Definitions

1.4.1 For the purpose of this EUROCONTROL Guideline, the following definitions shall apply:

1.4.2 **Implementation or System:** a conforming implementation to the FMTP Implementing Rule as defined by EC Regulation No.633/2007 and the EUROCONTROL FMTP Specification [Reference 2], which is uniquely defined by its IP address, TCP port and it's associated identification value.

1.4.3 **FMTP Connection:** the relationship between two conforming implementations that have reached the READY state as described in the EUROCONTROL FMTP Specification [Reference 2], by completing the identification procedure over an active TCP transport connection.

1.4.4 **FMTP Association:** the relationship between two conforming implementations that have reached the DATA_READY state as described in EUROCONTROL FMTP Specification [Reference 2], by completing the exchange of STARTUP messages over an FMTP connection.

1.5 Abbreviation - Symbols used

ASCII	American Standard Code for Information Interchange
ATC	Air Traffic Control
ATCU	Air Traffic Control Unit
ATM	Air Traffic Management
CHAP	Challenge-Handshake Authentication Protocol
COTR	Co-ordination and Transfer
EATM	European Air Traffic Management
ECIP	European Convergence and Implementation Plan
FDE	Flight Data Exchange
FDPS	Flight Data Processing System
FMTP	Flight Message Transfer Protocol
IANA	Internet Assigned Numbers Authority
ICAO	International Civil Aviation Organization
ICMP	Internet Control Message Protocol
ICD	Interface Control Document
IEC	International Electrotechnical Commission
IP	Internet Protocol
ISO	International Organization for Standardization
ITU-T	International Telecommunication Union - Telecommunication Standardization Sector
ISDN	Integrated Services Digital Network
MIB	Management Information Base
MSB	Most Significant Bit
MT	Message Transfer
MTP	Message Transfer Protocol
OLDI	On-Line Data Interchange
OSI	Open Systems Interconnection
PICS	Protocol Implementation Conformance Statement
PSTN	Public Switched Telephone Network
RFC	Request for Comment
SES	Single European Sky
SNMP	Simple Network Management Protocol
SSL	Secure Socket Layer
SUT	System Under Test

T<x>	Timer (where <x> is a single letter for referencing)
TCP	Transmission Control Protocol
TLS	Transport Layer Security

1.6 History

The ECAC strategy for the 1990s, containing the overall objective "to provide increasing airspace and control capacity urgently while maintaining a high level of safety", was adopted by the ECAC Transport Ministers in 1990. In addition to the overall objective, the ECAC Strategy specified five implementation objectives, addressing radar, communications, airspace management, common standards and specifications, and human factors. To achieve these objectives, the European Air Traffic Control Harmonisation and Implementation Programme (EATCHIP) was created.

A second ECAC Strategy, addressing capacity at airports and in terminal areas, was adopted by the ECAC Transport Ministers in 1992, and resulted in the creation of the Airport and Air Traffic Services Interface (APATSI) programme. A subsequent decision by the ECAC Transport Ministers has resulted in the absorption of this programme into EATCHIP, in a philosophy known as "gate-to-gate".

The first phase of EATCHIP was one of appraising the current situation in order to obtain, for the first time, a complete picture of the European ATC services, systems and procedures. This was followed by a programme development phase in which the deficiencies and problems identified in the first phase were addressed.

As a result of this second phase, two complementary programmes were established; the EATCHIP Work Programme (EWP) and the Convergence And Implementation Programme (CIP). The aim of these programmes is the accomplishment of the third phase of EATCHIP, to operationally integrate the European ATM system.

The basis for the harmonisation and integration process is the Convergence and Implementation Programme Document (CIPD) which provides a reference and a framework for national and multi-national plans. The CIPD contains CIP Objectives for which functional performance levels are defined, the applicability of which is dependent upon the subject airspace complexity.

Complementary to the CIP, as part of the EATCHIP Work Programme, operational requirements, CNS/ATM architecture, and technical specifications are being defined as a means of realisation of the CIP Objectives.

1.7 Link to the ATM 2000+ Strategy

In order to cope with the increase level of traffic and to bring further substantial gains in ATM capacity and efficiency to meet this predicted future demand, a uniform European ATM Strategy for the year 2000+ has been developed. This strategy is built on the previous work and results of the EATMS, EATCHIP and APATSI.

This ATM2000+ Strategy has led to the development of the EATMP Programme, which is described in the EATMP Work Programme (EWP) and to the revision of the CIP Process. New requirements emerging from EATMP after having

successfully passed the validation process and judged sufficiently matured were introduced in this document.

An EATMP Communication Strategy Document has been released on 13 Jan. 2003 as version 4.2. This document is an updated version of the original EATCHIP Communication Strategy released in 1998. This EUROCONTROL Guideline is fully in line with this strategy.

1.8 Link to the ECIP

The European Convergence and Implementation Plan (ECIP), which is an evolution of the Convergence and Implementation Programme, is a key element in the overall performance planning process for improving European Air Traffic Management (ATM). It draws together, and provides the framework for, the agreed common actions to be taken by the EUROCONTROL States and other European Civil Aviation Conference (ECAC) States participating in the European Air Traffic Management (EATM) to apply the high-level objectives, principles and Operational Improvements contained in the EUROCONTROL ATM 2000+ Strategy.

The ECIP contains more than 60 implementation objectives, each of which supports an operational, technical or procedural change to the European ATM system [Reference 2]. The objective associated to this EUROCONTROL guideline is the ITY-FMTP titled: “Apply a common flight message transfer protocol (FMTP)”. The objective aims at facilitating the monitoring and reporting of the implementation of a common flight message transfer protocol in European ATM in line with the EC regulations and through the SES implementation monitoring and reporting mechanism.

ITY-FMTP consists of a number of Lines of Actions (LoAs) as follows:

ITY-FMTP-ASP01 Implement the flight message transfer protocol (FMTP)

ITY-FMTP-REG01 Ensure that the flight message transfer protocol (FMTP) is used between ATS units and controlling military units

ITY-FMTP-REG02 Ensure compliance with regulation

It is to be noted that ITY-FMTP supersedes former ECIP Objective COM-04 which lead to the publication of the first edition of this Guideline.

1.9 Link to the Single European Sky

At the end of 2003 an agreement was reached between the European Parliament and the Council. It was an important step towards implementing the Single European Sky following two key principles:

- Establish a decision-making and regulatory framework, which will improve air safety standards and, at the same time , remedy the structural problems afflicting air traffic control (within this framework, the European Organization for the Safety of Air Navigation, EUROCONTROL, will provide technical support to the commission).
- Mobilise all concerned in a comprehensive reform of air traffic management by reorganizing the provision and supervision of air traffic control services and speeding up the development and introduction of new technologies. The organisational, operational, economic, financial, social and technical aspects of SES will be covered by a wide-ranging action programme designed to ensure that airspace is organised and used in a way meeting the needs of

civil and military air traffic.

In 2004-2005 the commission, with the collaboration of EUROCONTROL, turn to the task of drafting the detailed rules needed in order to put this package of legislation into action.

On some issues, EUROCONTROL is mandated to prepare the implementing rules. The single sky committee is an emanation of the Council of the European Union whereby Member States representatives collaborate with Commission in the rule making process

The drafting of an associated Single European Sky interoperability Implementing Rule on FMTP has been mandated to EUROCONTROL. This lead to the specification of FMTP Implementing Rule published as EC Regulation No.633/2007 and the EUROCONTROL Specification FMTP [Reference 2] endorsed as a Community Specification. This EUROCONTROL Guideline includes complementary provisions that will assist implementation by all stakeholders and partners involved.

1.10 Reference Documents

The following documents and standards contain provisions which, through reference in this text, constitute provisions of this EUROCONTROL Guideline.

At the time of publication of this EUROCONTROL Guideline, the editions indicated for the referenced documents and standards were valid.

Revisions of the referenced documents shall not form part of the provisions of this EUROCONTROL Guideline until they are formally reviewed and incorporated into this EUROCONTROL Guideline.

Any conflict between this EUROCONTROL Guideline and reference documents should be immediately reported to EUROCONTROL. In case of conflict between this EUROCONTROL Guideline and the contents of the EC regulation No 633/2007 [Reference 1] or the EUROCONTROL FMTP Specification [Reference 2] those referenced documents shall take precedence.

1. European Union Commission Regulation (EC) No 633/2007 of 7 June 2007 (laying down requirements for the application of a flight message transfer protocol used for the purpose of notification, coordination and transfer of flights between air traffic control units)

2. EUROCONTROL Specification of Interoperability and Performance Requirements for the Flight Message Transfer Protocol (FMTP), EUROCONTROL-SPEC-0100, Edition 2.0, 2007

3. EUROCONTROL Specification for On-Line Data Interchange (OLDI), EUROCONTROL-SPEC-0106, Edition 4.1, 2008

4. EUROCONTROL European Convergence and Implementation Plan for the years 2009-2013, Detailed objective descriptions, EATM Reference 08/05/20-20, September 2008

5. EUROCONTROL Standard Document Flight Data Exchange - Interface Control Document Part 1 : Point-to-Point and Limited Networking Circuits,

COM.ET1.ST12-STD.01-01, Edition 1.0, 1998.

6. EUROCONTROL FMTP Interoperability Test Plan, EATM Reference 05/06/07-3, Edition 1.0, 6 June 2005
7. EUROCONTROL Guidelines for Implementation Support (EGIS) Part 5, Communication & Navigation Specifications, Chapter 13, Flight Message Transfer Protocol (FMTP), Edition 1.0, 2005
8. Internet Engineer Task Force (IETF) STD0007, RFC0793:1981, Transmission Control Protocol.
9. Internet Engineer Task Force (IETF) BCP0018, RFC2277:1998, IETF Policy on Character Sets and Languages.
10. Internet Engineer Task Force (IETF) RFC4301:2005, Security Architecture for the Internet Protocol (IPsec).
11. Internet Engineer Task Force (IETF) RFC4443:2006, Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification.
12. Internet Engineer Task Force (IETF) RFC2460:1998, Internet Protocol, Version 6 (IPv6) Specification.
13. Internet Engineer Task Force (IETF) BCP0028, RFC2488:1999, Enhancing TCP over Satellite Channels Using Standard Mechanisms.
14. Internet Engineer Task Force (IETF) RFC5246:2008, The Transport Layer Security (TLS) Protocol Version 1.2.
15. Internet Engineer Task Force (IETF) RFC2460:1998, Internet Protocol, Version 6 (IPv6) Specification.
16. ISO/IEC 7498-1:1994 (2nd edition), Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model.
17. ITU-T Recommendation X.25 (1996), Interface between Data Terminal Equipment (DTE) and Data Circuit-terminating Equipment (DCE) for terminals operating in the packet mode and connected to public data networks by dedicated circuit.
18. ICAO Convention, Annex 10, Volume III, Part I, Chapter 3 – Aeronautical Telecommunications Network, July 2008.
19. ICAO Document 9896 ATN/IPS Manual (Pending Publication)

1.11 Notations

1.11.1 For the purpose of this EUROCONTROL Guideline, binary values or a sequence of bits are denoted in hexadecimal using the notation 'd'H, where the letter d stands for a digit or a sequence of hexadecimal digits.

Intentionally blank

2. TECHNICAL OVERVIEW

2.1 Protocol Stack

2.1.1 The FMTP protocol stack is illustrated in Figure 2. The figure places the protocols in the framework of the Open Systems Interconnection (OSI) Basic Reference Model [Reference 16], by aligning the stack with the corresponding OSI model layers. However, this protocol stack does not implement OSI network, transport or upper layer protocols.

2.1.2 The Message Transfer Protocol (MTP) makes use of a specific header for data exchanges over TCP which is not illustrated in Figure 2.

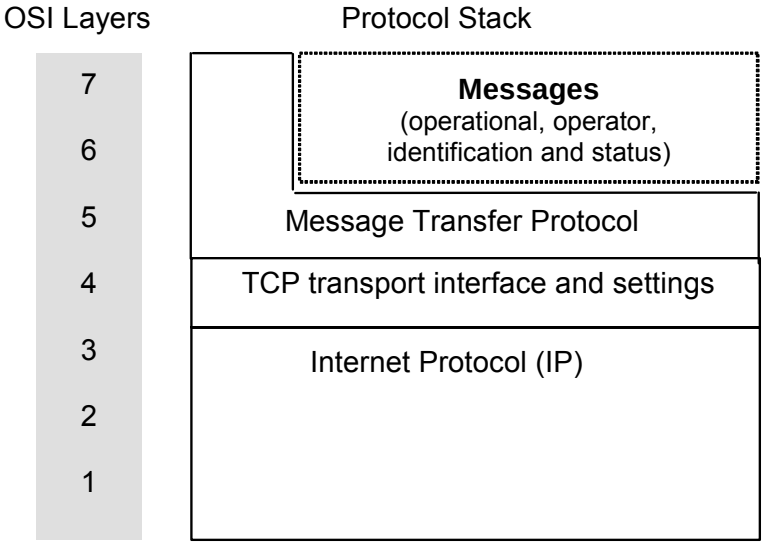

Figure 2 - FMTP Protocol Stack

2.2 Relation to the former EUROCONTROL FDE ICD Part 1 Standard

2.2.1 FMTP is based on the former EUROCONTROL Standard for Flight Data Exchange Interface Control Document - Part 1 [Reference 5].

2.2.2 The Message Transfer Protocol has modified to interface to the new transport layer and to handle the new system identification phase.

2.2.3 The Message Transfer Protocol makes use of TCP/IP services instead of X.25 which modifies the handling of lower layer connection establishment and termination.

2.2.4 TCP is a client server protocol, leading to revisions of connection establishment.

2.2.5 The Message Transfer Protocol describes the creation and management of FMTP connections and FMTP associations.

2.2.6 The Message Transfer Protocol services and descriptions have been reviewed and improved.

2.2.7 The potential co-hosting of FMTP implementations is explained in this Guideline.

2.2.8 The FMTP messages and message headers extend those of the former FDE ICD Part 1.

2.2.9 The Message Transfer Protocol includes a new system identification phase, following TCP connection establishment, to identify the peers of the FMTP connection that is being created; this replaces the former use of X.25 NSAP addresses with embedded ATC Unit Identifiers and Selectors.

2.2.10 The former ATC Unit Identification schema is abandoned.

2.2.11 The state transitions for Tr timer timeout have been reviewed and modified.

2.3 Relation to SES FMTP Implementing Rule

2.3.1 The Single European Sky Committee has approved the publication of the interoperability Implementing Rule on FMTP No.633/2008 [Reference 1] on the basis of material prepared by EUROCONTROL.

2.3.2 EUROCONTROL Specifications on FMTP [Reference 2] have been recognised as Community Specifications meeting the interoperability and performance requirements for the flight message transfer protocol (FMTP).

3. GENERAL GUIDELINES

3.1 Service Interfaces

3.1.1 The MT-User is an operational application such as OLDI [Reference 3], or some intermediate resource that is bound to such an operational application, that interfaces to the FMTP protocol stack.

NOTE - *EUROCONTROL Guideline is drafted on the basis of an MT-User interfacing to the FMTP Message Transfer Protocol as illustrated in Figure 3.*

3.1.2 The FMTP Specification requires conforming implementations to implement the connection establishment, association establishment, data transfer and release services.

NOTES

1. *The EUROCONTROL FMTP Specification makes use of abbreviated terms MT-CON, MT-ASSOC and MT-DIS to represent the service interface as described in section A.3 of the previous edition of this Guideline (i.e. MT-Connect, MT-Disconnect, MT-Associate. MT-STOP corresponds to former MT-Stop and MT-DATA corresponds to former MT-Data.*
2. *There is a direct correspondence between then events and actions described in section 5 of the FMTP Specification and section A.3 of the previous version of this guideline.*
3. *The naming convention for the services is a local matter that does affect interoperability.*

3.1.3 The Message Transfer Protocol interacts with the TCP transport layer through the T-Connect and T-Disconnect service primitives. This interaction may be a direct mapping or an indirect mapping through an intermediate programming interface (API) such as X/Open XTI or a socket call.

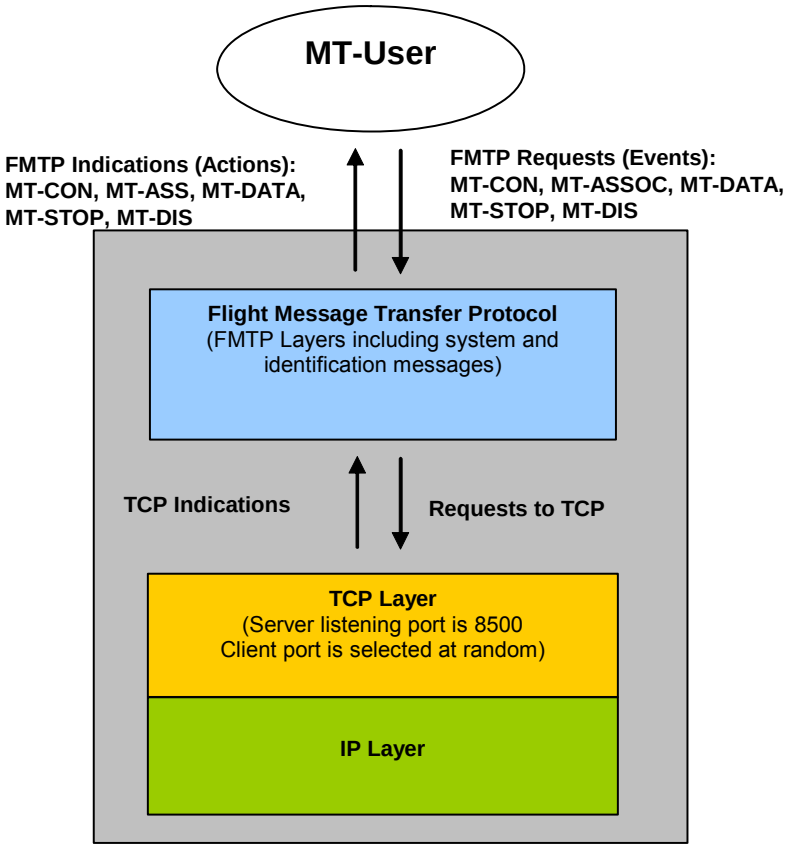

Figure 3 - Service Interfaces

3.2 Implementation Arrangements and Addressing

3.2.1 An implementation may require to co-host multiple FMTP implementations in order to share network and/or application resources.

3.2.2 Co-hosted implementations that share the same lower layer addressing values (IP address and TCP server port number) must be distinguished by different FMTP identification values.

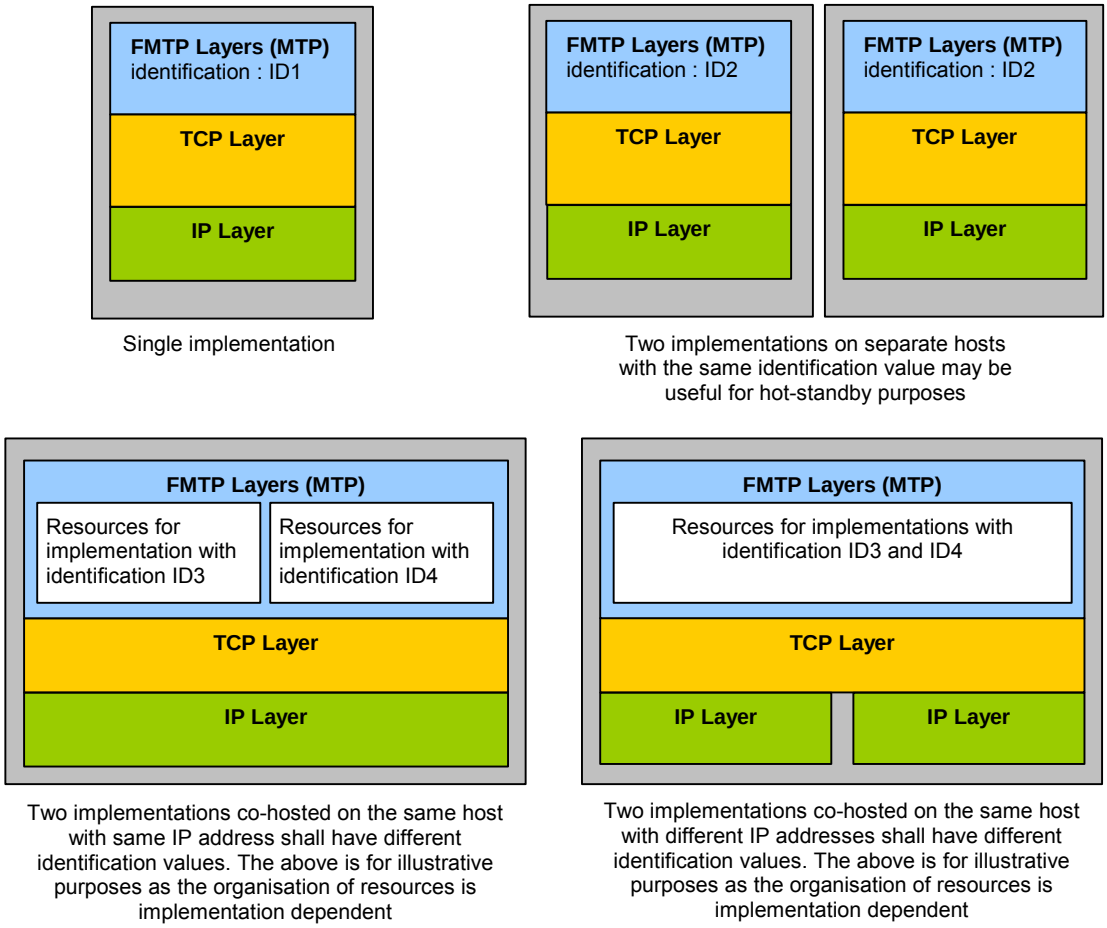

Figure 4 - Implementation Arrangement Examples

3.2.3 Although implementations are to make use of TCP port 8500, it should be a configuration parameter of the implementation.

NOTES

1. *TCP port number 8500 has been registered with the Internet Assigned Numbers Authority (IANA).*
2. *Other TCP port numbers may be agreed on a bilateral basis for testing purposes.*

3.2.4 The EC Regulation No.633/2007 specifies IPv6 [Reference 12] to ensure interoperability across EATM systems. Indeed, ANSPs have largely deployed IPv4 national networks that overlap in terms of IPv4 address domains. IPv6 has been identified as being able to provide a scalable interoperability architecture without changing existing deployments.

3.2.5 Network administrators are to assign globally scoped IPv6 addresses for implementations under their authority. The IPv6 address of the FMTP system will be derived by the local network management authority from the prefix

2001:4b50::/32 according by the local network management authority.

NOTES

1. *EUROCONTROL acts as a Local Internet Registry for the 2001:4B50::/32 prefix and assigns /48 prefixes to ANSPs and stakeholders.*
2. *Implementations that do not support IPv6 will require the use of additional devices to interwork with remote conforming IP version 6 implementations.*
3. *Such additional devices can be front-end system or a network address translation device converting between IPv4 and IPv6 (NAT-PT).*
4. *In such cases, the combination of the end-system and the external device can be considered as the implementation that conforms to the EC regulation.*
5. *Implementations should support IPv6 and IPv4 to allow for greater flexibility during the system lifecycle.*

3.2.5.1 To access the network an implementation should define a default path to a router of the provider network.

3.2.5.2 Implementations should be capable of storing or assigning more than one IP address for remote implementations. For example, in the case of connection establishment failure, a new connection with alternative IP addresses can be initiated (see Annex B).

3.3 FMTP Identification Values

3.3.1 FMTP identification value assignments are not specified in the EUROCONTROL FMTP Specification; they should be assigned in a manner to be unique to the implementation and can be as long as 32 octets.

3.3.2 To obtain uniqueness, FMTP identification values could be derived from the OLDI system letter code assignment (transferred as part of ICAO field type 3) or the ICAO location indicator.

3.4 FMTP Character Set and Bit -Ordering

3.4.1 The characters sets that have been selected for FMTP follow the recommendation of the Internet Engineering Task Force (IETF) outlined in RFC2277 [Reference 9] section 3.1.

3.4.2 The FMTP character set selection implies that characters are transmitted as single-octets in which the most significant bit (MSB) equals 0.

3.4.2.1 The FMTP Specification requires implementations to be compatible with big-endian computer systems operating over the transport interface.

NOTES

1. *This is mainly applicable to software routines that would directly write to or read from the TCP interface.*
2. *When a single or multi-octet represents a numerical quantity, the TCP transport protocol transmits the most significant bit first.*
3. *Historically, computer hardware processed two-octet entities such as 16-bit integers in one of two ways. These were termed big-endian and little-endian. Big-endian computers store the higher-order octet first, that is at the lower address in memory. As a result, when big-endian computers write data to the network interface the higher order octet appears first. On the other hand, little-endian computers store the lower-order octet first. Computer networks are designed to transmit the higher order octet which is often termed network byte order or big-endian serialisation. This means that little-endian computers need to convert the byte order of multi-octet integers when accessing the network interface.*
4. *Sub-transport layers may reverse bit-order transmission e.g. over a local area network (LAN), this is transparent to the transport protocol.*

3.5 Dual Stack

3.5.1 When a conforming implementation also implements former EUROCONTROL Standard FDE ICD Part 1 [Reference 5] this is termed a dual stack implementation.

3.5.2 Dual stack implementations are essential migration components as ATCUs will be confronted with the need to support FDE ICD Part 1 connections with some peers and FMTP connections with others. Such implementations must ensure that not more than one association with the same remote implementation is established.

Intentionally blank

4. FMTP PROTOCOL GUIDELINES

4.1 FMTP State Machine Diagrams

4.1.1 The EUROCONTROL FMTP Specification [Reference 2] defines a state machine for incoming and outgoing connections that can be illustrated as below.

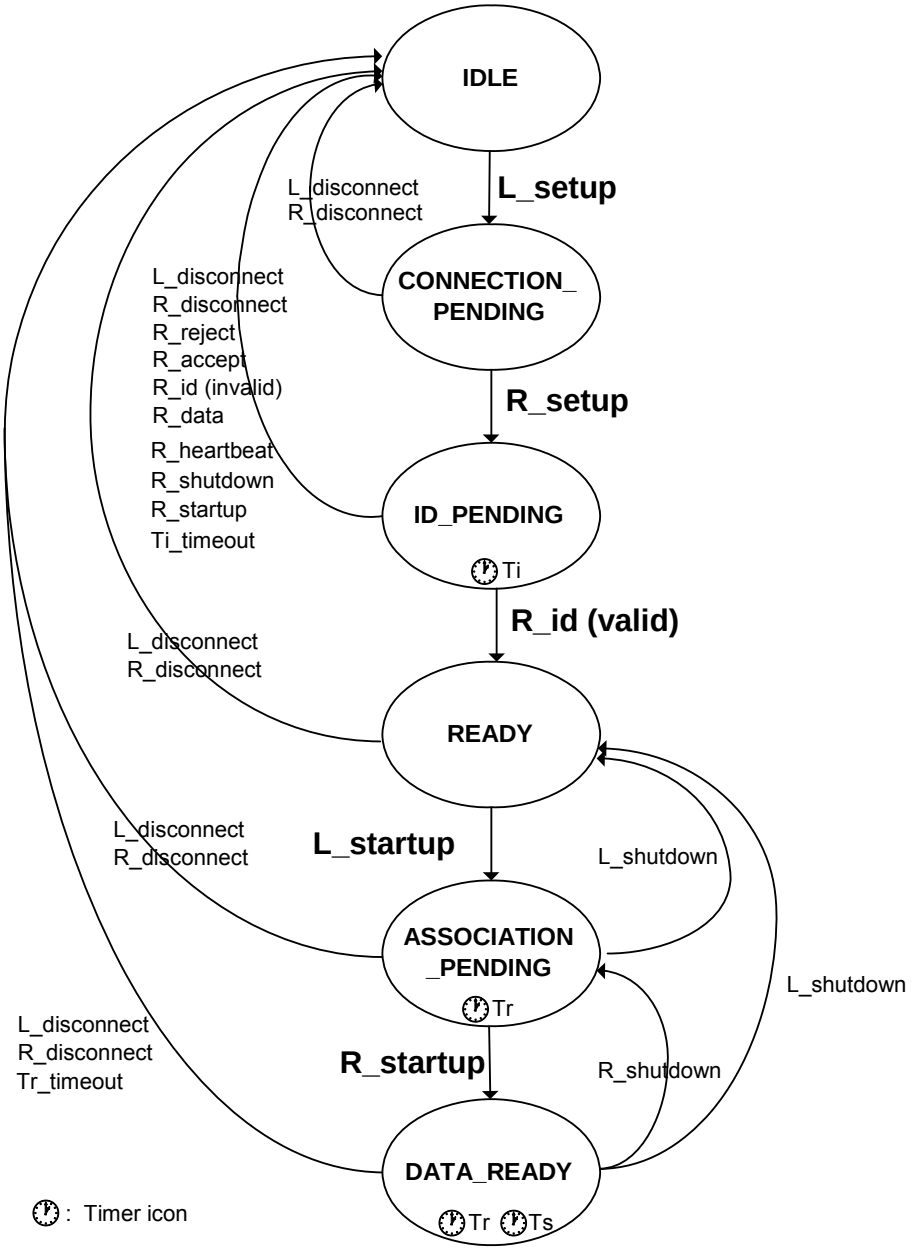

Figure 5 - State Transition Diagram: Outgoing FMTP Association Establishment

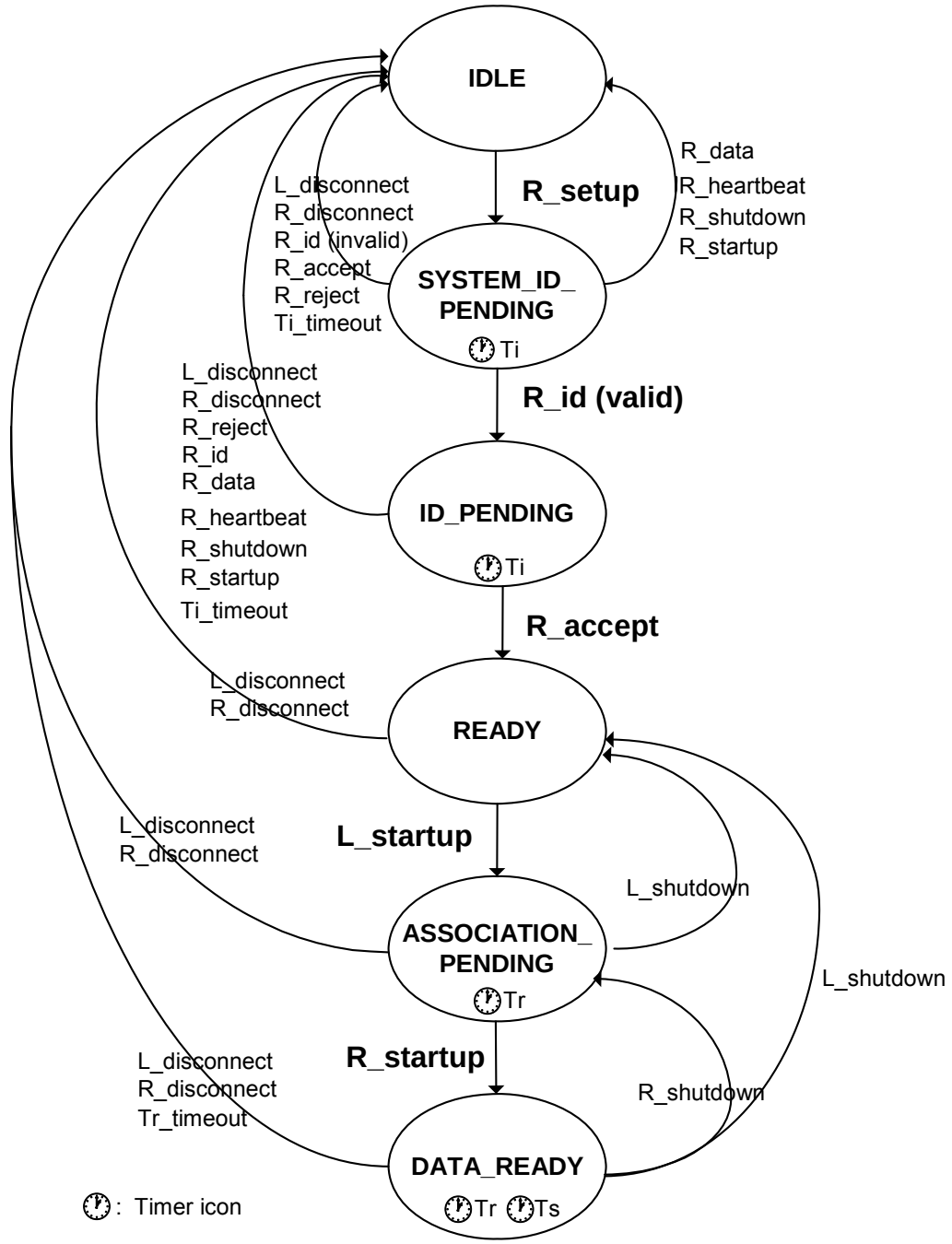

Figure 6 - State Transition Diagram: Incoming FMTP Connection Establishment

4.2 Connection Establishment

4.2.1 TCP, is a client-server protocol resulting in different outgoing and incoming state transitions. To facilitate the connection procedure the client or server roles of the implementations are to be pre-defined by bilateral agreement.

NOTE - *This is approach is often taken when establishing X.25 connections whereby the caller and called partners are pre-defined.*

4.2.2 The MT-User cannot create the main process listening on the incoming TCP port as no services for this purpose have been defined. Implementations acting as server would normally create a binding between the Message Transfer Protocol and the transport layer interface. This binding procedure may be part of the server system boot process or initiated when the application is launched which brings the server to the IDLE state; ready to serve incoming FMTP connection requests.

4.2.3 To establish an FMTP association between two implementations, it is necessary that the client initiates an FMTP connection (TCP connection establishment and system identification) by invoking the MT-CON service and that both peers invoke the MT-ASSOC service.

4.2.4 A single association is to be established between the same pair of FMTP implementations, uniquely identified by their {IP Address; TCP Port; Identification value}. This can be managed as follows:

- New FMTP connection requests, with a remote implementation with whom an active FMTP connection already exists, should be prohibited.
- The MT-User may not know whether the Message Transfer Protocol should act as TCP client or server for a given association. If the MT-User invokes the MT-CON service but the Message Transfer Protocol should act as TCP server, connection establishment with the remote implementation should be prohibited.
- FMTP implementations acting as server can make use of the REJECT message, which will fail the identification phase, if an active FMTP connection already exists with the same remote implementation.
- FMTP implementations acting as client can also make use of the REJECT message if an active FMTP connection already exists with the same remote implementation by detecting a discrepancy in the identification values sent by the server.

4.2.5 When invoking the MT-CON service, there is no need to implement a timer as a T-Disconnect→Ind service primitive will be received from the transport stack in cases of failure or time-out.

4.2.6 Once the TCP connection is established, the creation of a separate child process(es) or thread(s) may occur to support a given connection. For example, a server implementation, may wish to support a given connection with specific operating system resources.

4.2.7 Once the FMTP connection is established and enters the READY state, the protocol can be aligned with the former EUROCONTROL Standard Document FDE ICD Part 1 [Reference 5]. The protocol will remain in this state until the MT-User requests the transmission of a STARTUP message by invoking the MT-

ASSOC service.

4.3 Association Establishment

4.3.1 The FMTP protocol can remain in the READY state as long as the transport connection is not explicitly disconnected. To alert of such situations, a timer can be defined. When the timer expires it should be re-started but repetitive time-outs of this timer should be signalled to a management position.

4.3.2 The MT-ASSOC service may be invoked automatically following the MT-CON indication (this would automate the transmission of a STARTUP message) once the protocol enters the READY state.

4.3.3 To support association establishment, the protocol can remain in the ASSOCIATION_PENDING state sending STARTUPs every Tr seconds. Repetitive time-outs of the Tr timer should be signalled to a management position.

4.4 Connection and Association Integrity

4.4.1 Connection establishment is controlled by the timer Ti, which will need to higher than the round-trip time and time for the peer to generate a response. Ti can be set to a large value such as 120s.

4.4.2 The integrity of the FMTP association between two applications is ensured by the HEARTBEAT facility. The use of a TCP keep-alive mechanism is subject to bilateral agreement.

4.4.3 The HEARTBEAT facility is controlled by timers Tr and Ts which are typically set to Ts=60s, Tr=120s. These timers can be reduced to provide faster association failure detection as long as (Ts < transmit time + Tr of the peer) and vice-versa.

4.4.4 Following expiry of timer Tr in the DATA_READY state, the protocol enters the ASSOCIATION_PENDING state. The Message Transfer Protocol will try to recover through the exchange of STARTUPs without the intervention of the MT-User.

4.4.5 A communications failure can be a result of:

- the TCP transport connection fails (e.g. line failure, protocol error),
- one of the two applications or systems fails which does not terminate the association (this could be due to hardware or software failure; in some cases, the underlying TCP transport connection can still be open), or
- the remote implementation never invokes the MT-ASSOC service over an existing MT-Connection and the implementation remains in the READY or ASSOCIATION_PENDING state.

The TCP transport connection failure and detected by a T-Disconnect indication and should be interpreted as if it were a FMTP connection release by the remote implementation.

An application or system failure should be detected by the expiry of a time-out for the receipt of an expected HEARTBEAT message (timer Tr).

Excessive pending in the READY state should be signalled in order to invoke the MT-DIS service to terminate the FMTP connection and the underlying TCP transport connection.

Repetitive Tr time-outs in ASSOCIATION_PENDING should be signalled in order to invoke the MT-DIS service to terminate the FMTP connection and the underlying TCP transport connection.

4.5 Disconnection

4.5.1 The former EUROCONTROL Standard FDE ICD Part 1 [Reference 5] did not mandate a distinction between release of the FDE association and the underlying X.25 connection. The MT-STOP and MT-DIS services have been defined to differentiate between FMTP Association and/or Connection release.

4.5.2 When implementations acting as server are required to issue the T-Disconnect→Req service primitive, it should not terminate the main process listening for incoming TCP transport connection requests. It releases the TCP transport connection of the corresponding FMTP connection by terminating the TCP socket and any other processes or threads, which may have been created to the support the specific FMTP connection and/or association.

4.5.3 The MT-User cannot terminate the main process listening on the incoming TCP port as no services for this purpose have been defined. An implementation may wish to add this functionality.

4.5.4 In the absence of an existing transport connection, invoking the T-Disconnect→Req service primitive following a connection request results in the deletion of the transmission control block (typically a socket) as described in the TCP standard [Reference 8].

4.6 Data Transfer

4.6.1 The state of the protocol determines which types of messages can be transmitted. In particular:

- No data can be transferred until the TCP connection is established which is why system identification cannot occur prior or during TCP transport connection establishment.
- No MT-User data can be transferred in the CONNECTION_PENDING state, as system identification is incomplete.
- MT-User data containing operator or status messages can be transferred when in the READY, ASSOCIATION_PENDING or DATA_READY states.
- MT-User data containing operational data can only be transferred when in the DATA_READY state.
- Identification messages are transferred in the CONNECTION_PENDING, SYSTEM_ID_PENDING and ID_PENDING states.
- STARTUP messages can only be transferred when in the READY and ASSOCIATION_PENDING states.
- SHUTDOWN messages can only be transferred when in the ASSOCIATION_PENDING and DATA_READY states.
- HEARTBEAT messages can only be transferred when in the DATA_READY state.

4.6.2 TCP is a byte-stream protocol; therefore a header is required to avoid the concatenation of user messages upon receipt. The below figure illustrates the Message Transfer Protocol header followed by the data field which is used to transfer the MT-User messages.

VERSION	RESERVED	LENGTH	TYP	DATA
1 octet	*1 octet*	*2 octets*	*1 octet*	*up to 10240 octets*

Figure 7 - FMTP Header

4.6.3 The length field of the MTP header is a double-octet representing a numerical quantity indicating the size of the header and data fields. For example, a length value of 260 octets corresponds to the bit sequence 0000000100000100. It is transmitted as octet 00000001 ('1'H) first and then followed by octet 00000100 ('4'H).

4.6.4 Operational messages are defined in the EUROCONTROL OLDI specification [Reference 3] and transferred with a TYP value of 1.

4.6.5 Operator free-text messages were initially defined in the former EUROCONTROL Standard FDE ICD Part 1 [Reference 5] and are transferred with a TYP value of 2. Their format and use is subject to bilateral agreement.

4.6.6 Identification messages are defined in the EUROCONTROL FMTP Specification [Reference 2] and transferred with a TYP value of 3. Apart from the ACCEPT and REJECT identification messages, they always contains the local and remote identification values separated by a hyphen.

VERSION	RESERVED	LENGTH	TYP	Local Id	-	Remote Id
'02'H	*'00'H*	*2 octets*	*'03'H*	*up to 32 octets*	*-*	*up to 32 octets*

Figure 8 - FMTP Identification Message

VERSION	RESERVED	LENGTH	TYP	Characters A, C, C, E, P, T
'02'H	*'00'H*	*'00'H '0B'H*	*'03'H*	*'41'H '43'H '43'H '45'H '50'H '54'H*

Figure 9 - FMTP ACCEPT Identification Message

VERSION	RESERVED	LENGTH	TYP	Characters R, E, J, E, C, T
'02'H	*'00'H*	*'00'H '0B'H*	*'03'H*	*'52'H '45'H '4A'H '45'H '43'H '54'H*

Figure 10 - FMTP REJECT Identification Message

4.6.7 STARTUP, SHUTDOWN and HEARTBEAT system messages are defined in the EUROCONTROL FMTP Specification [Reference 2] and transferred with a TYP value of 4.

4.6.8 Status text messages were initially defined in the former EUROCONTROL Standard FDE ICD Part 1 [Reference 5] and are transferred with a TYP value of 5. Their format and use is subject to bilateral agreement.

4.6.9 When operator or status messages are supported, a user interface should be provided to display the received messages and to allow the creation of messages for transmission.

4.7 Security

4.7.1 The source IP address and TCP destination port number of incoming IP packets are to be validated against a local list of valid addresses for the remote system. If an invalid address is detected, the incoming IP packets shall be dropped.

4.7.2 IP address validation and the exchange of identification messages are the security provisions defined by the EUROCONTROL FMTP specification [Reference 2].

However, security can be enhanced as follows:

- Identification values can be binary encoded keys instead of ASCII characters.
- The use exchange of the identification messages is a three-way handshake; it may be enhanced by taking advantage of this 3-way handshake as a means to implement a form of Challenge-Handshake Authentication Protocol (CHAP).
- FMTP communications can be secured at the transport layer by making use of Transport Layer Security (TLS) [Reference 14]. TLS is designed to protect an underlying TCP connection against tampering, interception and forgery and is based on the former Secure Socket Layer (SSL) protocol.
- Security mechanisms such as IPsec [Reference 10] encryption, integrity and authentication can be delivered by the IP network service provider or enabled as part of the MT-CON service.

4.8 QoS and Network Management

4.8.1 For troubleshooting purposes, an implementation should enable the internet control message protocol (ICMPv6) as specified in [Reference 11].

4.8.2 The IP protocol provides a series of associated protocols and quality of service (QoS) parameters to enable or improve data delivery. In addition, applications can also make use of the IPv6 header Flow Label and Traffic Class Fields.

However, use of these features should be co-ordinated with the network administrators and alignment with the ICAO table of priorities defined in Annex 10, Volume III, Part I [Reference 18] and the Differentiated Services as defined in ICAO Document 9896 [Reference 19] should be ensured.

The following bullets provide a list of typical approaches to enable QoS for applications but if required, the use of Differentiated Services is recommended.

- RSVP proposes a way to dynamically create the aggregate reservation, classify the traffic for which the aggregate reservation applies, determine how much bandwidth is needed to achieve the requirement, and recover the bandwidth when the sub-reservations are no longer required. It also contains recommendations concerning algorithms and policies for predictive reservations.

- The 8-bit Traffic Class field in the IPv6 header is available for use by originating nodes and/or forwarding routers to identify and distinguish between different classes or priorities of IPv6 packets. This field is handled by each node of the network when forwarding the packets. It is intended to be re-named the Differentiated Service (DS) field as the primary use of this field is for the Differentiated Services QoS approach.

- The 20-bit Flow Label field in the IPv6 header may be used by a source to label sequences of packets for which it requests special handling by the IPv6 routers, such as non-default quality of service or "real-time" service. It is in the early draft stages and the primary use of this field in for the Integrated Services approach.

- Differentiated Services: Differentiation achieves scalability by aggregating traffic classification state which is conveyed by means of IP-layer packet marking using the DS field. Packets are classified and marked to receive a particular per-hop forwarding behaviour on nodes along their path. Sophisticated classification, marking, policing, and shaping operations need only be implemented at network boundaries or hosts.

4.8.3 It is custom to deploy the Simple Network Management Protocol (SNMP) between a network management station and a remote management entity in a device termed agent. The SNMP agent provides a means to access the local management information base (MIB) of the application, operating system or hardware features of a system. A management information base for FMTP is specified in Annex A to this Guideline.

4.8.4 There is no standard equivalent to the X.25 hunt group facility [Reference 17] for IP hosts, further guidelines are presented in Annex B to this Guideline.

4.9 System Performance and Testing

4.9.1 It is essential that implementations document the conformance statement (PICS) made available in Annex to the EUROCONTROL FMTP Specification.

4.9.2 To assess the system performance of an implementation it is recommended to make use of the EUROCONTROL ETIC test tool. This same reference tool can be used for extensive testing and conformance assessment.

4.9.3 An implementation should be dimensioned in order to support a minimum of 100 simultaneous TCP transport connections.

NOTE - *An implementation may use the IP stack for TCP or UDP applications other than FMTP.*

4.9.4 An implementation should be dimensioned in order to support the required peak message rate and throughput requirements. These values would be subject to the number of OLDI links, the supported message set(s) and formats (OLDI/ADEXP). Typically, operational message peak average rates of large ATCUs are of the order of 5 messages per second, generating a throughput of 4000 bits per second (aggregate of incoming and outgoing traffic). As the above values include LAM messages, we can determine from these values that the

average OLDI message length is between 120 and 180 octets.

4.9.5 When operating over satellite links, implementations should investigate the use of RFC 2488 [Reference 13].

Intentionally blank

5. USE OF FMTP FOR OTHER APPLICATIONS

5.1 Character Sets

5.1.1 Although the FMTP Specification limits data exchange to the transfer of ASCII characters, implementations may wish to include support for binary transfers as this does not modify the FMTP protocol.

This approach applies to the encoding of identification values and MT-User data.

5.2 Data Transfer

5.2.1 An FMTP implementation is to satisfy the requirement of supporting the transmission of MT-User data up to and including 10240 octets however implementations may wish to exceed this value as the length field of the MTP header allows MT-User data up to 65531 octets (2^{16} – 5 (size of header)).

5.2.2 Timers Ti, Ts, Tr can be optimised to suit the operating environment.

NOTE - *EUROCONTROL has made custom ETIC conformance sessions available through the FMT/ETIC OneSky Team.*

Intentionally blank

A. ANNEX A - SYSTEMS MANAGEMENT

A.1. SNMP Management Overview

To perform remote systems management it is custom to implement a Simple Network Management Protocol (SNMP) Management Information Base (MIB) which is remotely accessible through the associated SNMP agent.

By default, existing SNMP agents residing on hosts provide limited information on the host system. In order to extend this environment to encompass application management it is required to build a private MIB II and associated agent that interacts between external requests and the MIB II contents.

Particular care should be taken when purchasing SNMP compliant systems to ensure that support of IPv6 is planned. Furthermore, network management standards evolve constantly requiring a reassessment of this MIB prior to any implementation.

This Annex defines an example private MIB II for FMTP.

A.2. Definitions

FMTP System : an implementation to the EUROCONTROL FMTP Specification.

A.3. Assumptions

This MIB assumes that the implementation:

- is based on an IPv4 SNMP agent;
- supports both TCP client and server mode of operation;
- supports both TCP/IPv4 and TCP/IPv6 mode of communication to establish FMTP associations;
- can support multiple co-hosted systems each capable of initiating and servicing multiple FMTP associations with remote systems; and
- provides a full configuration interface, using a graphical administration console, and a configuration file which is considered as a non-volatile source of information to store and retrieve configuration information. This SNMP agent should be integrated to the implementation in order to fetch the running configuration parameters from the configuration file and dimension its SNMP tables.

Furthermore, although an implementation is uniquely identified by it's IP address and identification value, this MIB assumes that the identification values of remote implementations are unique. As a result, the identification values are sufficient to identify the remote FMTP systems within the MIB.

A.4. SNMP References

The below SNMPv2 Network Management Framework references were used at the time of developing the FMTP MIB:

1. STD 17, RFC 1213 - the core set of managed objects for the Internet suite of protocols (MIB II).
2. STD 15, RFC 1157 - the protocol used for accessing managed information.

3. RFC 1902 Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)
4. RFC 1903 Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2)
5. RFC 1904 Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2)
6. RFC 1905 Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)
7. RFC 1906 Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)
8. RFC 1907 Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)
9. RFC 1908 Coexistence between Version 1 and Version 2 of the Internet-standard Network Management Framework

A.5. MIB Overview and Structure

A.5.1. Introduction

This section gives an overview of all Managed Objects supported by the private MIB II. It is assumed that the host system already has an agent which implements the MIB-II 'system' and 'interfaces' group which are mandatory. The below defines an example application management environment which is specific to FMTP.

A.5.2. MIB Overview

A.5.2.1. Organisation

The following figure illustrates the organisation of the groups and objects that compose the management information base (MIB).

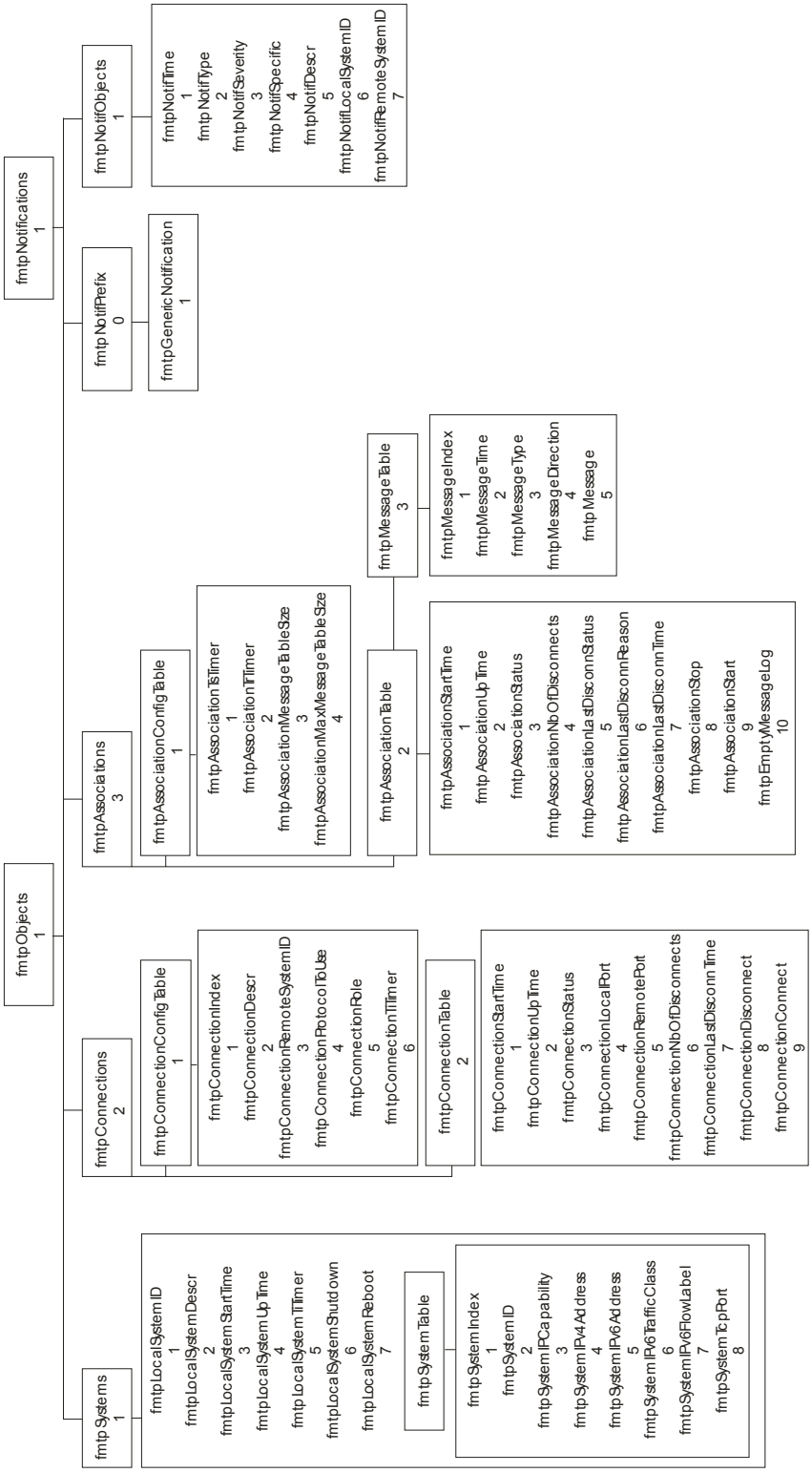

Figure A.1 - MIB Structure

A.5.2.2. Co-hosted Implementations

In the event of co-hosted implementations, SNMP sub-agent technology can be deployed. This would mean that each implementation is considered as sub-agent with the MIB specified in this Annex and reporting to the local SNMP master agent. The combination of the SNMP master agent and sub-agents acts as an SNMP agent with a single IP address.

A.5.2.3. fmtpSystems group

The fmtpSystems group represents information relative to the FMTP implementation, provided that there is only one logical FMTP implementation by SNMP agent. The values of these parameters will be read from a non-volatile source at FMTP system start-up.

The fmtpSystems group contains following objects:

- **fmtpLocalSystemID**: a unique 32-byte string value, corresponding to the ATC "system identification" parameter of the running implementation, as described in Annex B, B.2.6.2.1 of the FMTP specification.
- **fmtpLocalSystemDescr**: a free text value, describing the FMTP running implementation. Typical value could be a more user friendly description of the site, the version of the implementation or any other relevant identification information.
- **fmtpLocalSystemStartTime**: The absolute time when the FMTP implementation started.
- **fmtpLocalSystemUpTime**: The time interval from which the implementation is running, i.e. the elapsed time since the fmtpSystemStartTime.
- **fmtpLocalSystemTiTimer**: The value of the Ti timer to be applied regardless of the system initiating the TCP transport connection.
- **fmtpLocalSystemShutdown**: an integer value only aimed at being set (written to). When set to command(2), the SNMP agent will initiate the graceful shutdown of the running implementation, disconnection all TCP connections and stopping all FMTP associations.
- **fmtpLocalSystemReboot**: an integer value only aimed at being set (written to). When set to command(2), the SNMP agent will initiate the graceful shutdown and restart of the FMTP implementation.
- **fmtpSystemTable**: This table contains all the relevant properties the FMTP systems, present in the FMTP network. This table includes the local FMTP system as well as the remote FMTP systems.

A.5.2.4. fmtpSystemTable

The fmtpSystemTable is indexed by fmtpSystemIndex and contains following objects:

- **fmtpSystemIndex**: an integer value, with maximum as many values as there are remote FMTP systems. One remote host may have several entries in this table, where minimum one parameter differs.

- **fmtpSystemID**: This object contains the 32-byte FMTP system identifier, which is unique for each FMTP system within a specific connection, as described in Annex B, B.2.6.2.1 of the FMTP specification.
- **fmtpSystemIPCapability**: an integer value, with possible associated values: “IPv4”, “IPv6”, “IPv4 and IPv6”.
- **fmtpSystemIPv4Address**: a IpAddress value, giving the IPv4 address of the system, if applicable, or "0.0.0.0".
- **fmtpSystemIPv6Address**: a Ipv6Address value, giving the IPv6 address of the system, if applicable, or the value “::” (all zeroes).
- **fmtpSystemIPv6TrafficClass**: an integer value, giving the IPv6 traffic class of the system, if applicable, or the value “0”.
- **fmtpSystemIPv6FlowLabel**: an integer value, giving the IPv6 flow label field of the system, if applicable, or the value “0”.
- **fmtpSystemTcpPort**: an integer value, giving the TCP port to be connected to, when acting as server.

A.5.2.5. fmtpConnectionConfigTable

The fmtpConnectionConfigTable contains some read-write parameters related to the TCP connectivity between the local system and any remote neighbour host.

It contains references to the fmtpSystemTable, which describes in full the TCP and IP parameters of the two systems involved.

It also describes the IP capability of the remote system – the IP versions supported the system – and the role the remote system will play in the TCP transport connection set-up hand shaking.

The values of these parameters will be read from a non-volatile source at system start-up. There should be one entry per remote neighbour system.

NOTE - *Parameters in the fmtpConnectionConfigTable can be altered using the SNMP access to the FMTP system. It should be noted that their new value will only take effect when issuing a new “fmtpConnectionConnect” command or at the next FMTP system reboot.*

The fmtpConnectionConfigTable table is indexed by fmtpConnectionIndex, and contains following objects:

- **fmtpConnectionIndex**: an integer value, increasing series of integers, with as many values as there are remote FMTP systems.
- **fmtpConnectionDescr**: a string value, giving an identification of the FMTP connection to the remote system. A typical value could be the concatenation of the local fmtpSystemID and the fmtpSystemID of the remote system. Though, this is not necessarily the case.
- **fmtpConnectionRemoteSystemID**: a 32-byte string value representing the unique FMTP identifier of the host.

- **fmtpConnectionProtocolToUse**: an integer value, with possible associated values: “IPv4” and “IPv6”. It indicates what protocol will be used for the FMTP connection between the local FMTP system and the remote one.
- **fmtpConnectionRole**: an integer value, it defines the role of the remote system with possible associated values: “server” or “client”. If the value is “server”, the TCP transport connection is initiated by the local FMTP implementation, connecting to the fmtpSystemTcpPort of the remote system identified by fmtpConnectionRemoteSystemID. If value is “client” the TCP transport connection will be initiated by the remote system.
- **fmtpConnectionTiTimer**: The value of the Ti timer to be applied by the client implementation while in the identification phase.

NOTE - *The parameter fmtpConnectionRemoteSystemID refers to the identification of the FMTP system, as described in Annex B, B.2.6.2.1. of the FMTP specification.*

A.5.2.6. fmtpConnectionTable

The fmtpConnectionTable provides all read-only parameters related to FMTP connections to all remote neighbouring systems as well as two command variables. Non-existing connections will be represented with all values set to “0”, with the exception of the fmtpConnectionNbOfDisconnects and fmtpConnectionLastDisconnTime parameters.

There is one entry per remote neighbour host, the values are provided through the API of the FMTP system.

The fmtpConnections table is indexed by fmtpConnectionIndex, and contains following objects:

- **fmtpConnectionStartTime**: string timestamp value when the current FMTP connection reaches the “ready” status.
- **fmtpConnectionUpTime**: time value, elapsed time since the fmtpConnectionStartTime.
- **fmtpConnectionStatus**: an integer value, giving the status of the current FMTP connection. If the FMTP connection is not established, its value is associated with “idle”. If the FMTP connection is established, its value is associated with “ready”. Other possible values are “connection_pending”, “system_id_pending” or “id_pending”.
- **fmtpConnectionLocalPort**: an integer value, giving the TCP port number used locally for this TCP transport connection with the remote FMTP system.
- **fmtpConnectionRemotePort**: an integer value, giving the TCP port number of the remote system for this TCP transport connection.
- **fmtpConnectionNbOfDisconnects**: an integer value, giving the number of TCP connections with this remote system, that have been disconnected (reached the "idle" status) since the fmtpSystemStartTime.
- **fmtpConnectionLastDisconnTime**: string timestamp value when the last tcp disconnect occurred.

- **fmtpConnectionDisconnect**: an integer value only aimed at being set (written to). When set to command(2), the SNMP agent will initiate the disconnection of the TCP transport connection to this remote system, stopping the FMTP association, if existing.
- **fmtpConnectionConnect**: an integer value only aimed at being set (written to). When set to command(2), the SNMP agent will initiate the establishment of the TCP transport connection to this remote system, in compliance with the fmtpConnectionRole.

NOTES

1. *The parameter fmtpConnectionStatus refers to the states as described in the "State Machine Diagrams", illustrated in section 4.3 of this Guideline.*
2. *The fmtpConnectionDisconnect and fmtpConnectionConnect provide an SNMP interface to connect or disconnect the local FMTP system to or from a remote system. They refer to the MT-User interface services MT-CON and MT-DIS, described in section 3.1 of this Guideline.*

A.5.2.7. fmtpAssociationConfigTable

The fmtpAssociationConfigTable contains some read-write parameters related to the higher layer connectivity between the local system and any remote neighbour host.

It contains 2 kinds of parameters:

- timers
- message log parameters

The values of these parameters will be read from a non-volatile source at FMTP system start-up. There should be one entry per remote neighbour host.

NOTE - Parameters in the fmtpAssociationConfigTable can be altered using the SNMP access to the FMTP system. It should be noted that their new value will only take effect when issuing a "fmtpAssociationStartup" command or at the next underlying TCP transport connection connect or the next FMTP system reboot.

The fmtpAssociationConfigTable is indexed by fmtpConnectionIndex, and contains following objects:

- **fmtpAssociationTsTimer**: an integer value, giving the Ts timer.
- **fmtpAssociationTrTimer**: an integer value, giving the Tr timer.
- **fmtpAssociationMessageTableSize**: the maximum size (# of messages) of the fmtpMessages table. When more messages are exchanged, the oldest ones will be replaced.
- **fmtpAssociationMaxMessageSize**: the maximum size (# of characters) of each message. When longer messages are exchanged, their representation in the SNMP agent will be truncated.

A.5.2.8. fmtpAssociationTable

The fmtpAssociationTable provides all read-only parameters related to FMTP associations with all remote neighbouring systems as well as three command variables. Non-existing FMTP associations will be represented with all values set to "0", with the exception of the fmtpAssociationNbOfDisconnects, fmtpAssociationLastDisconnStatus, fmtpAssociationLastDisconnReason and fmtpAssociationLastDisconnTime parameters.

There is one entry per remote FMTP system which are identified through the system identification value within the fmtpSystemTable.

The fmtpAssociationTable is indexed by fmtpConnectionIndex, and contains the following objects:

- **fmtpAssociationStartTime**: string timestamp value when the current FMTP association reaches the "data_ready" status.
- **fmtpAssociationUpTime**: time value, elapsed time since the fmtpAssociationStartTime.
- **fmtpAssociationStatus**: an integer value, giving the status of the current FMTP association. If the FMTP association is not running, its value is associated with "invalid". If it is running, its value is associated with "ready", "association_pending" or "data_ready"..
- **fmtpAssociationNbOfDisconnects**: an integer value, giving the number of FMTP associations using this FMTP connection, that have been terminated (reached the "ready" or "idle" status) since the fmtpConnectionStartTime.
- **fmtpAssociationLastDisconnStatus**: an integer value, giving the status of the last FMTP associations termination. Possible other values are "idle", "connection_pending", "system_id_pending", "id_pending", "ready" or "association_pending". If no FMTP association termination occurred, its value is associated with "idle".
- **fmtpAssociationLastDisconnReason**: an integer value, giving the reason for the last disconnect. Possible values are "remote-shutdown", "local-shutdown", "ti-timeout", "tr-timeout", "local-disconnect", "remote-disconnect" or "other".
- **fmtpAssociationLastDisconnTime**: string timestamp value when the last FMTP association disconnect occurred.
- **fmtpAssociationStop**: an integer value only aimed at being set (written to). When set to command(2), the SNMP agent will initiate the stop of the FMTP association.
- **fmtpAssociationStart**: an integer value only aimed at being set (written to). When set to command(2), the SNMP agent will initiate the establishment of the FMTP association, if the underlying TCP transport connection still exists.
- **fmtpEmptyMessageLog**: an integer value only aimed at being set (written to). When set to command(2), the SNMP agent will initiate the deletion of the fmtpMessages table.

NOTES

1. *The parameter fmtpAssociationStatus refers to the states as described in the "State Machine Diagrams", illustrated in section 4.3 of this Guideline.*
2. *All parameters related to past statuses of FMTP associations, have been dropped, with the exception of fmtpAssociationNbOfDisconnects, fmtpAssociationLastDisconnStatus, fmtpAssociationLastDisconnReason and fmtpAssociationLastDisconnTime.*
3. *The fmtpAssociationCommands table provides an SNMP interface to shutdown or restart an FMTP association, using the same underlying TCP transport connection. They refer to the MT-User interface services MT-STOP and MT-ASSOC, described in section 3.1 of this Guideline.*

A.5.2.9. fmtpMessageTable

The fmtpMessageTable provides read-only information related to messages exchanged between the local FMTP system and the remote neighbouring systems. Each row will be created and added to the table as the messages are exchanged, up to fmtpAssociationMessageTableSize messages. Additional messages will replace the oldest ones.

There is one entry per exchanged message.

The fmtpMessageTable is indexed by fmtpConnectionIndex and by fmtpMessageIndex, and contains following objects:

- **fmtpMessageIndex**: an integer value, with as many increasing and neighbouring values as there have been messages exchanged, up to fmtpAssociationMessageTableSize.
- **fmtpMessageTime**: string timestamp value, based on the host system time, when the message was sent or received.
- **fmtpMessageType** : an integer value, giving the type of the message. Possible values are "operational", "operator", "identification", "system", "status" or "other".
- **fmtpMessageDirection** : an integer value, giving the direction of the message flow. Possible values are "incoming" or "outgoing".
- **fmtpMessage** : a string value, containing the first fmtpAssociationMaxMessageSize of characters of the exchanged message.

NOTES

1. *The message log will increase until it has its maximum size. New messages will replace the oldest messages. The message log can be emptied by using the fmtpEmptyMessageLog command.*
2. *All statistical information on messages has been dropped.*

A.6. ASN.1 Description

```
FMTPMIB DEFINITIONS ::= BEGIN

    IMPORTS
         Counter32, Integer32, IpAddress,
         MODULE-IDENTITY,
         NOTIFICATION-TYPE, OBJECT-TYPE, TimeTicks, enterprises
              FROM SNMPv2-SMI
         TimeInterval, DateAndTime, DisplayString, TEXTUAL-CONVENTION
              FROM SNMPv2-TC
         Ipv6Address
              FROM IPV6-TC
  OBJECT-GROUP, MODULE-COMPLIANCE, NOTIFICATION-GROUP
              FROM SNMPv2-CONF;

    fmtp MODULE-IDENTITY
         LAST-UPDATED "200502220000Z"
         ORGANIZATION "EUROCONTROL"
         CONTACT-INFO
              "E-mail: info@eurocontrol.int
              Postal: EUROCONTROL
              Rue de la fusee, 96
              B1030 Brussels
              Tel: +32(2)729 90 11"
         DESCRIPTION
              "Management information of the FMTP State Machine for
              EUROCONTROL"

    ::= { enterprises eurocontrol(3067) organisation (3)
  flightdataexchange (1) 2}

    -- Textual conventions
         CommandStatus ::= TEXTUAL-CONVENTION
         STATUS current
         DESCRIPTION
              "A status specification, including a return-status from
              an SNMP SET Request. The value 'unknown' is the default
              value, 'command' is used to issue the SNMP SET Request,
              SNMP successful provides feed-back from the agent if no
              feedback can be given concerning the execution of the
              command, command successful provides feed-back
              related to the outcome of the executed command.
              The only value that can be set is 'command'. Setting
              the other values will generate a 'wrongValue' SNMP
              error.
               'unknown': is the value returned by the agent for
              any SNMP GET request;
              . 'command': is the only value that can be SET;
              . 'snmpSuccessful': is the return value of the SNMP SET
              request, if the agent can not wait for the
              end of the execution of the command;
              . 'commandSuccessful': is the return value of the SNMP
              SET request, if the agent has waited for the end of the
```

```
          execution of the command;
          Failure of the SNMP SET request or failure in the
          execution of the command result in 'inconsistentValue'
          or 'resourceUnavailable' SNMP errors respectively."
     SYNTAX   INTEGER {
          unknown(1),                 -- default value
          command(2),                 -- sending the command
          snmpSuccessful(3), -- response value
          commandSuccessful(4)        -- response value
     }


fmtpObjects OBJECT IDENTIFIER ::= { fmtp 1 }

-- System information

fmtpSystems OBJECT IDENTIFIER ::= { fmtpObjects 1 }

-- the Lower Layer: TCP transport connection information

fmtpConnections OBJECT IDENTIFIER ::= { fmtpObjects 2 }

-- the Upper Layer: FMTP associations

fmtpAssociations OBJECT IDENTIFIER ::= { fmtpObjects 3 }

-- System information

-- the fmtpSystems group and table

fmtpLocalSystemID OBJECT-TYPE
     SYNTAX DisplayString(SIZE (32))
     MAX-ACCESS read-only
     STATUS current
     DESCRIPTION
          "A string value uniquely identifying the local fmtp
          system. It also refers to the fmtpSystemID in the
          fmtpSystemTable."
::={ fmtpSystems 1 }

fmtpLocalSystemDescr OBJECT-TYPE
     SYNTAX DisplayString (SIZE (0..64))
     MAX-ACCESS read-only
     STATUS current
     DESCRIPTION
          "This object contains a free text that describes
          the local fmtp system. It can be used
          as a label in the SNMP Management system."
::={ fmtpSystems 2 }

fmtpLocalSystemStartTime OBJECT-TYPE
     SYNTAX DateAndTime
     MAX-ACCESS read-only
     STATUS current
     DESCRIPTION
          "This object identifies the system (host) time
          this fmtp application has started. The format is
```

```
          as in 1992-5-26,13:30:15.0,-4:0"
     ::={ fmtpSystems 3 }

     fmtpLocalSystemUpTime OBJECT-TYPE
          SYNTAX TimeTicks
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
               "This object identifies the time (in hundreds of
               seconds) since the fmtpLocalSystemStartTime."
     ::={ fmtpSystems 4 }

     fmtpLocalSystemTiTimer OBJECT-TYPE
          SYNTAX TimeInterval
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
               "This object is relevant to a server implementation
               and stores the value of the Ti timer to be applied
               regardless of the system initiating the TCP transport
               connection."
     ::={ fmtpSystems 5 }

     fmtpLocalSystemShutdown OBJECT-TYPE
          SYNTAX CommandStatus
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
               "This object provides the means to shutdown
               the fmtp application.
               The value 'unknown' is returned after an SNMP
               GET request, only the value 'command' can be
               SET, the other values are returned by the agent
               after an SNMP SET request.
               Setting other values than 'command' will result
               in an 'wrongValue' SNMP error."
          DEFVAL { unknown }
     ::={ fmtpSystems 6 }

     fmtpLocalSystemReboot OBJECT-TYPE
          SYNTAX CommandStatus
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
               "This object provides the means to reboot (stop and
               start) the fmtp application.
               The value 'unknown' is returned after an SNMP
               GET request, only the value 'command' can be
               SET, the other values are returned by the agent
               after an SNMP SET request.
               Setting other values than 'command' will result
               in an 'wrongValue' SNMP error."
          DEFVAL { unknown }
     ::={ fmtpSystems 7 }

     fmtpSystemTable OBJECT-TYPE
          SYNTAX SEQUENCE OF FmtpSystemEntry
```

```
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
               "Table provides in full the identification and the
               TCP/IP parameters of every fmtp system to which the
               local system can be connected."
     ::={ fmtpSystems 8 }

     fmtpSystemEntry OBJECT-TYPE
          SYNTAX FmtpSystemEntry
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
               "An entry (conceptual row) in the fmtpSystemTable."
          INDEX { fmtpSystemIndex }
     ::={ fmtpSystemTable 1 }

          FmtpSystemEntry
          ::=SEQUENCE {
               fmtpSystemIndex
                    Integer32,
               fmtpSystemID
                    DisplayString,
               fmtpSystemIPCapability
                    INTEGER,
               fmtpSystemIPv4Address
                    IpAddress,
               fmtpSystemIPv6Address
                    Ipv6Address,
               fmtpSystemIPv6TrafficClass
                    Integer32,
               fmtpSystemIPv6FlowLabel
                    Integer32,
               fmtpSystemTcpPort
                    Integer32
          }

     fmtpSystemIndex OBJECT-TYPE
SYNTAX Integer32 (1..2147483647)
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
               "This object indexes the fmtpSystemTable"
     ::={ fmtpSystemEntry 1 }

     fmtpSystemID OBJECT-TYPE
          SYNTAX DisplayString (SIZE(32))
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
               "This object contains the 32-byte fmtp system
               identifier, which is unique for each fmtp system
               within a specific FMTP connection."
     ::={ fmtpSystemEntry 2 }

     fmtpSystemIPCapability OBJECT-TYPE
          SYNTAX INTEGER {
```

```
          ipv4(1),
          ipv6(2),
          ipv4andv6(3)
          }
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
               "This object provides the IP capability of the
               fmtp host. Possible values are ipv4(1), ipv6(2)
               or ipv4andv6(3)."
     ::={ fmtpSystemEntry 3 }

     fmtpSystemIPv4Address OBJECT-TYPE
          SYNTAX IpAddress
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
               "This object contains the IPv4 address of the
               fmtp host, or all zeros if IPv4 is not supported."
     ::={ fmtpSystemEntry 4 }

     fmtpSystemIPv6Address OBJECT-TYPE
          SYNTAX Ipv6Address
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
               "This object contains the IPv6 address of the
               fmtp host, or all zeros if IPv6 is not supported."
     ::={ fmtpSystemEntry 5 }

     fmtpSystemIPv6TrafficClass OBJECT-TYPE
          SYNTAX Integer32
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
               "This object contains the IPv6 Traffic Class
               (Priority) of the fmtp host, or zero if IPv6 is
               not supported."
     ::={ fmtpSystemEntry 6 }

     fmtpSystemIPv6FlowLabel OBJECT-TYPE
          SYNTAX Integer32
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
               "This object contains the IPv6 flow label of the
               fmtp host, or zero if IPv6 is not supported."
     ::={ fmtpSystemEntry 7 }

     fmtpSystemTcpPort OBJECT-TYPE
          SYNTAX Integer32
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
               "This object contains the TCP port used for the
               fmtp application. This port number should be 8500
               for an operational server and 8501 for a test server."
```

```
        ::={ fmtpSystemEntry 8 }

        -- the fmtp TCP Layer

        -- the fmtpConnections tables

        fmtpConnectionConfigTable OBJECT-TYPE
             SYNTAX SEQUENCE OF FmtpConnectionConfigEntry
             MAX-ACCESS not-accessible
             STATUS current
             DESCRIPTION
                  "This table contains some READ_WRITE parameters related
                  to the lower layer connectivity between the local system
                  and any remote fmtp system. It contains references to
                  the fmtpSystemTable , which describes in full the TCP
                  and IP parameters of the two systems involved. It also
                  describes the IP capacity of the remote host, the mode
                  the local system will connect or be connected, and the
                  role the remote system will play in the TCP connection
                  set-up handshaking. The values of these parameters will
                  be read from a configuration file at fmtp system start-
                  up. There should be one entry per remote fmtp system.
                  Some parameters in the table can be altered. It should
                  be noted that their new value will only take effect when
                  issuing a 'fmtpConnectionConnect' command or at the next
                  fmtp system reboot."
        ::={ fmtpConnections 1 }

        fmtpConnectionConfigEntry OBJECT-TYPE
             SYNTAX FmtpConnectionConfigEntry
             MAX-ACCESS not-accessible
             STATUS current
             DESCRIPTION
                  "An entry (conceptual row) in the
   fmtpConnectionConfigTable."
             INDEX { fmtpConnectionIndex }
        ::={ fmtpConnectionConfigTable 1 }

             FmtpConnectionConfigEntry
             ::=SEQUENCE {
                  fmtpConnectionIndex
                       Integer32,
                  fmtpConnectionDescr
                       DisplayString,
                  fmtpConnectionRemoteSystemID
                       DisplayString,
                  fmtpConnectionProtocolToUse
                       INTEGER,
                  fmtpConnectionRole
                       INTEGER,
                  fmtpConnectionTiTimer
                       TimeInterval
             }

        fmtpConnectionIndex OBJECT-TYPE
             SYNTAX Integer32 (1..2147483647)
             MAX-ACCESS not-accessible
```

```
        STATUS current
        DESCRIPTION
             "An integer value, increasing series of integers,
             with as many values as there are remote fmtp systems."
    ::={ fmtpConnectionConfigEntry 1 }

    fmtpConnectionDescr OBJECT-TYPE
        SYNTAX DisplayString (SIZE(0..64))
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
             "A string value, giving an identification of the FMTP
             connection to the remote system. A typical value could
             be the concatenation of the local fmtpLocalSystemID and
             the fmtpSystemID of the remote system. "
    ::={ fmtpConnectionConfigEntry 2 }

    fmtpConnectionRemoteSystemID OBJECT-TYPE
        SYNTAX DisplayString (SIZE(32))
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
             "A string value, referencing to the fmtpSystemID object
             in the fmtpSystems table, providing the remote host ip
             and tcp parameters."
    ::={ fmtpConnectionConfigEntry 3 }

    fmtpConnectionProtocolToUse OBJECT-TYPE
        SYNTAX INTEGER {
        ipv4(1),
        ipv6(2)
        }
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
             "An integer value, with possible associated values:
             'ipv4' and 'ipv6'. It indicates what protocol will be
             used for TCP transport connection between the local fmtp
             system and the remote one."
    ::={ fmtpConnectionConfigEntry 4 }

    fmtpConnectionRole OBJECT-TYPE
        SYNTAX INTEGER {
        client(1),
        server(2)
             }
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
             "An integer value, with possible associated values:
             'server' or 'client'.
             If this object has value 'server', the remote system
             acts as TCP server for the connection. The TCP
             connection is initiated by the local fmtp
             implementation, connecting to the fmtpSystemTcpPort
             value defined for the remote system. If this object has
             value 'client', the the remote system acts as TCP client
```

```
          for the connection. The TCP connection will be initiated
          by the remote system."
     ::={ fmtpConnectionConfigEntry 5 }

     fmtpConnectionTiTimer OBJECT-TYPE
          SYNTAX TimeInterval
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
               "This object is relevant to a client implementation and
               stores the value of the Ti timer to be applied by the
               client implementation while in the identification
               phase."
     ::={ fmtpConnectionConfigEntry 6 }

     -- the fmtpConnections table

     fmtpConnectionTable OBJECT-TYPE
          SYNTAX SEQUENCE OF FmtpConnectionEntry
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
               "A table that provides READ_ONLY parameters related to
               TCP connections to all remote systems.
               Non-existing connections will be represented with all
               values set to '0', with the exception of the
               fmtpConnectionNbOfDisconnects and
               fmtpConnectionLastDisconnTime parameters. There is one
               entry per remote system, the values
               are provided through the API of the fmtp system. The
               table also includes the commands to connect and
               disconnect the TCP connection and system
               identification."
     ::={ fmtpConnections 2 }

     fmtpConnectionEntry OBJECT-TYPE
          SYNTAX FmtpConnectionEntry
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
               "An entry (conceptual row) in the fmtpConnectionTable."
          INDEX { fmtpConnectionIndex }
     ::={ fmtpConnectionTable 1 }

          FmtpConnectionEntry
          ::=SEQUENCE {
               fmtpConnectionStartTime
                    DateAndTime,
               fmtpConnectionUpTime
                    TimeTicks,
               fmtpConnectionStatus
                    INTEGER,
               fmtpConnectionLocalPort
                    Integer32,
               fmtpConnectionRemotePort
                    Integer32,
               fmtpConnectionNbOfDisconnects
```

```
                Counter32,
            fmtpConnectionLastDisconnTime
                DateAndTime,
            fmtpConnectionDisconnect
                CommandStatus,
            fmtpConnectionConnect
                CommandStatus
        }

    fmtpConnectionStartTime OBJECT-TYPE
        SYNTAX DateAndTime
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "String timestamp value, based on the host system time,
            when the current FMTP connection reaches the 'ready'
            status."
    ::={ fmtpConnectionEntry 1 }

    fmtpConnectionUpTime OBJECT-TYPE
        SYNTAX TimeTicks
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "Time value, elapsed time since the
            fmtpConnectionStartTime."
    ::={ fmtpConnectionEntry 2 }

    fmtpConnectionStatus OBJECT-TYPE
        SYNTAX INTEGER {
        idle(1),
        connPending(2),
        sysidPending(3),
        idPending(4),
        ready(5)
        }
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "An integer value, giving the status of the current FMTP
            connection. If the connection is not established, its
            value is associated with 'idle'. If the connection is
            established, its value is associated with 'ready'.
            Other possible values are 'connPending', 'sysidPending'
            or 'idPending'."
    ::={ fmtpConnectionEntry 3 }

    fmtpConnectionLocalPort OBJECT-TYPE
        SYNTAX Integer32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "The TCP port number used locally for the TCP transport
            connection with the remote FMTP system."
    ::={ fmtpConnectionEntry 4 }

    fmtpConnectionRemotePort OBJECT-TYPE
```

```
            SYNTAX Integer32
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
                 "The TCP port number of the remote system for the
                 TCP transport connection."
       ::={ fmtpConnectionEntry 5 }

       fmtpConnectionNbOfDisconnects OBJECT-TYPE
            SYNTAX Counter32
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
                 "An integer value, giving the number of TCP connections
                 with this remote system, that have been disconnected
                 (reached the 'idle' status) since the
  fmtpSystemStartTime."
       ::={ fmtpConnectionEntry 6 }

       fmtpConnectionLastDisconnTime OBJECT-TYPE
            SYNTAX DateAndTime
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
                 "String timestamp value, based on the host system time,
                 when the last tcp disconnect occurred."
       ::={ fmtpConnectionEntry 7 }

       fmtpConnectionDisconnect OBJECT-TYPE
            SYNTAX CommandStatus
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
                 "This variable provides a way to terminate a
                 connection by setting this object to command(2).
                 The value 'unknown' is returned after an SNMP
                 GET request, only the value 'command' can be
                 SET, the other values are returned by the agent
                 after an SNMP SET request.
                 Terminating an FMTP connection also stops any running
                 FMTP association that uses it.
                 Setting other values than 'command' will result
                 in an 'wrongValue' SNMP error."
       ::={ fmtpConnectionEntry 8 }

       fmtpConnectionConnect OBJECT-TYPE
            SYNTAX CommandStatus
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
                 "This variable provides a way to start a new FMTP
                 connection by setting this object to command(2).
                 The value 'unknown' is returned after an SNMP
                 GET request, only the value 'command' can be
                 SET, the other values are returned by the agent
                 after an SNMP SET request.
                 Setting other values than 'command' will result
```

```
                in an 'wrongValue' SNMP error."
        ::={ fmtpConnectionEntry 9 }

        -- the Upper Layer: FMTP Associations

        -- the fmtpAssociations tables

        fmtpAssociationConfigTable OBJECT-TYPE
             SYNTAX SEQUENCE OF FmtpAssociationConfigEntry
             MAX-ACCESS not-accessible
             STATUS current
             DESCRIPTION
                  "This table contains all READ_WRITE parameters related
                  to the higher layer connectivity between the local
                  system and any remote neighbour host."
        ::={ fmtpAssociations 1 }

        fmtpAssociationConfigEntry OBJECT-TYPE
             SYNTAX FmtpAssociationConfigEntry
             MAX-ACCESS not-accessible
             STATUS current
             DESCRIPTION
                  "An entry (conceptual row) in the
    fmtpAssociationConfigTable."
             INDEX { fmtpConnectionIndex }
        ::={ fmtpAssociationConfigTable 1 }

             FmtpAssociationConfigEntry
             ::=SEQUENCE {
                  fmtpAssociationTsTimer
                       TimeInterval,
                  fmtpAssociationTrTimer
                       TimeInterval,
                  fmtpAssociationMessageTableSize
                       Integer32,
                  fmtpAssociationMaxMessageSize
                       Integer32
             }

        fmtpAssociationTsTimer OBJECT-TYPE
             SYNTAX TimeInterval
             MAX-ACCESS read-write
             STATUS current
             DESCRIPTION
                  "An integer value, giving the Ts timer."
        ::={ fmtpAssociationConfigEntry 1 }

        fmtpAssociationTrTimer OBJECT-TYPE
             SYNTAX TimeInterval
             MAX-ACCESS read-write
             STATUS current
             DESCRIPTION
                  "An integer value, giving the Tr timer."
        ::={ fmtpAssociationConfigEntry 2 }

        fmtpAssociationMessageTableSize OBJECT-TYPE
             SYNTAX Integer32
```

```
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
               "The maximum size (# of messages) of the fmtpMessages
               table. When more messages are exchanged, the oldest ones
               will be replaced."
     ::={ fmtpAssociationConfigEntry 3 }

     fmtpAssociationMaxMessageSize OBJECT-TYPE
          SYNTAX Integer32
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
               "The maximum size (# of characters) of each message.
               When longer messages are exchanged, their representation
               in the SNMP agent will be truncated."
     ::={ fmtpAssociationConfigEntry 4 }

     -- the fmtpAssociations table

     fmtpAssociationTable OBJECT-TYPE
          SYNTAX SEQUENCE OF FmtpAssociationEntry
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
               "A table that provides READ_ONLY parameters related to
               FMTP associations to all remote systems.
               Non-existing FMTP associations will be represented with
               all values set to '0', with the exception of the
               fmtpAssociationNbOfDisconnects,
               fmtpAssociationLastDisconnStatus
               fmtpAssociationLastDisconnReason and
               fmtpAssociationLastDisconnTime parameters. There is one
               entry per remote fmtp system, the values are provided
               through the API of the fmtp system. The table also
               includes the commands to stop and start an FMTP
               association, and to clear the message log."
     ::={ fmtpAssociations 2 }

     fmtpAssociationEntry OBJECT-TYPE
          SYNTAX FmtpAssociationEntry
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
               "An entry (conceptual row) in the fmtpAssociationTable."
          INDEX { fmtpConnectionIndex }
     ::={ fmtpAssociationTable 1 }

          FmtpAssociationEntry
          ::=SEQUENCE {
               fmtpAssociationStartTime
                    DateAndTime,
               fmtpAssociationUpTime
                    TimeTicks,
               fmtpAssociationStatus
                    INTEGER,
               fmtpAssociationNbOfDisconnects
```

```
                    Counter32,
               fmtpAssociationLastDisconnStatus
                    INTEGER,
               fmtpAssociationLastDisconnReason
                    INTEGER,
               fmtpAssociationLastDisconnTime
                    DateAndTime,
               fmtpAssociationStop
                    CommandStatus,
               fmtpAssociationStart
                    CommandStatus,
               fmtpEmptyMessageLog
                    CommandStatus
          }

     fmtpAssociationStartTime OBJECT-TYPE
          SYNTAX DateAndTime
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
               "String timestamp value, based on the host system time,
               when the current FMTP association reaches the
               'data_ready' status."
     ::={ fmtpAssociationEntry 1 }

     fmtpAssociationUpTime OBJECT-TYPE
          SYNTAX TimeTicks
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
               "Time value, elapsed time since the
               fmtpAssociationStartTime."
     ::={ fmtpAssociationEntry 2 }

     fmtpAssociationStatus OBJECT-TYPE
          SYNTAX INTEGER {
          invalid(1),
          ready (2),
          assocPending(3),
          dataReady(4)
          }
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
               "An integer value, giving the status of the current FMTP
               association. If the FMTP association is not running, its
               value is associated with 'invalid'. Other states of the
               transition are also represented by the 'invalid' value.
               If it is running, its value is associated with
               'dataReady'. Possible other value is 'assocPending'."
     ::={ fmtpAssociationEntry 3 }

     fmtpAssociationNbOfDisconnects OBJECT-TYPE
          SYNTAX Counter32
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
```

```
            "An integer value, giving the number of FMTP
            associations using this FMTP connection, that have been
            terminated (reached the 'ready' or 'idle' status) since
            the fmtpConnectionStartTime."
       ::={ fmtpAssociationEntry 4 }

       fmtpAssociationLastDisconnStatus OBJECT-TYPE
            SYNTAX INTEGER {
            idle(1),
            connPending(2),
            sysidPending(3),
            idPending(4),
            ready(5),
            assocPending(6)
            }
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
                 "An integer value, giving the status of the last FMTP
                 associations termination. Possible other values are
                 'idle', 'connPending', 'sysidPending', 'idPending',
                 'ready' or 'assocPending'. If no FMTP associations
                 termination occurred, its value is associated with
                 'idle'."
       ::={ fmtpAssociationEntry 5 }

       fmtpAssociationLastDisconnReason OBJECT-TYPE
            SYNTAX INTEGER {
            localShutdown(1),
            remoteShutdown(2),
            tiTimeout(3),
            trTimeout(4),
            localDisconnect(5),
            remoteDisconnect(6),
            other(7)
            }
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
                 "An integer value, giving the reason for the last
                 disconnect.
                 Possible values are 'remoteShutdown', 'localShutdown',
                 'tiTimeout', 'trTimeout', 'localDisconnect',
                 'remoteDisconnect' or 'other'."
       ::={ fmtpAssociationEntry 6 }

       fmtpAssociationLastDisconnTime OBJECT-TYPE
            SYNTAX DateAndTime
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
                 "String timestamp value, based on the host system time,
                 when the last FMTP association disconnect occurred."
       ::={ fmtpAssociationEntry 7 }

       fmtpAssociationStop OBJECT-TYPE
            SYNTAX CommandStatus
```

```
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
             "This variable provides a way to stop a running FMTP
             association by setting this object to command(2).
             The value 'unknown' is returned after an SNMP
             GET request, only the value 'command' can be
             SET, the other values are returned by the agent
             after an SNMP SET request.
             Setting other values than 'command' will result
             in an 'wrongValue' SNMP error."
    ::={ fmtpAssociationEntry 8 }

    fmtpAssociationStart OBJECT-TYPE
        SYNTAX CommandStatus
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
             "This variable provides a way to start a new FMTP
             association by setting this object to command(2).
             The value 'unknown' is returned after an SNMP
             GET request, only the value 'command' can be
             SET, the other values are returned by the agent
             after an SNMP SET request.
             Setting other values than 'command' will result
             in an 'wrongValue' SNMP error."
    ::={ fmtpAssociationEntry 9 }

    fmtpEmptyMessageLog OBJECT-TYPE
        SYNTAX CommandStatus
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
             "This variable provides a way to empty the
             fmtpMessageTable by setting this object to
             command(2).
             The value 'unknown' is returned after an SNMP
             GET request, only the value 'command' can be
             SET, the other values are returned by the agent
             after an SNMP SET request.
             Setting other values than 'command' will result
             in an 'wrongValue' SNMP error."
    ::={ fmtpAssociationEntry 10 }

    fmtpMessageTable OBJECT-TYPE
        SYNTAX SEQUENCE OF FmtpMessageEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
             "This table provides READ_ONLY information related
             to messages exchanged between the local fmtp system
             and the remote systems. When the table
             has been filled, new messages will replace the older
             ones. There is one entry per exchanged message."
    ::={ fmtpAssociations 3 }

    fmtpMessageEntry OBJECT-TYPE
```

```
          SYNTAX FmtpMessageEntry
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
               "An entry (conceptual row) in the fmtpMessageTable."
          INDEX { fmtpConnectionIndex , fmtpMessageIndex }
     ::={ fmtpMessageTable 1 }

          FmtpMessageEntry
          ::=SEQUENCE {
               fmtpMessageIndex
                    Integer32,
               fmtpMessageTime
                    DateAndTime,
               fmtpMessageType
                    INTEGER,
               fmtpMessageDirection
                    INTEGER,
               fmtpMessage
                    OCTET STRING
          }

     fmtpMessageIndex OBJECT-TYPE
          SYNTAX Integer32 (1..2147483647)
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
               "An integer value, with as many increasing and
               neighbouring values as there have been messages
               exchanged, up to fmtpAssociationMessageTableSize."
     ::={ fmtpMessageEntry 1 }

     fmtpMessageTime OBJECT-TYPE
          SYNTAX DateAndTime
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
               "String timestamp value, based on the host
               system time, when the message was sent or received."
     ::={ fmtpMessageEntry 2 }

     fmtpMessageType OBJECT-TYPE
          SYNTAX INTEGER {
          operational(1),
          operator(2),
          identification(3),
          system(4),
          status(5),
          other(6)
          }
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
               "An integer value, giving the type of the message.
               Possible values are 'operational', 'operator',
               'identification', 'system', 'status' or 'other'."
     ::={ fmtpMessageEntry 3 }
```

```
fmtpMessageDirection OBJECT-TYPE
     SYNTAX INTEGER {
     incoming(1),
     outgoing(2)
     }
     MAX-ACCESS read-only
     STATUS current
     DESCRIPTION
          "An integer value, giving the direction of
          the message flow. Possible values are 'incoming'
          or 'outgoing'."
::={ fmtpMessageEntry 4 }

fmtpMessage OBJECT-TYPE
     SYNTAX OCTET STRING (SIZE (0..2048))
     MAX-ACCESS read-only
     STATUS current
     DESCRIPTION
          "A string value, containing the first
          fmtpAssociationMaxMessageSize
          of characters of the exchanged message."
::={ fmtpMessageEntry 5 }

-- FMTP Notifications
fmtpNotifications
     OBJECT IDENTIFIER ::= { fmtp 2 }

fmtpNotifPrefix
     OBJECT IDENTIFIER ::= { fmtpNotifications 0 }

fmtpNotifObjects
     OBJECT IDENTIFIER ::= { fmtpNotifications 1 }

fmtpNotifTime OBJECT-TYPE
     SYNTAX DateAndTime
     MAX-ACCESS accessible-for-notify
     STATUS current
     DESCRIPTION
          "The absolute time sent as first variable in a
          notification."
::={ fmtpNotifObjects 1 }

fmtpNotifType OBJECT-TYPE
     SYNTAX Integer32
     MAX-ACCESS accessible-for-notify
     STATUS current
     DESCRIPTION
          "The event type of an FMTP notification."
::={ fmtpNotifObjects 2 }

fmtpNotifSeverity OBJECT-TYPE
     SYNTAX INTEGER {
     harmless(1),
     warning(2),
     minor(3),
     major(4),
```

```
          critical(5),
          fatal(6)
          }
          MAX-ACCESS accessible-for-notify
          STATUS current
          DESCRIPTION
               "The severity of an FMTP notification."
     ::={ fmtpNotifObjects 3 }

     fmtpNotifSpecific OBJECT-TYPE
          SYNTAX OBJECT IDENTIFIER

          MAX-ACCESS accessible-for-notify
          STATUS current
          DESCRIPTION
               "The OID of a specific SNMP variable instance
               to which this notification is related.
               If this object is not applicable, the value should
               be '0.0' (zeroDotZero)."
     ::={ fmtpNotifObjects 4 }

     fmtpNotifDescr OBJECT-TYPE
          SYNTAX DisplayString
          MAX-ACCESS accessible-for-notify
          STATUS current
          DESCRIPTION
               "A free text that describes the event."
     ::={ fmtpNotifObjects 5 }

     fmtpNotifLocalSystemID OBJECT-TYPE
          SYNTAX DisplayString (SIZE(32))
          MAX-ACCESS accessible-for-notify
          STATUS current
          DESCRIPTION
               "The system ID of the FMTP system sending the event."
     ::={ fmtpNotifObjects 6 }

     fmtpNotifRemoteSystemID OBJECT-TYPE
          SYNTAX DisplayString (SIZE(32))
          MAX-ACCESS accessible-for-notify
          STATUS current
          DESCRIPTION
               "The system ID of the remote FMTP system involved in
               the event."
     ::={ fmtpNotifObjects 7 }

     fmtpGenericNotification NOTIFICATION-TYPE
          OBJECTS { fmtpNotifTime, fmtpNotifType, fmtpNotifSeverity,
               fmtpNotifLocalSystemID, fmtpNotifRemoteSystemID,
               fmtpNotifSpecific, fmtpNotifDescr
          }
          STATUS current
          DESCRIPTION
               "This is a canonical notification canvas aimed at
               sending FMTP events to a manager.
               If the fmtpConnectionRemoteSystemID is not applicable,
               then the value of the fmtpLocalSystemID is recommended."
```

```
        ::={ fmtpNotifPrefix 1 }

        -- conformance information
        fmtpConformance
             OBJECT IDENTIFIER ::= { fmtp 3 }

        fmtpGroups
             OBJECT IDENTIFIER ::= { fmtpConformance 1 }
        fmtpCompliances
             OBJECT IDENTIFIER ::= { fmtpConformance 2 }

        fmtpCompliance MODULE-COMPLIANCE
             STATUS current
             DESCRIPTION
                  "EUROCONTROL Flight Message Transfer Protocol"
             REFERENCE
                  "EGIS.COM.FMTP"
             MODULE FMTPMIB
             MANDATORY-GROUPS {
        fmtpSystemGroup, fmtpConnectionGroup,
        fmtpAssociationGroup
             }
        ::={ fmtpCompliances 1 }

        -- Units of conformance

        fmtpSystemGroup OBJECT-GROUP
             OBJECTS {
                  fmtpLocalSystemDescr, fmtpLocalSystemStartTime,
                  fmtpLocalSystemUpTime, fmtpLocalSystemTiTimer,
                  fmtpLocalSystemShutdown, fmtpLocalSystemReboot,
                  fmtpSystemID, fmtpSystemIPCapability,
                  fmtpSystemIPv4Address, fmtpSystemIPv6Address,
                  fmtpSystemIPv6TrafficClass, fmtpSystemIPv6FlowLabel,
                  fmtpSystemTcpPort
             }
             STATUS current
             DESCRIPTION
                  "The FMTP system related objects are compulsory."
        ::={ fmtpGroups 1 }

        fmtpConnectionGroup OBJECT-GROUP
             OBJECTS {
                  fmtpConnectionDescr, fmtpLocalSystemID,
                  fmtpConnectionRemoteSystemID,
    fmtpConnectionProtocolToUse,
                  fmtpConnectionRole, fmtpConnectionTiTimer,
                  fmtpConnectionStartTime, fmtpConnectionUpTime,
                  fmtpConnectionStatus, fmtpConnectionLocalPort,
                  fmtpConnectionRemotePort, fmtpConnectionNbOfDisconnects,
                  fmtpConnectionLastDisconnTime, fmtpConnectionDisconnect,
                  fmtpConnectionConnect
             }
             STATUS current
             DESCRIPTION
                  "The connection related objects are compulsory."
        ::={ fmtpGroups 2 }
```

```
     fmtpAssociationGroup OBJECT-GROUP
          OBJECTS {
               fmtpAssociationTsTimer, fmtpAssociationTrTimer,
               fmtpAssociationStartTime, fmtpAssociationUpTime,
               fmtpAssociationStatus, fmtpAssociationNbOfDisconnects,
               fmtpAssociationLastDisconnStatus,
fmtpAssociationLastDisconnReason,
               fmtpAssociationLastDisconnTime, fmtpAssociationStop,
               fmtpAssociationStart
          }
          STATUS current
          DESCRIPTION
               "The FMTP association related objects are compulsory."
     ::={ fmtpGroups 3 }

     fmtpMessageGroup OBJECT-GROUP
          OBJECTS {
               fmtpAssociationMessageTableSize,
fmtpAssociationMaxMessageSize,
               fmtpEmptyMessageLog, fmtpMessageTime, fmtpMessageType,
               fmtpMessageDirection, fmtpMessage
          }
          STATUS current
          DESCRIPTION
               "The fmtpMessageTable is optional."
     ::={ fmtpGroups 4 }

     fmtpNotificationObjectGroup OBJECT-GROUP
          OBJECTS {
               fmtpNotifTime, fmtpNotifType, fmtpNotifSeverity,
               fmtpNotifSpecific, fmtpNotifDescr,
               fmtpNotifLocalSystemID, fmtpNotifRemoteSystemID
          }
          STATUS current
          DESCRIPTION
               "The notification objects are required only when
               the fmtpNotificationGroup is implemented."
     ::={ fmtpGroups 5 }

     fmtpNotificationGroup NOTIFICATION-GROUP
          NOTIFICATIONS {
               fmtpGenericNotification
          }
          STATUS current
          DESCRIPTION
               "The Generic Notification is optional."
     ::={ fmtpGroups 6 }

     END
```

Intentionally blank

B. ANNEX B - SERVICE AVAILABILITY AND RELIABILITY

B.1. Introduction

B.1.1. The hunt group optional user facility of X.25 [Reference 17] allowed FDE ICD Part 1 implementations to improve service availability and reliability. Indeed, this user facility, if subscribed to, distributes incoming calls having an address associated with the hunt group across a designated grouping of DTE/DCE interfaces.

B.1.2. In IP there is no standard protocol element equivalent to the X.25 hunt group facility. However, a series of operating system dependent or vendor dependent products can achieve a similar function. They are all based on the concept of clustering servers accessible via a name or virtual IP address known to remote implementations. The name or IP address is then translated to the real physical IP address of the active end-system within a cluster or translated to one of the servers as a result of load-balancing methods.

B.2. Methods

B.2.1. Multiple Target IP Addresses

This method involves the assignment of several IP addresses to reach the remote implementation. It is assumed that each IP address corresponds to different physical resource. If connection establishment fails with a given address, connection establishment is then attempted with the alternative IP address(es).

B.2.2. Round-Robin DNS

Round-Robin DNS, maps a single name to the different IP address in a round-robin manner; thus different clients will be mapped to different servers in the cluster for the ideal situation. In this way, the load is distributed amongst the servers. It is not such a reliable method, as when a server fails, clients who mapped the name to the IP address of that server will still find the server is unavailable until there is a DNS query update.

B.2.3. Dedicated Load-Balancers

B.2.3.1. A load-balancer acting as a front-end to of cluster can distribute load amongst the servers or be designed to forward to one active server until failure. The servers can be made to appear as one virtual service by defining a single IP address, so that the end users see a virtual server and not a cluster of servers. Load balancing can be achieved per connection and failures can be hidden to the remote implementations.

B.2.3.2. Load balancing can be implemented at the IP or the application layer. However, load-balancing at IP-level gives a better opportunity to access commercial products and stands a better chance of being operating system independent. Furthermore, load-balancing at application-level would involve embedding the load-balancer with protocol elements of this specification.

B.2.3.3. Load-balancing at IP-level involves the translation of the IP address known to remote implementations to the addresses of the real servers. This is based on network address translation.

B.2.3.4. Load-balancers create a new single-point failure, therefore it must be implemented in a fault-tolerant system fashion. Typically, this involves the exchanges of heartbeat messages between load-balancers and switch-over techniques.

B.2.4. Architecture Example

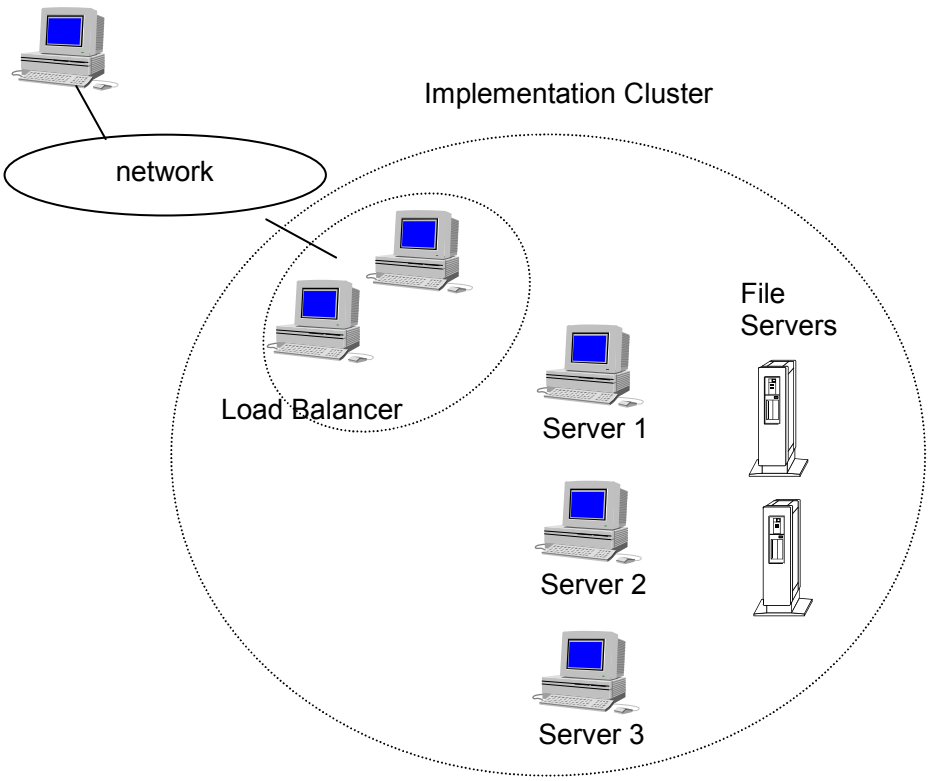

Figure B.1 - Reliable Architecture Example

C. ANNEX C - CONFORMANCE TESTING METHODOLOGY

C.1. Introduction

C.1.1. It is important that implementations of FMTP are such that there is a high level of confidence for interoperation between Air Traffic Control Units (ATCUs).

C.1.2. Implementations of FMTP are undertaken by Member States and Air Navigation Service Providers in a manner that is likely to rely on procurement from various sources. To achieve a high level of confidence that such implementations will interoperate, a common set of conformance test requirements is required to standardise preparation for test, testing and presentation of results.

C.2. Methods and Practices

C.2.1. Member States or Air Navigation Service Providers that need to comply with the FMTP Implementing Rule need to perform a 'Verification of Systems' as described in its Article 5. However, as a minimum all Member State or Air Navigation Service Providers need to complete the conformance statements indicated in the EUROCONTROL FMTP Specification Annex A [Reference 2].

C.3. Testing FMTP Implementations

C.3.1. Introduction

C.3.1.1. The PICS proformas that are included in the EUROCONTROL FMTP Specifications Annex A, can be used as the first step in performing a conformance test.

C.3.1.2. In order to provide confidence in and support for FMTP Interface within an ATCU to the interworking between co-operating FMTP applications, it is desirable for each to be tested for conformance against external references e.g. test documents or a common test tool. Such testing is focuses on the external behaviour of the System Under Test (SuT) and is intended to test for interworking rather than the serviceability of the end system.

C.3.2. Testing of the FMTP Protocol

C.3.2.1. EUROCONTROL has developed a test-tool named ETIC version 3.0 that can be used to validate conformity to the FMTP protocol.

C.3.2.2. EUROCONTROL has developed an abstract interoperability test plan that can be used to validate interoperability between co-operating ATCUs [Reference 6].

C.3.2.3. A series of bilaterally agreed tests should be agreed and conducted between co-operating ATCUs on the basis of [Reference 6].

C.3.2.4. The results of tests should be recorded and agreed between the co-operating parties.

C.4. Notification

C.4.1. Member States and Air Navigation Service Providers that are required to comply with the FMTP Implementing Rule need to follow the notification procedure defined in its Annex IV.

C.4.2. Member States and Air Navigation Service Providers should complete the result template of the interoperability test plan [Reference 6].

C.4.3. Member States and Air Navigation Service Providers should forward test plan result templates or details of any test results to EUROCONTROL.

C.4.4. Member States and Air Navigation Service Providers should forward the ETIC conformance test certificate to be made available on the FMTP/ETIC EUROCONTROL OneSky Team.

C.4.5. Member States and Air Navigation Service Providers should upload the ETIC conformance test certificate to the EUROCONTROL on-line FMTP Database.