FDL Cell formats and Meta Signalling

Richard Black

14th August, 18th December 1991
24th February 1992, 4th February 1993
9th March 1994

1 Introduction

This document described the FDL (fourty-eight data link) instance of MSDL. There are two variants of FDL. The principal one is the public form used on all point to links. The second variant is a form which is private within a Fairisle Switch and used for distributed connection management. The former will be described in this document with differences noted as applicable.

2 Cell format

The control information is carried in the four octet header of the cell. This consists of two octets of VCI and two octets of FAS. The user payload of the cell carries forty eight octets of data.

2.1 VCI

The VCI is allocated by the receiving entity except in the cases of promiscuous associations. Each entity may be prepared to allocate as many or as few VCIs as it wishes. For this reason promiscuous associations ought to be allocated a VCI which is likely to be available for reception on all entities.

The VCI consists of two octets. In the remainder of this section they shall be referred to as a pair where the first element of the pair is the one that is transmitted first on the network.

VCIs of 0x00,0x00 and 0xff,0xff are reserved for special purposes by entities and may not be used for communication. They shall never be allocated or transmitted on.[1]

2.2 FAS

The FAS, otherwise known as user control information has two purposes. One purpose is to indicate framing information to allow for optimisations at the FDL level for example when blocking cells together or when interrupting a host. There are two bits of framing information. Bit 7 of the second octet is defined to be set on the first cell of a block. Bit 6 of the second octet is defined to be set on the last cell of a block.

The other six bits in the second octet and all of the first octet are defined to be opaque at the FDL level. FDL networks use a variant of SAR1a which is fully described in the "Segmentation and Reassembly" document found elsewhere in this collection. This variant is to permit the temporary interworking of this protocol with newer standard conformant equipment.

[1] On the Fairisle Port Controller VCI 0xff,0xff is needed to determine how deep the switch fabric is. Other (point to point) devices, must *not* therefore transmit on this VCI.

2.3 Cell organisation

The cell header is shown in table 1

Octet	7	6	5	4	3	2	1	0	
0	VCI								
1	VCI								
2	FAS								
3	fs	fe	FAS						frame start and frame end flags

Table 1: FDL Cell Header

3 Meta Signalling

The Meta Signalling varies slightly between the two variants of FDL. Since not all hardware can necessarily distinguish the source of an FDL cell it is strictly forbidden to send a FDL Meta Signalling cell of the incorrect variant.[2]

3.1 VCI and FAS thereof

Meta Signalling or connection management is an example of a promiscuous association. It has therefore, as the first such association, been allocated VCI 0x01,0x00. For the switch variant care should be taken to ensure that the cell is sent to the local management entity.[3]

Meta Signalling cells, since on a promiscuous association, can arrive in any order and are therefore not considered as blocks. They shall therefore be sent with both the frame start and frame end bits set. The sequence field should contain the value 0x01 and the assmeble field many contain any legal value.

3.2 FDL MSC data format

The format adopted for FDL Meta Signalling information is displayed in table 2.

For connection management commands it is important to note that the only fields of this structure which are sent end to end are the OrigAddr, OrigPort, RespAddr, RespPort and appropriate parts of the argument. This structure is *only* sent on networks which use FDL.

3.3 The Fairisle Variant

Inside a fairisle switch the FDL Meta Signalling cell fields "OrigVCI" and "RespVCI" contain a full complement of four octets which are the fabric and port-controller route octets, followed by the two octet VCI as for the point to point variant.

3.4 FDL Checksum

A simple checksum has been implemented on FDL Meta Signalling cells. This is generated on all transmitted cells and checked on received cells. Cells which do not match the checksum are to

[2]eg. the Fairisle Port Controllers cannot distinguish on arrival whether the cell came from the "loop-back" or "receive" fifos. Incorrect variants would therefore cause considerable chaos.

[3]ie. Set the portc loop-back bit

Octet	Field	Description
0	FDLVariant	A descriptor, 1 is point to point, 2 is fairisle
1	Reserved	
2	ProtVersion	Protocol version field, currently 1
3	MSCommand	Connection request reply etc.
4,5	OrigVCI	First and second octets of the originator's VCI
6,7		Reserved should be 0
8,9	RespVCI	First and second octets of the responder's VCI
10,11		Reserved should be 0
12-15	OrigAddr	Four octets of the originator's local address
16-19	OrigPort	Four octets of the originator's port
20-23	RespAddr	Four octets of the responder's local address
24-27	RespPort	Four octets of the responder's port
28-43	Argument	Appropriate argument to the MSCommand eg. QoS
44-47	Checksum	A checksum on the other fields - see section 3.4.

Table 2: FDL Meta Signalling Cell format

be ignored.[4] The checksum is a four octet value stored in the last four octets of the cell. It is generated in eleven stages each manipulating four octets of the cell and the four octet checksum value. The checksum is believed to be the same as that used by the XTP protocol (there is no other similarity with that protocol).

Initialising with zero, each stage exclusive-ors the current value with the next four octets. The value is then rotated so that the most significant bit of octet n becomes the least significant bit of octet $n-1$ except that the least significant bit of octet 0 becomes the most significant bit of octet 3.[5] Finally the value is stored in the last four octets of the cell. Note that the rotation occurs after the last exclusive-or - that is eleven times in total.[6]

3.5 Meta Signalling Commands

The meta signalling commands used on FDL are identical to those used by all other current implementaions of MSDL. They are therefore described in that generic document.

[4] It is of course considered appropriate to generate a diagnostic.

[5] What this is really doing is: take each word in the cell, convert it from network to host endian, xor it with the checksum and rotate the checksum left one bit. The checksum is finally converted back to network endian before it is stored.

[6] See the Wanda file kernel/netmsnl/rjbarp.c for an example implementation