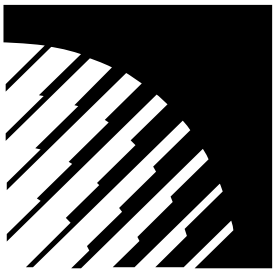

The ATM Forum

Technical Committee

Multi-service Extensions to FUNI v2.0 Specification

AF-SAA-0109.000

February,1999

Multi-service Extensions to FUNI v2.0 Specification
Version 01.02

© 1999 by The ATM Forum. This specification/document may be reproduced and distributed in whole, but (except as provided in the next sentence) not in part, for internal and informational use only and not for commercial distribution. Notwithstanding the foregoing sentence, any protocol implementation conformance statements (PICS) or implementation conformance statements (ICS) contained in this specification/document may be separately reproduced and distributed provided that it is reproduced and distributed in whole, but not in part, for uses other than commercial distribution. All other rights reserved. Except as expressly stated in this notice, no part of this specification/document may be reproduced or transmitted in any form or by any means, or stored in any information storage and retrieval system, without the prior written permission of The ATM Forum.

The information in this publication is believed to be accurate as of its publication date. Such information is subject to change without notice and The ATM Forum is not responsible for any errors. The ATM Forum does not assume any responsibility to update or correct any information in this publication. Notwithstanding anything to the contrary, neither The ATM Forum nor the publisher make any representation or warranty, expressed or implied, concerning the completeness, accuracy, or applicability of any information contained in this publication. No liability of any kind shall be assumed by The ATM Forum or the publisher as a result of reliance upon any information contained in this publication.

The receipt or any use of this document or its contents does not in any way create by implication or otherwise:

- Any express or implied license or right to or under any ATM Forum member company's patent, copyright, trademark or trade secret rights which are or may be associated with the ideas, techniques, concepts or expressions contained herein; nor
- Any warranty or representation that any ATM Forum member companies will announce any product(s) and/or service(s) related thereto, or if such announcements are made, that such announced product(s) and/or service(s) embody any or all of the ideas, technologies, or concepts contained herein; nor
- Any form of relationship between any ATM Forum member companies and the recipient or user of this document.

Implementation or use of specific ATM standards or recommendations and ATM Forum specifications will be voluntary, and no company shall agree or be obliged to implement them by virtue of participation in The ATM Forum.

The ATM Forum is a non-profit international organization accelerating industry cooperation on ATM technology. The ATM Forum does not, expressly or otherwise, endorse or promote any specific products or services.

NOTE: The user's attention is called to the possibility that implementation of the ATM interoperability specification contained herein may require use of an invention covered by patent rights held by ATM Forum Member companies or others. By publication of this ATM interoperability specification, no position is taken by The ATM Forum with respect to validity of any patent claims or of any patent rights related thereto or the ability to obtain the license to use such rights. ATM Forum Member companies agree to grant licenses under the relevant patents they own on reasonable and nondiscriminatory terms and conditions to applicants desiring to obtain such a license. For additional information contact:

The ATM Forum
Worldwide Headquarters
2570 West El Camino Real, Suite 304
Mountain View, CA 94040-1313
Tel: +1-650-949-6700
Fax: +1-650-949-6705

TABLE OF CONTENTS

1. SCOPE 4

2. INTRODUCTION 4

2.1 PURPOSE 4
2.2 TERMINOLOGY 4
2.3 ACRONYMS LIST 5

3. RELEVANT SPECIFICATIONS 5

4. REFERENCE MODEL 6

5. FRAGMENTATION/REASSEMBLY OVERVIEW 6

6. FRAGMENTATION/REASSEMBLY MODEL 7

7. FRAGMENTATION/REASSEMBLY FORMAT 8

8. FRAGMENTATION PROCEDURE 9

8.1 FRAGMENTATION PROCEDURE 9
8.2 REASSEMBLY PROCEDURE 10

9. FRAGMENT AND FRAME SIZES 11

10. FRAGMENTATION EXAMPLE 12

11. INFORMATIVE ANNEX A – DYNAMIC FRAGMENTATION 13

TABLE OF FIGURES

FIGURE 4-1 REFERENCE MODEL OF THE MULTI-SERVICE FUNI 6
FIGURE 6-1 FUNI FRAGMENTATION/REASSEMBLY REFERENCE DIAGRAM 7
FIGURE 7-1 FUNI FRAME FRAGMENT FORMAT 8
FIGURE 10-1 AN EXAMPLE OF FRAGMENTATION 12

1. SCOPE

The scope of this document is to provide the necessary incremental specifications to add to the FUNI v2.0 [1] compliant devices the capability to handle multi-services, particularly Voice over Frame Based UNI.

The specifications in this document are additions to the FUNI v2.0 [1] specifications. Any changes from the FUNI v2.0 [1] specifications will be explicitly pointed out and will not be required for devices to claim compliance with FUNI v2.0 [1]. Such changes, if any, are only required to claim compliance with the specifications stated in this document.

2. INTRODUCTION

PURPOSE

In order to be able to mix real-time/delay-sensitive traffic, such as voice, with traffic that is less sensitive to delay, such as asynchronous data, on the same interface, it is necessary to reduce the size of all packets and interleave them. For this reason, fragmenting long packets on the multi-service FUNI is necessary.

This specification provides the fragmentation and reassembly process for the transmitting and receiving DTEs/DCEs respectively. This specification, which supports fragmentation/ reassembly across a FUNI interface between the DTE-DCE peers, is based on the frame relay fragmentation/reassembly process specified in the Frame Relay Forum's FRF.12 [3] implementation agreement.

TERMINOLOGY

In this specification **(R)** and **Shall** indicate a requirement that must be implemented to claim compliance with this specification. **(O)** is an Option that may be implemented. **(CR)** is a Conditional Requirement which must be implemented if the particular option, to which it is related, is implemented.

When Should is used in the context of a specification, the implementation of the item referred to, as stated, is highly desirable, but not required to claim compliance.

When May or Optional is used, the implementation of the item referred to is Not Required, and is left to the discretion of the implementers.

When the statement "Not Applicable" is used, the item is outside the scope of this specification.

ACRONYMS LIST

ATM - Asynchronous Transfer Mode

DCE - Data Communications Equipment

DTE - Data Terminal Equipment

FA - Frame Address

FCS - Frame Check Sequence

FUNI - Frame-based User-to-Network Interface

PPP - Point-to-Point Protocol

PVC - Permanent Virtual Connection

QoS - Quality of Service

SVC - Switched Virtual Connection

VC - Virtual Connection

3. RELEVANT SPECIFICATIONS

The following is a list of specifications on which this specification is based:

1. ATM Forum af-saa-0088.000, “Frame Based User-To-Network Interface (FUNI) Specification v2.0”, July 1997.
2. IETF RFC 1990, K. Sklower, B. Lloyd, G. McGregor, D. Carr, T. Coradetti, “The PPP Multilink Protocol (MP)”, August 1996.
3. Frame Relay Forum FRF.12, “Frame Relay Fragmentation Implementation Agreement,” December 1997.

4. REFERENCE MODEL

Figure 4-1 below depicts a reference model of the Multi-service FUNI, where user's real-time (e.g., voice) and asynchronous (e.g., data) services can be multiplexed and handled on a single FUNI interface to the ATM network.

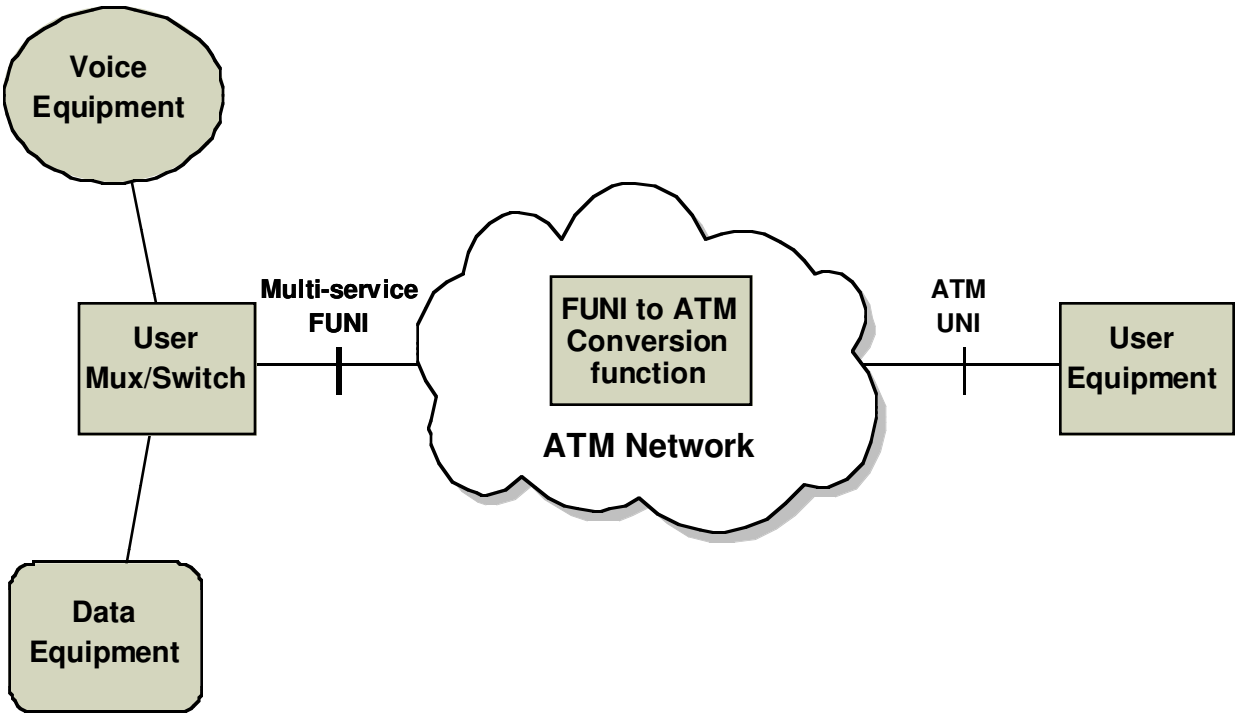

Figure 4-1 Reference Model of the Multi-Service FUNI

5. FRAGMENTATION/REASSEMBLY OVERVIEW

To properly support voice and other real-time (delay-sensitive) data on lower-speed FUNI links, it is necessary to fragment long frames that share the same FUNI link into shorter frames. The shorter frames eliminate excessive frame delays which could be encountered with longer frames. Fragmentation enables interleaving delay-sensitive traffic on one VC with fragments of a long frame on another VC utilizing the same interface.

Fragmentation may be used with both two-octet and four-octet FUNI headers, and both two-octet and four-octet FCS fields. To simplify the diagrams, the figures that include FUNI headers and FCS fields will only illustrate the two-octet header and the two-octet FCS. The four-octet

header and/or four-octet FCS differ from the shown diagrams only in the size of the respective field.

6. FRAGMENTATION/REASSEMBLY MODEL

FUNI Bidirectional DTE-DCE fragmentation/reassembly is used in order to allow real-time and data frames to share the same FUNI interface between a DTE and the ATM network. Fragmentation/reassembly is strictly local to the interface, and the fragment size can be optimally configured to provide the proper delay and delay variation based upon the logical speed of the DTE interface (the logical speed of an interface may be slower than the physical clocking rate if a channelized physical interface is used).

For FUNI fragmentation, the DTE and DCE interfaces act as fragmentation and reassembly peers, as shown in Figure 6.1:

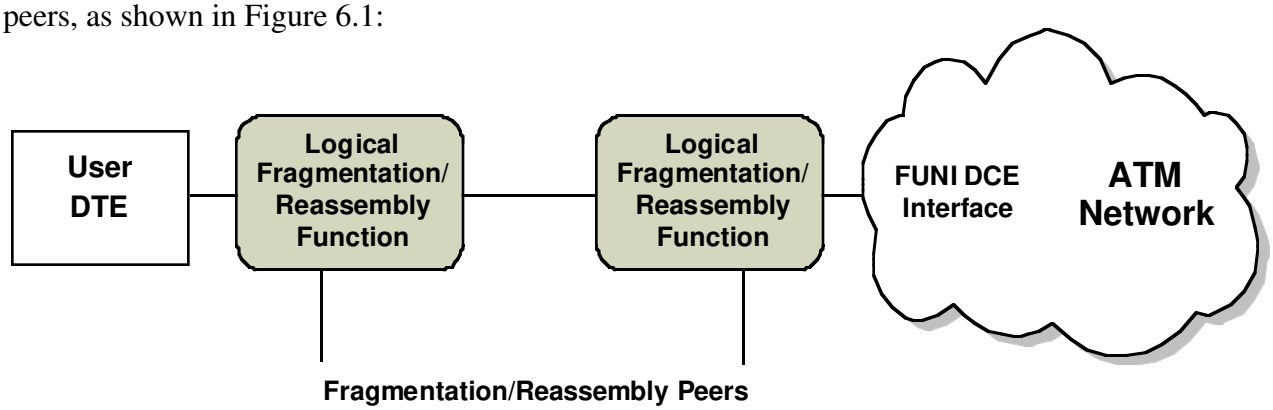

Figure 6-1 FUNI Fragmentation/Reassembly Reference Diagram

Note that the "Logical Fragmentation/Reassembly Function" depicted in Figure 6.1 is shown as a standalone function for illustration purpose only. This function is expected to be implemented as part of the DTE and DCE interface implementations.

FUNI fragmentation/reassembly is provisioned on an interface-by-interface basis. When FUNI fragmentation/reassembly is used on an interface, then all frames on all VCs (including all PVCs, SVCs, and signaling channels) are preceded by the fragmentation header (described in Section 7).

7. FRAGMENTATION/REASSEMBLY FORMAT

O-01 Fragmentation/Reassembly May be used to enable multiple services on the same FUNI.

O-02 Fragmentation/Reassembly May be used with both two-octet and four-octet FUNI headers and both two-octet and four-octet FCS fields.

When Fragmentation/Reassembly is used on a FUNI, the following Requirements Apply

R-01 A two-octet Fragmentation Header Shall precede the FUNI header.

R-02 The format for each fragment of a frame Shall be as shown in Figure 7-1 below for a two-octet FUNI Header. Fragment format for a four-octet FUNI header differs from Figure 7-1 only in the size of the FUNI header. The four-octet FUNI Header format Shall be the same as that shown in af-saa-0088.000 [1], CR4, Section 3.9. The Fragment Payload Shall be a Fragment of the FUNI Frame Payload of a length determined by the FUNI user.

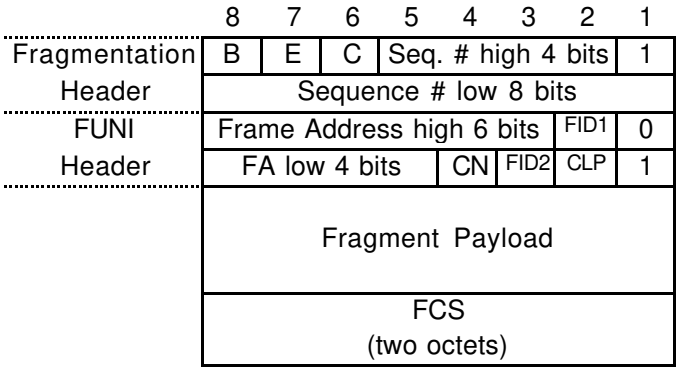

Figure 7-1 FUNI Frame Fragment Format

R-03 The value of the different fields in the Fragment Format Shall be as follows **:**

The (B)eginning fragment bit is a one-bit field set to 1 on the first frame fragment derived from the original frame and set to 0 for all other fragments from the same frame.

The (E)nding fragment bit is a one-bit field set to 1 on the last fragment of a frame and set to 0 for all other fragments of the same frame. A frame fragment may have both the (B)eginning and (E)nding fragment bits set to 1 when the whole frame fits in one fragment.

The (C)ontrol bit is set to 0 in all fragments. It is reserved for future control functions.

The sequence number is a 12-bit binary number that is incremented modulo 2^{12} for every frame fragment transmitted on a VC. There is a separate sequence number maintained for each VC across the interface.

R-03, Continued

The lowest order bit of the first octet of the fragmentation header (bit 1 of octet 1 in Figure 7-1) Shall be set to 1. This is to distinguish the fragmentation header from the FUNI header (which contains a 0 in the corresponding field).

R-04 Peer entities (DTE or DCE) of a FUNI Shall be identically configured for fragmentation functionality, i.e., both Shall be configured with fragmentation enabled or both with fragmentation disabled.

R-05 A peer that is configured for fragmentation Shall discard received frames that do not contain the fragmentation header (lowest order bit in the first octet of the header is not set to 1).

R-06 A peer that is not configured for fragmentation Shall discard received frames containing a fragmentation header due to the violation of the FUNI header format (lowest order bit of the header's first octet being set to 1 instead of 0).

8. FRAGMENTATION PROCEDURE

FRAGMENTATION PROCEDURE

This fragmentation procedure is based on RFC 1990 [2]. Note that the procedure is identical for the two-octet and the four-octet FUNI header formats.

A FUNI transmitter Shall perform Frame Fragmentation according to the following procedure:

R-07 A series of frame fragments is created by removing the leading flag, the FUNI header, the original FCS and the trailing flag from the FUNI Frame and sending the remaining octets (FUNI Frame Payload), in their original order, as a series of frame fragments.

R-08 The resulting fragments Shall be transmitted in the same order as they occurred in the original frame prior to being fragmented.

Fragments from multiple VCs may be interleaved with each other on one interface (this is the principal objective of fragmentation).

R-09 Every fragment in the series Shall contain a fragmentation header and a copy of the FUNI header that was on the original unfragmented frame, including the CN, CLP, and FID bits.

R-10 The B bit Shall be set (to 1) in the first fragment of each frame. The E bit Shall be set (to 1) in the last fragment of each frame.

R-11 Following a VC becoming active, the first fragment sent on that VC may have the sequence number set to any value (including zero). The sequence number Shall subsequently be incremented by one for each fragment sent. The sequence number Shall be incremented without regard to the original frame boundaries, i.e., if the last fragment in one frame used sequence number "N", then the first fragment of the following frame Shall use sequence number "N+1" modulo 2^{12}.

R-12 Each VC Shall have its own fragmentation sequence number, independent of all other VCs.

The last two requirements (R-11 and R-12) enable detection of lost fragments (and bursts of lost fragments) independently for all active VCs.

Note that if sufficient fragments are sent on an active VC, the sequence number will wrap from all ones to zero, and will eventually also wrap past the original sequence number sent on that VC after it became active. This wrapping may or may not occur on an original FUNI frame boundary (it is transparent to frame boundaries).

See Section 110 for a fragmentation example.

REASSEMBLY PROCEDURE

A FUNI receiver Shall perform Reassembly of received frame fragments into the original FUNI frames according to the following procedure:

R-13 For each VC, the receiver Shall keep track of the incoming sequence numbers and maintain the most recently received sequence number. The receiver Shall detect the end of a reassembled frame when it receives a fragment with the (E)nding bit set to 1. Reassembly of the frame Shall be declared complete (and the frame delivered to the next network function) if all sequence numbers up to that fragment have been received.

R-14 Each of the congestion bits CN and CLP, from all the received FUNI Header copies in all the received fragments of one FUNI Frame, Shall be logically ORed. The resulting CN and CLP bits Shall be included in the reassembled FUNI Frame Header.

R-15 The receiver Shall detect lost fragments on a given VC when one or more sequence numbers from that VC are skipped. When a lost fragment or fragments are detected on a VC, the receiver Shall discard all currently incomplete frame reassemblies, and subsequently received fragments, for that VC until it receives the next fragment that bears the (B)eginning bit. The fragment bearing the (B)eginning bit Shall be used to begin reassembling a new frame.

R-16 In the event of an error (e.g., one or more fragments lost due to transmission error, or a reassembly buffer overflow), fragments which cannot be reconstructed back into the original frame Shall be discarded by the receiver.

9. FRAGMENT AND FRAME SIZES

This FUNI Specification does not require nor recommend any specific fragment size. The fragment size is configured in the transmitter, and two peer transmitters need not use the same fragment size. The above fragmentation and reassembly procedures assure proper interoperability with different fragment sizes.

The maximum fragment size should be configured on a per-interface basis. The optimal fragment size is the result of a tradeoff between the efficiency of large frames, the interface speed, and the required delay and delay variation characteristics of the applications. This is left to the discretion of the user.

R-17 Receivers Shall be able to reassemble complete frames up to at least 4096 or 9232 octets in length, depending on the FUNI mode in use. For the proper size, follow requirements in the af-saa-0088.000 [1] specification.

10. FRAGMENTATION EXAMPLE

An example of the FUNI fragmentation procedure is depicted in Figure 10-1. The octets in white indicate fragments of the payload of the original FUNI frame (three fragments in this example). For this example, the starting sequence number of 42 was chosen at random.

The reassembly process Logical ORs each of the CN and CLP bits in the three received FUNI Header copies (from the three fragments) and inserts them in the reassembled FUNI Frame Header. It reassembles the three data fragments into one FUNI Frame payload and inserts a new Frame Check Sequence at the end of that reassembled FUNI Frame.

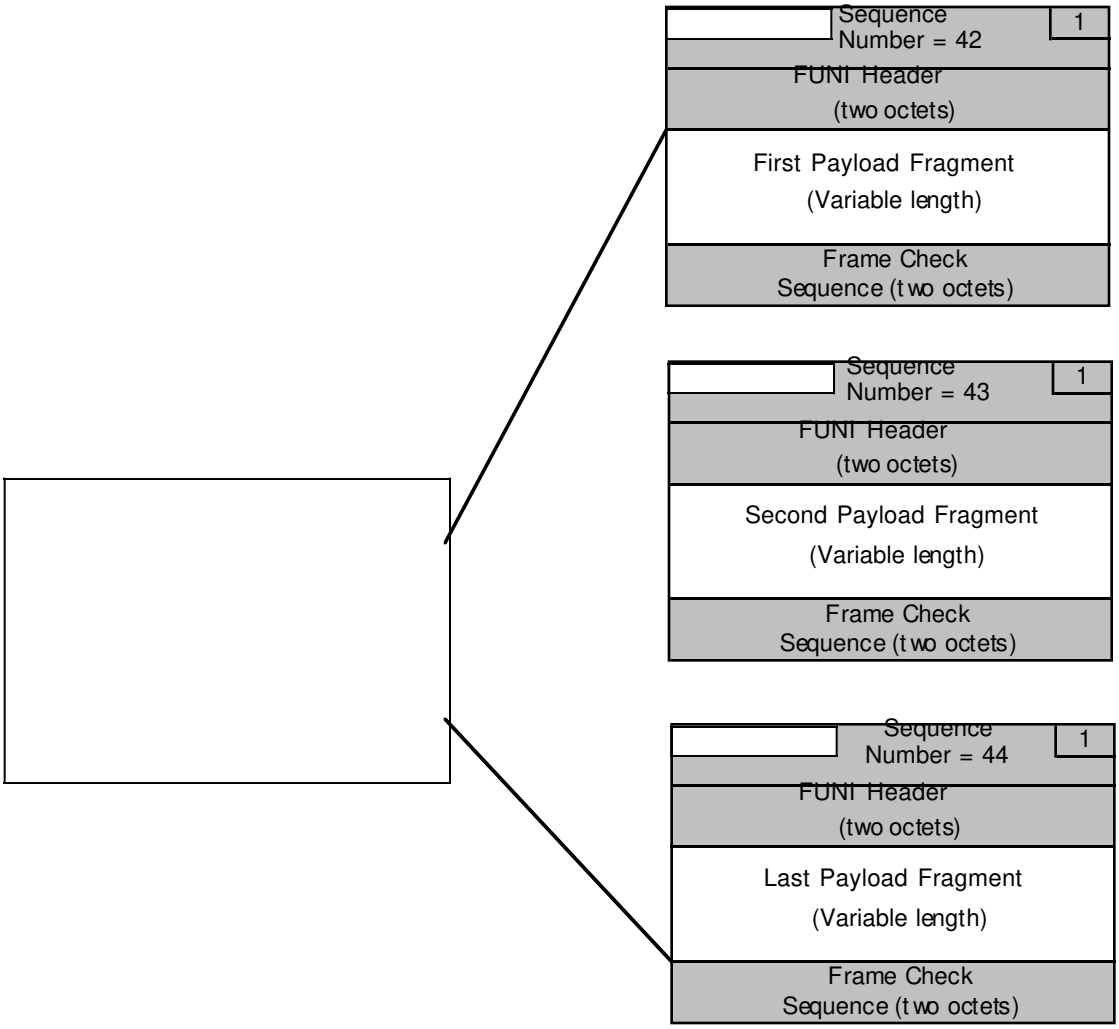

Figure 10-1 An Example of Fragmentation

11. INFORMATIVE ANNEX A – DYNAMIC FRAGMENTATION

This optional annex may be implemented to support voice and other real-time (delay-sensitive) data on lower-speed FUNI links where it is necessary to fragment long data frames that share the same FUNI link with shorter real-time data frames so that the shorter frames are not excessively delayed. Fragmentation thus enables interleaving of delay-sensitive traffic on one VC with fragments of a long frame on another VC utilizing the same interface.

In this specification the first data fragment in a series has the (B)eginning bit of the fragmentation header set, and the final data fragment has the (E)nding bit of the fragmentation header set. Every fragment in the series contains a fragmentation header and a copy of the same FUNI header that was on the original unfragmented frame, including the CN, CLP, and FID bits.

By exploiting the potential of dynamically alterable fragment sizes on the link QoS for real-time data is enhanced and link efficiency is maximized. To implement dynamic fragmentation a special condition for beginning and ending bits has to be met. This allows interruption of large data frames which are part to line for insertion of the small real-time frames. Therefore FDV can be minimized and QoS for the real-time data can be enhanced.

The simplest method of implementation maintains the setting of the (B)eginning bit of the fragmentation header for the primary data fragment. However the fragment payload equals the length of the original unfragmented data frame payload unless a small real-time frame is received while the payload of the data frame is part to line. In this scenario the frame is then interrupted at the next complete octet of payload. A FCS and end flag are then added to this fragment and the remaining part of the frame is interrupted to allow transmission of the real-time frame. The data frame can be further interrupted each time a real-time frame comes to line. To allow this implementation no end bit is set on any valid data fragment; rather a null frame fragment of six octets is required to indicate fragmentation end. This overrides the condition where a data fragment may have both the (B)eginning and (E)nding fragment bits set to 1. Therefore the real-time data is prioritized over the link and maximum efficiency is resumed when there is no real-time data active on a VC.

A frame can not be interrupted in its first five octets (flag and header) or last three octets (flag and FCS). A null fragment with its (E)nding bit set can not be interrupted. All fragmentation must be on an octet-by-octet basis due to the FCS operation. This procedure is bi-directional as is the standard fragmentation procedure covered in the main body of this document, i.e., it can be implemented at either fragmentation entity (DTE or DCE).

The fragmentation header is still present on all frames for control purposes. The low order bit of the first octet of the fragmentation header is set. This allows the fragmentation header to be distinguishable from the FUNI header. Therefore a fragmentation entity (DTE or DCE) can detect a misconfiguration of its peer, since the peers must be configured identically to use or not

use fragmentation across an interface. If a peer is configured for fragmentation and receives frames that do not contain the fragmentation header, those frames are discarded. If a peer is not configured for fragmentation and receives frames with the fragmentation header, those frames will be discarded due to the violation of the FUNI header format.

Therefore the receiver can keep track of the incoming sequence numbers for each VC and maintain the most recently received sequence number. The receiver detects the end of a reassembled frame when it receives the null fragment bearing the (E)nding bit. Reassembly of the frame is complete if all sequence numbers up to that fragment have been received. The detection of lost fragments or other errors are therefore dealt with in the same way as that indicated in the main body of this document.

The maximum fragment size should be configured on a per-interface basis. The optimal fragment size is the result of a tradeoff between the efficiencies of large frames, the interface speed, and the required delay and delay variation characteristics of the applications.

Receivers must be able to reassemble complete frames up to at least 4096 or 9232 octets in length, depending on the FUNI mode in use.