endace
accelerated

ERF Types

EDM11-01

Protection Against Harmful Interference

When present on equipment this manual pertains to, the statement "This device complies with part 15 of the FCC rules" specifies the equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the Federal Communications Commission [FCC] Rules.

These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment.

This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications.

Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at their own expense.

Extra Components and Materials

The product that this manual pertains to may include extra components and materials that are not essential to its basic operation, but are necessary to ensure compliance to the product standards required by the United States Federal Communications Commission, and the European EMC Directive. Modification or removal of these components and/or materials, is liable to cause non compliance to these standards, and in doing so invalidate the user's right to operate this equipment in a Class A industrial environment.

Disclaimer

Whilst every effort has been made to ensure accuracy, neither Endace Technology Limited nor any employee of the company, shall be liable on any ground whatsoever to any party in respect of decisions or actions they may make as a result of using this information.

Endace Technology Limited has taken great effort to verify the accuracy of this manual, but nothing herein should be construed as a warranty and Endace shall not be liable for technical or editorial errors or omissions contained herein.

In accordance with the Endace Technology Limited policy of continuing development, the information contained herein is subject to change without notice.

Published by:

Endace Technology® Ltd
Level 9
85 Alexandra Street

PO Box 19246
Hamilton 3244
New Zealand

Phone: +64 7 839 0540
Fax: +64 7 839 0543
support@endace.com
www.endace.com

International Locations

New Zealand
Endace Technology Ltd
Building 7, Lambie Drive
PO Box 76802
Manukau City 2104
New Zealand
Phone: +64 9 262 7260
Fax: +64 9 262 7261

Americas
Endace Network Systems Inc
14425 Penrose Place
Suite 225
Chantilly, VA 20151
United States of America
Phone: +1 703 964 3740
Fax: +1 703 378 0602

Europe, Middle East & Africa
Endace Europe® Ltd
Sheraton House
Castle Park
Cambridge CB3 0AX
United Kingdom
Phone: +44 1223 370 176
Fax: +44 1223 370 040

Copyright 2005 - 2008 Endace Technology Ltd. All rights reserved.

Endace, the Endace logo, Endace Accelerated, DAG, NinjaBox and NinjaProbe are trademarks or registered trademarks in New Zealand, or other countries, of Endace Technology Limited. Applied Watch and the Applied Watch logo are registered trademarks of Applied Watch Technologies LLC in the USA. All other product or service names are the property of their respective owners. Product and company names used are for identification purposes only and such use does not imply any agreement between Endace and any named company, or any sponsorship or endorsement by any named company.

Use of the Endace products described in this document is subject to the Endace Terms of Trade and the Endace End User License Agreement (EULA).

Contents

Introduction 1

Overview 1
Support 1

Extensible Record Format 3

Introduction 3
DAG Card Extensible Record Format Types 3
ERF Types for each DAG card 4
Generic ERF Header 6
ERF 1. TYPE_POS_HDLC 8
ERF 2. TYPE_ETH 9
ERF 3. TYPE_ATM 10
ERF 4. TYPE_AAL5 11
ERF 5. TYPE_MC_HDLC 12
ERF 6. TYPE_MC_RAW 13
ERF 7. TYPE_MC_ATM 14
ERF 8. TYPE_MC_RAW_CHANNEL 15
ERF 9. TYPE_MC_AAL5 16
ERF 10. TYPE_COLOR_HDLC_POS 17
ERF 11. TYPE_COLOR_ETH 18
ERF 12. TYPE_MC_AAL2 19
ERF 13. TYPE_IP_COUNTER 20
ERF 14. TYPE_TCP_FLOW_COUNTER 21
ERF 15. TYPE_DSM_COLOR_HDLC_POS 22
ERF 16. TYPE_DSM_COLOR_ETH 23
ERF 17. TYPE_MC_HDLC_POS 24
ERF 18. TYPE_AAL2 25
ERF 19. TYPE_COLOR_HASH_POS 26
ERF 20. TYPE_COLOR_HASH_ETH 27
ERF 21. TYPE_INFINIBAND 28
ERF 22. TYPE_IPV4 29
ERF 23. TYPE_IPV6 30
ERF 24. TYPE_RAW_LINK 31
ERF 48. TYPE_PAD 32
Extensible Record Format Timestamps 33
Overview 33
DAG card resolutions 33
Example code 33

Extension Headers (EH) 34

Introduction 34
Extension Headers Types 35
EH 3. Classification 36
EH 4. Intercept ID 37
EH 5. Raw_Link 38

Version History 39

Introduction

Overview

This document identifies and explains:

- the Endace Extensible Record Format (ERF), see Extensible Record Format (ERF) (page 3) and
- the Extension Headers, see Extension Headers (EH) (page 34).

Support

If any problems are encountered with Endace hardware, firmware or supplied software, contact Endace Technical Support via the email address support@endace.com.

Supplying detailed information about a problem enables a more concise first-response.

Extensible Record Format

Introduction

Endace DAG monitoring interface cards produce trace files in their own native format, known as the Extensible Record Format (ERF). The ERF file contains of a series of records. Each record describes one packet.

An ERF file consists only of ERF records; there is no special file header. This allows concatenation and splitting to be performed arbitrarily on ERF record boundaries.

DAG Card Extensible Record Format Types

The Endace DAG cards produce extensible record format types that include:

Number	Type	Description
0:	TYPE_LEGACY	Old style record
1:	TYPE_HDLC_POS	Packet over SONET / SDH frames, using either PPP or CISCO HDLC framing.
2:	TYPE_ETH	Ethernet
3:	TYPE_ATM	ATM cell
4:	TYPE_AAL5	reassembled AAL5 frame
5:	TYPE_MC_HDLC	Multi-channel HDLC frame
6:	TYPE_MC_RAW	Multi-channel Raw time slot link data
7:	TYPE_MC_ATM	Multi-channel ATM Cell
8:	TYPE_MC_RAW_CHANNEL	Multi-channel Raw link data
9:	TYPE_MC_AAL5	Multi-channel AAL5 frame
10:	TYPE_COLOR_HDLC_POS	HDLC format like TYPE_HDLC_POS, but with the LCNTR field reassigned as COLOR
11:	TYPE_COLOR_ETH	Ethernet format like TYPE_ETH, but with the LCNTR field reassigned as COLOR
12:	TYPE_MC_AAL2	Multi-channel AAL2 frame
13:	TYPE_IP_COUNTER	IP Counter ERF Record
14:	TYPE_TCP_FLOW_COUNTER	TCP Flow Counter ERF Record
15:	TYPE_DSM_COLOR_HDLC_POS	HDLC format like TYPE_HDLC_POS, but with the LCNTR field reassigned as DSM COLOR
16:	TYPE_DSM_COLOR_ETH	Ethernet format like TYPE_ETH, but with the LCNTR field reassigned as DSM COLOR
17:	TYPE_COLOR_MC_HDLC_POS	Multi-channel HDLC like TYPE_MC_HDLC, but with the LCNTR field reassigned as COLOUR
18:	TYPE_AAL2	Reassembled AAL2 Frame Record
19:	TYPE_COLOR_HASH_POS	Colored PoS HDLC record with Hash load balancing
20:	TYPE_COLOR_HASH_ETH	Colored Ethernet variable length record with Hash load balancing
21:	TYPE_INFINIBAND	Infiniband Variable Length Record
22:	TYPE_IPV4	IPV4 Variable Length Record
23:	TYPE_IPV6	IPV6 Variable Length Record
24	TYPE_RAW_LINK	Raw link data, typically SONET or SDH Frame
32-47:	-	Reserved for CoProcess Development Kit (CDK) Users and Internal use
48:	TYPE_PAD	Pad Record type

ERF Types for each DAG card

The Extensible Record Format (ERF) types used by each DAG card is listed below.

Card	Type	Extensible Record Format Type
DAG 3.7D	Type 1 Type 3	PoS HDLC Record ATM Cell Record
DAG 3.7GP/GF	Type 2	Ethernet Record
DAG 3.7T	Type 4 Type 5 Type 6 Type 7 Type 8 Type 9 Type 12	Reassembled AAL5 Frame Record Multi-channel HDLC Frame Record Multi-channel RAW Time Slot Link Data Record Multi-channel ATM Cell Record Multi-channel RAW Channel: Multi-channel RAW Link Data Multi-channel AAL5: Multi-channel AAL5 Frame Multi-channel AAL25: Multi-channel AAL2 Frame
DAG 3.8S	Type 1 Type 3 Type 4 Type 10	PoS HDLC Record ATM Cell Record Reassembled AAL5 Frame Record* Colored PoS HDLC Record*
DAG 4.3GE	Type 2 Type 11	Ethernet Record Colored Ethernet Record*
DAG 4.3S	Type 1 Type 3 Type 4 Type 10	PoS HDLC Record ATM Cell Record Reassembled AAL5 Frame Record* Colored PoS HDLC Record*
DAG 4.5G2/G4/	Type 2 Type 16	Ethernet Record DSM Color Ethernet record
DAG 5.0SG2	Type 1 Type 2 Type 10 Type 11 Type 15 Type 16 Type 19 Type 20	PoS HDLC Record Ethernet Record Colored PoS HDLC Record* Colored Ethernet Record* DSM Color HDLC PoS Record DSM Color Ethernet Record Colored PoS HDLC Record with Hash Load Balancing Colored Ethernet Record with Hash Load Balancing
DAG 5.0SG2A	Type 1 Type 2 Type 10 Type 11 Type 15 Type 16 Type 19 Type 20	PoS HDLC Record Ethernet Record Colored PoS HDLC Record* Colored Ethernet Record* DSM Color HDLC PoS Record DSM Color Ethernet Record Colored PoS HDLC Record with Hash Load Balancing Colored Ethernet Record with Hash Load Balancing
DAG 5.2SXA	Type 1 Type 2 Type 10 Type 11 Type 15 Type 16 Type 19 Type 20	PoS HDLC Record Ethernet Record Colored PoS HDLC Record* Colored Ethernet Record* DSM Color HDLC PoS Record DSM Color Ethernet Record Colored PoS HDLC Record with Hash Load Balancing Colored Ethernet Record with Hash Load Balancing
DAG 5.2X	Type 2 Type 16	Ethernet Record DSM Color Ethernet Record

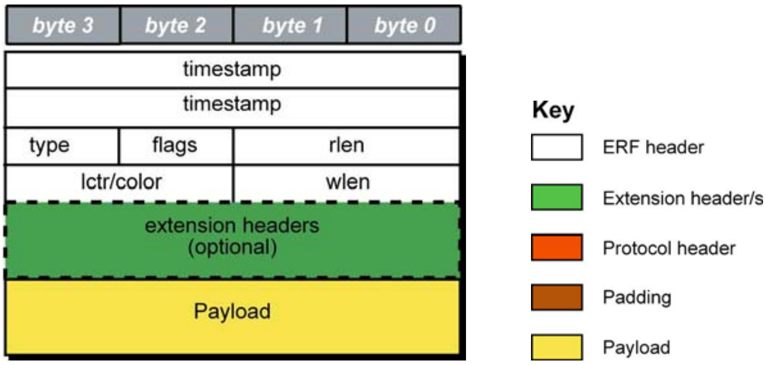
DAG 5.4S-12 DAG 5.4SG-48	Type 1 Type 2 Type 10 Type 11 Type 19 Type 20	PoS HDLC Record Ethernet Record Colored PoS HDLC Record* Colored Ethernet Record* Colored PoS HDLC Record with Hash Load Balancing Colored Ethernet Record with Hash Load Balancing
DAG 5.4GA DAG 5.4SA-12 DAG 5.4SGA-48	Type 1 Type 2 Type 10 Type 11 Type 19 Type 20	PoS HDLC Record Ethernet Record Colored PoS HDLC Record* Colored Ethernet Record* Colored PoS HDLC Record with Hash Load Balancing Colored Ethernet Record with Hash Load Balancing
DAG 6.1SE	Type 1 Type 2	PoS HDLC Record Ethernet Record
DAG 6.2SE	Type 1 Type 2 Type 15 Type 16	PoS HDLC Record Ethernet Record DSM Color HDLC PoS Record DSM Color Ethernet Record
DAG 7.1S	Type 1 Type 3 Type 4 Type 5 Type 6 Type 7 Type 9 Type 12 Type 18	PoS HDLC Record ATM Cell Record Reassembled AAL5 Frame Record Multi-channel HDLC Frame Record Multi-channel RAW Time Slot Link Data Record Multi-channel ATM Cell Record Multi-channel AAL5: Multi-channel AAL5 Frame Multi-channel AAL25: Multi-channel AAL2 Frame Reassembled AAL2 Frame Record
DAG 8.1SX	Type 1 Type 2	PoS HDLC Record Ethernet Record
DAG 8.1X	Type 2	Ethernet Record
DAG 8.2X	Type 2 Type 16	Ethernet Record DSM Color Ethernet record
DAG 8.4I	Type 21	Infiniband

* Requires Endace Coprocessor and appropriate Firmware.

Generic ERF Header

All ERF records share some common fields. Timestamps are in little-endian (Pentium® native) byte order. All other fields are in big-endian (network) byte order. All payload data is captured as a byte stream in network order, no byte or re-ordering is applied.

The generic ERF header is shown below:



The fields are described below:

timestamp		The time of arrival of the cell, an ERF 64-bit timestamp.
type	Bit 7	Extension header present.
	Bit 6:0	Extension header type. See table below:
flags		This byte is divided into several fields as follows:

Bits	Description
1-0:	Binary enumeration of capture interface: 11 Interface 3 or D 10 Interface 2 or C 01 Interface 1 or B 00 Interface 0 or A Cards with more than four interfaces typically use Multichannel ERF types (type 5 to 9, 12 and 17) which provide a separate larger interface field.
2:	Varying length record. When set, packets shorter than the snap length are not padded and rlen resembles wlen. When clear, longer packets are snapped off at snap length and shorter packets are padded up to the snap length. rlen resembles snap length. Setting novarlen and slen greater than 256 bytes is wasteful of bandwidth
3:	Truncated record - insufficient buffer space. • `wlen` is still correct for the packet on the wire. • `rlen` is still correct for the resulting record. But, rlen is shorter than expected from snap length or wlen values. **Note:** truncation is depreciated and this bit is unlikely to be set in an ERF record.
4:	RX error. An error in the received data. Present on the wire
5:	DS error. An internal error generated inside the card annotator. Not present on the wire.
6:	Reserved
7:	Reserved

rlen		Record length in bytes. Total length of the record transferred over the PCI bus to storage. The timestamp of the next ERF record starts exactly rlen bytes after the start of the timestamp of the current ERF record.
lctr		Depending upon the ERF type this 16 bit field is either a loss counter of color field. The loss counter records the number of packets lost between the DAG card and the stream buffer due to overloading on the PCI bus. The loss is recorded between the current record and the previous record captured on the same stream/interface. The color field is explained under the appropriate type details.

wlen		Wire length. Packet length "on the wire" including some protocol overhead. The exact interpretation of this quantity depends on physical medium. This may contain padding.
extension headers		Extension headers in an ERF record allow extra data relating to each packet to be transported to the host. Extension header/s are present if bit 7 of the type field is '1'. If bit 7 is '0', no extension headers are present (ensures backwards compatibility). **Note:** There can be more than one Extension header attached to a ERF record.
Payload		Payload is the actual data in the record. It can be calculated by either : • Payload = rlen - ERF header - Extension headers (optional) - Protocol header - Padding

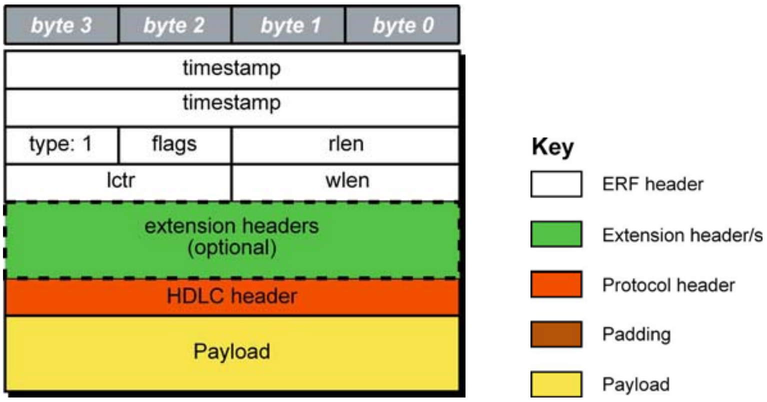
Extension header types

Number	Type	Description
0:	TYPE_LEGACY	Old style record
1:	TYPE_HDLC_POS	Packet over SONET / SDH frames, using either PPP or CISCO HDLC framing.
2:	TYPE_ETH	Ethernet
3:	TYPE_ATM	ATM cell
4:	TYPE_AAL5	reassembled AAL5 frame
5:	TYPE_MC_HDLC	Multi-channel HDLC frame
6:	TYPE_MC_RAW	Multi-channel Raw time slot link data
7:	TYPE_MC_ATM	Multi-channel ATM Cell
8:	TYPE_MC_RAW_ CHANNEL	Multi-channel Raw link data
9:	TYPE_MC_AAL5	Multi-channel AAL5 frame
10:	TYPE_COLOR_HDLC_ POS	HDLC format like TYPE_HDLC_POS, but with the LCNTR field reassigned as COLOR
11:	TYPE_COLOR_ETH	Ethernet format like TYPE_ETH, but with the LCNTR field reassigned as COLOR
12:	TYPE_MC_AAL2	Multi-channel AAL2 frame
13:	TYPE_IP_COUNTER	IP Counter ERF Record
14:	TYPE_TCP_FLOW_ COUNTER	TCP Flow Counter ERF Record
15:	TYPE_DSM_COLOR_ HDLC_POS	HDLC format like TYPE_HDLC_POS, but with the LCNTR field reassigned as DSM COLOR
16:	TYPE_DSM_COLOR_ ETH	Ethernet format like TYPE_ETH, but with the LCNTR field reassigned as DSM COLOR
17:	TYPE_COLOR_MC_ HDLC_POS	Multi-channel HDLC like TYPE_MC_HDLC, but with the LCNTR field reassigned as COLOUR
18:	TYPE_AAL2	Reassembled AAL2 Frame Record
19:	TYPE_COLOR_HASH_ POS	Colored PoS HDLC record with Hash load balancing
20:	TYPE_COLOR_HASH_ ETH	Colored Ethernet variable length record with Hash load balancing
21:	TYPE_INFINIBAND	Infiniband Variable Length Record
22:	TYPE_IPV4	IPV4 Variable Length Record
23:	TYPE_IPV6	IPV6 Variable Length Record
24	TYPE_RAW_LINK	Raw link data, typically SONET or SDH Frame
32-47:	-	Reserved for CoProcess Development Kit (CDK) Users and Internal use
48:	TYPE_PAD	Pad Record type

ERF 1. TYPE_POS_HDLC

Type	Bit 7	1 = Extension header present. See Extension Headers (page 34).
	Bits 6:0	Type 1
Short description	TYPE_POS_HDLC	
Long description	Type 1 PoS HDLC Record	
Use	This record format is for HDLC data links. For example: • Packet over SONET • Point-to-Point Protocol [PPP] over SONET • Frame Relay • MTP2 (SS7)	

The TYPE_POS HDLC record is shown below:



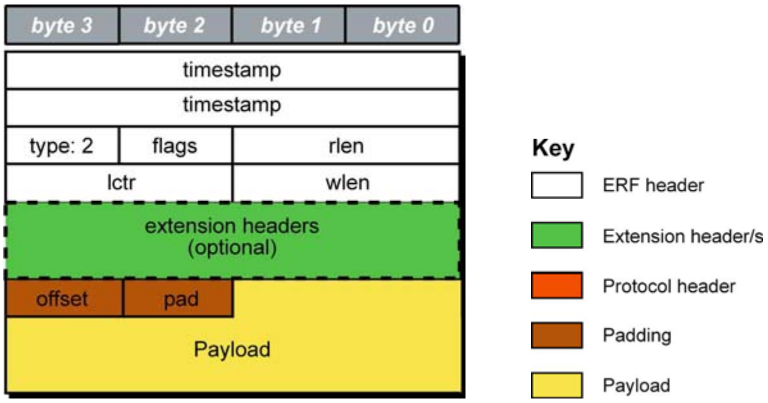
The following is a description of the TYPE_POS_HDLC record format:

Field	Description
HDLC Header (4 bytes)	Protocol Header. Length may vary depending on protocol, typically 4 bytes.
Payload (bytes of record)	Payload = rlen - ERF header (16 bytes) - Extension headers (optional) - Protocol header (4 bytes)

ERF 2. TYPE_ETH

Type	Bit 7	1 = Extension header present. See Extension Headers (page 34).
	Bits 6:0	Type 2
Short description	TYPE_ETH	
Long description	Type 2 Ethernet Record	
Use	This record format is for Ethernet [802.3] data links.	

The TYPE_ETH record is shown below:



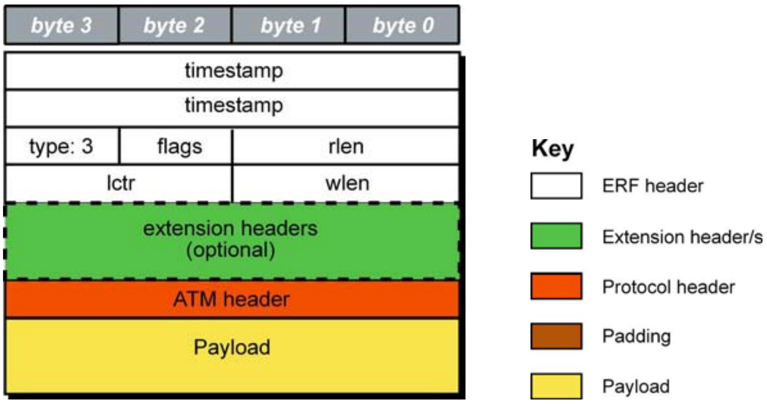
The following is a description of the TYPE_ETH record format:

Field	Description
Offset (1 byte)	Number of bytes **not** captured from start of frame. Typically used to skip link layer headers when not required in order to save bandwidth and space. **Note: This field is currently not implemented, contents should be disregarded.**
Pad (1 byte)	The Ethernet frame begins immediately after the pad byte so that the layer 3 [IP] header is 32-bit aligned. This is typically used to skip link layer headers when they are not required in order to save bandwidth and space.
Payload (bytes of record)	Payload = rlen - ERF header (16 bytes) - Extension headers (optional) - Padding (2 bytes)

ERF 3. TYPE_ATM

Type	Bit 7	1 = Extension header present. See Extension Headers (page 34).
	Bits 6:0	Type 3
Short description	TYPE_ATM	
Long description	Type 3 ATM Cell Record	
Use	This record format is for ATM cell capture.	

The TYPE_ATM record is shown below:



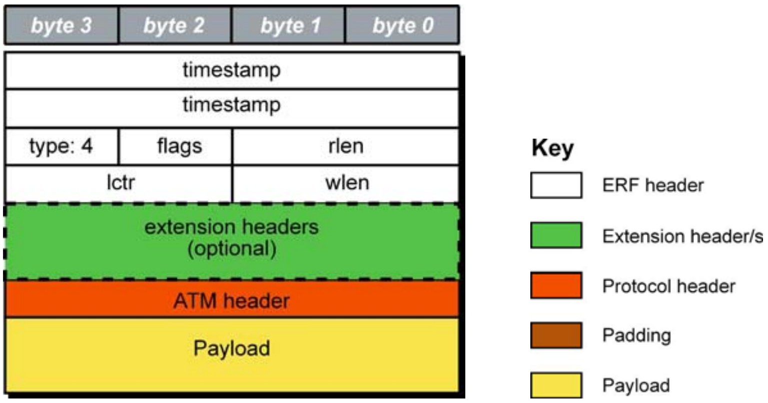
The following is a description of the TYPE_ATM record format:

Field	Description
ATM Header (4 bytes)	Protocol header. Does not include the 8-bit HEC.
Flags (1 byte)	ATM cells should not have the variable length flag set.
Payload (bytes of cell)	Payload = 48 bytes of cell + HEC (1 byte)

ERF 4. TYPE_AAL5

Type	Bit 7	1 = Extension header present. See Extension Headers (page 34).
	Bits 6:0	Type
Short description	TYPE_AAL5	
Long description	Type 4 Reassembled AAL5 Frame Record	
Use	This record format is for reassembled ATM AAL5 frames.	

The TYPE_AAL5 record is shown below:



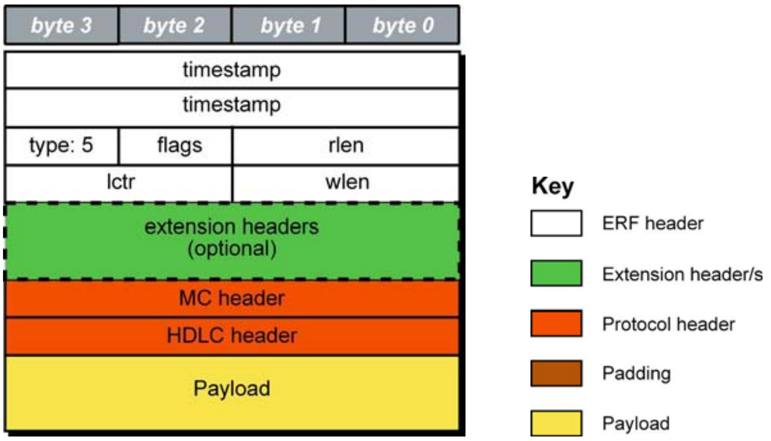
The following is a description of the TYPE_AAL5 record format:

Field	Description
ATM header (4 bytes)	Protocol header of first cell in the frame not including the 8-bit HEC, all other cells in fame must have identical headers so are not included.
Payload (4 bytes)	Payload contains all cells in the frame: • trailing padding (0 - 47 bytes) • 1 byte `cpcs-un` field • 1 byte `cpi` field • 2 byte `length` filed, and • 4 byte `crc` field
Flags (1 byte)	The rx error flag in the ERF haders is set should the AAL5 crc fail.
Payload (bytes of AAL5 frame)	Payload = rlen - ERF header (16 bytes) - Extension headers (optional) - Protocol header (4 bytes)

ERF 5. TYPE_MC_HDLC

Type	Bit 7	1 = Extension header present. See Extension Headers (page 34).
	Bits 6:0	Type 5
Short description	TYPE_MC_HDLC	
Long description	Type 5 Multi-channel HDLC Frame Record	
Use	This record format is for channelized HDLC data links. For example E1, T1 and J1.	

The TYPE_MC_HDLC record is shown below:



The following is a description of the TYPE_MC_HDLC record format:

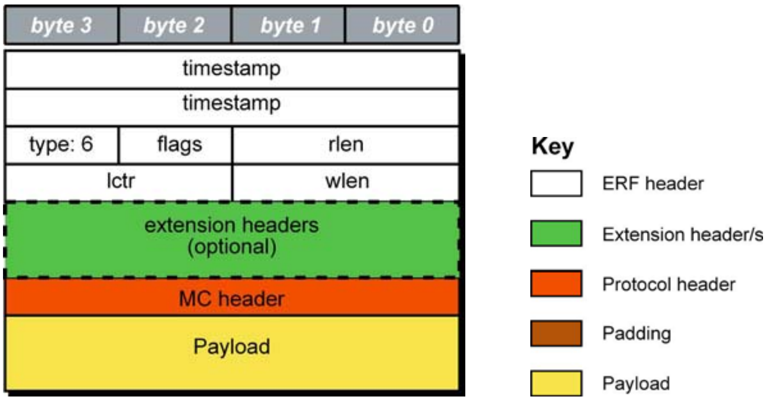
<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>flags
(1 byte)</td><td>This field is the same as normal ERF types but capture interface is always zero.
• Fixed length mode not supported.
• RX Error is set if any MC Header Error bit is set.</td></tr>
<tr><td>MC header
(4 bytes)</td><td>Protocol Header. This field is divided into the following:
<table>
<tr><th>Bits</th><th>Attribute</th></tr>
<tr><td>0-9</td><td>Connection Number [0-1023].</td></tr>
<tr><td>10-15</td><td>Reserved.</td></tr>
<tr><td>16-23</td><td>Reserved.</td></tr>
<tr><td>24</td><td>FCS Error.</td></tr>
<tr><td>25</td><td>Short Record Error [<5 Bytes].</td></tr>
<tr><td>26</td><td>Long Record Error [>2047 Bytes].</td></tr>
<tr><td>27</td><td>Aborted Frame Error.</td></tr>
<tr><td>28</td><td>Octet Error. The closing flag was not octet aligned after bit stuffing.</td></tr>
<tr><td>29</td><td>Lost Byte Error. The internal data path had an unrecoverable error.</td></tr>
<tr><td>30</td><td>1^{ST} Rec. This is the first record received since this connection was configured.</td></tr>
<tr><td>31</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>HDLC header
(4 bytes)</td><td>Protocol header. Length may vary depending on protocol.</td></tr>
<tr><td>Payload
(bytes of packet)</td><td>Payload = rlen - ERF header (16 bytes) - Extension headers (optional) - Protocol header (8 bytes)</td></tr>
</table>

Note: When using this record type with the DAG 3.7T card the Interface number is 0, and the connection number is defined by the programmed context.
When using this record type with the DAG 7.1S card the interface number is used for the four ports, and the connection number is the VC identifier, as defined in the *EDM01-17 DAG 7.1S Card User Guide*.

ERF 6. TYPE_MC_RAW

Type	Bit 7	1 = Extension header present. See Extension Headers (page 34).
	Bits 6:0	Type 6
Short description	TYPE_MC_RAW	
Long description	Type 6 Multi-Channel RAW Time Slot Link Data Record	
Use	This record format is for the RAW capture from data links. For example; E1, T1 and J1.	

The TYPE_MC_RAW record is below:



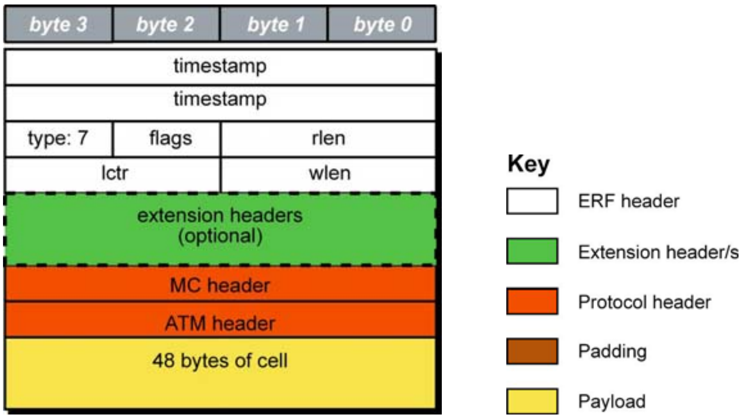
The following is a description of the TYPE_MC_RAW record format:

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>Flags
(1 byte)</td><td>This field is the same as normal ERF types but capture interface is always zero.
• Fixed length mode not supported.
• RX Error is set if any MC Header Error bit is set.</td></tr>
<tr><td>MC header
(4 bytes)</td><td>Protocol header. This field is divided into the following:
<table>
<tr><th>Bits</th><th>Attribute</th></tr>
<tr><td>0-3:</td><td>Physical Interface [0-15].</td></tr>
<tr><td>4-15:</td><td>Reserved.</td></tr>
<tr><td>16-23:</td><td>Reserved.</td></tr>
<tr><td>24:</td><td>Reserved.</td></tr>
<tr><td>25:</td><td>Short Record [<6 Bytes].</td></tr>
<tr><td>26:</td><td>Long Record [>2047 Bytes]</td></tr>
<tr><td>27:</td><td>Reserved.</td></tr>
<tr><td>28:</td><td>Reserved.</td></tr>
<tr><td>29:</td><td>Lost Byte. The internal datapath had an unrecoverable error.</td></tr>
<tr><td>30:</td><td>1st Rec. This is the first record received since this connection was configured.</td></tr>
<tr><td>31:</td><td>Reserved.</td></tr>
</table></td></tr>
<tr><td>Payload
(bytes of raw link data)</td><td>Payload = rlen - ERF header (16 bytes) - Extension headers (optional) - Protocol header (4 bytes)
This field is divided into the following:
<table>
<tr><th>Data type</th><th>Description</th></tr>
<tr><td>T1:</td><td>24 bytes for 24 time slots.</td></tr>
<tr><td>E1:</td><td>31 bytes for time slots 0-31. Slot 16 is signaling information.</td></tr>
<tr><td>Framed E1:</td><td>30 bytes of data for time slots 1-31, slot 0 used for framing is not captured. Slot 16 is signaling information.</td></tr>
</table></td></tr>
</table>

ERF 7. TYPE_MC_ATM

Type	Bit 7	1 = Extension header present. See Extension Headers (page 34).
	Bits 6:0	Type 7
Short description	TYPE_MC_ATM	
Long description	Type 7 Multi-channel ATM Cell Record	
Use	This record format is for ATM cells on channelized data links. For example; E1, T1 and J1.	

The TYPE_MC_ATM record is shown below:



The following is a description of the TYPE_MC_ATM record format:

Field	Description
flags (1 byte)	This field is the same as normal ERF types but capture interface is always zero. • Fixed length mode not supported. • RX Error is set if any MC Header Error bit is set.
MC header (4 bytes)	Protocol header. This field is divided into the following:
ATM header (4 bytes)	Protocol header. The ATM HEC channel is not captured. This record has a fixed length of 72 bytes. This does not include the 8-bit HEC.
Payload (bytes of cell)	Payload = 48 bytes of cell - HEC (1 byte)

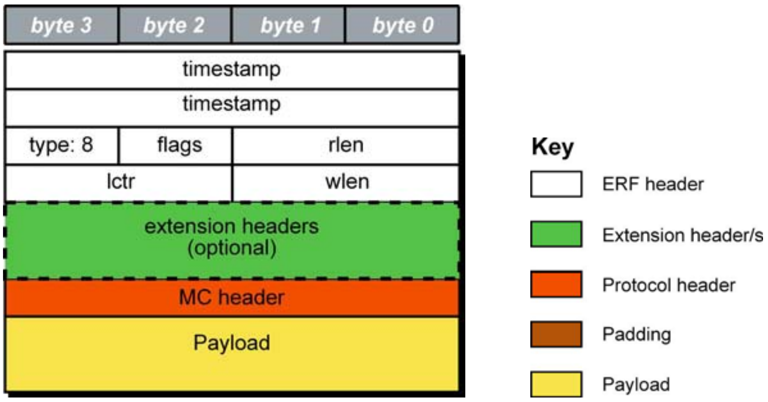
MC header (4 bytes):

Bit	Description
0-9:	Connection number (0-1023). 512 connections are supported by DAG 3.7T card. For the DAG 7.1S card refer to *EDM01-17 DAG 7.1S Card User Guide* for details. Refer to the Channelized Configuration > Configuration File.
10-14:	Reserved.
15:	Multiplexed from IMA group into ATM stream. When bit 15 of the MC Header is set the bottom 9 bits (Connection Number/IMA ID) shall be treated as an IMA Group ID instead of a connection number.
16-19:	Physical port [0-15] cell was captured on. Physical ID is interpreted from the firmware perspective. For example, if a cable is plugged into port 0, examining the ERF MC Header field will give a Physical ID of 11. This is a little counter-intuitive and reflects the internal processing required. From the software/user perspective, this could be interpreted as the Logical ID, and as such, we can convert from the Logical to Physical ID using the provided `dagutil` function, `dagutil_37t_line_get_logical` which will return the Software Physical ID/Firmware Logical ID. In this case, assuming data is coming in on a cable plugged into port 0, we will convert 11 back to 0.
20-23:	Reserved.
24:	Lost Byte. The internal datapath had an unrecoverable error.
25:	HEC corrected.
26:	OAM Cell CRC-10 Error [not implemented].
27:	OAM Cell.
28:	1st Cell. This is the first cell received since this connection was configured.
29-31:	Reserved.

ERF 8. TYPE_MC_RAW_CHANNEL

Type	Bit 7	1 = Extension header present. See Extension Headers (page 34).
	Bits 6:0	Type 8
Short description	TYPE_MC_RAW_CHANNEL	
Long description	Type 8 Multi-channel RAW Channel Multi-channel RAW Link Data Record	
Use	This record format captures complete RAW channelized data links. For example, E1, T1 and J1.	

The TYPE_MC_RAW_CHANNEL record is shown below:



The following is a description of the TYPE_MC_RAW_CHANNEL record format:

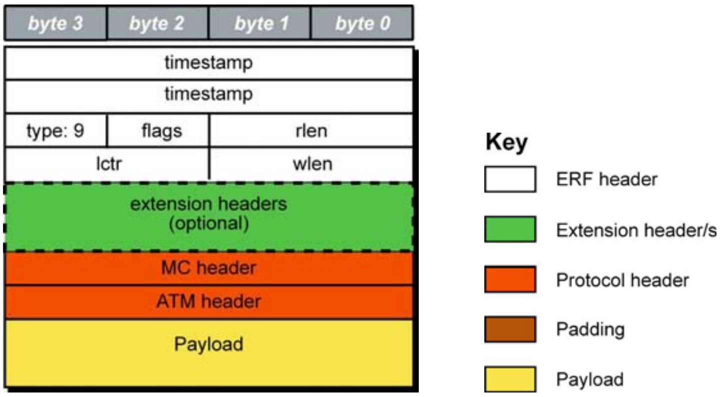
Field	Description
flags (1 byte)	This field is the same as normal ERF types but capture interface is always zero. • Fixed length mode not supported. • RX Error is set if any MC Header Error bit is set.
MC header (4 bytes)	Protocol header. This field is divided into the following: **Bits** / **Attributes** 0-9: Connection number (0-1023). 10-28: Reserved. 29: Lost Byte Error. The internal datapath had an unrecoverable error. 30: 1st Rec. This is the first record received since this connection was configured. 31: Reserved.
Payload (bytes of data)	Payload = rlen - ERF header (16 bytes) - Extension headers (optional) - Protocol header (4 bytes)

Note: When using this record type with the DAG 3.7T card the Interface number is 0, and the connection number is defined by the programmed context.
When using this record type with the DAG 7.1S card the interface number is used for the four ports, and the connection number is the VC identifier, as defined in the DAG 7.1S Card User Guide.

ERF 9. TYPE_MC_AAL5

Type	Bit 7	1 = Extension header present. See Extension Headers (page 34).
	Bits 6:0	Type 9
Short description	TYPE_MC_AAL5	
Long description	Type 9 Multi-channel AAL5: Multi-channel AAL5 Frame Record	
Use	This record format for reassembled ATM AAL5 frames from channelized data links. For example; E1, T1, J1.	

The TYPE_MC_AAL5 record is shown below:



The following is a description of the TYPE_MC_AAL5 record format:

Field	Description
flags (1 byte)	This field is the same as normal ERF types but capture interface is always zero. • Fixed length mode not supported. • RX Error is set if any MC. Header Error bit is set.
wlen (2 bytes)	This contains the length of the AAL5 frame including the ATM Header but not including the ERF Header. The ERF record will always be 64 bit aligned, if the AAL5 frame is not 64 bit aligned the record will be padded at the end of the record with the value 0x00. This padding will not be included in the `wlen` count.
MC header (4 bytes)	Protocol Header. This field is divided into the following:

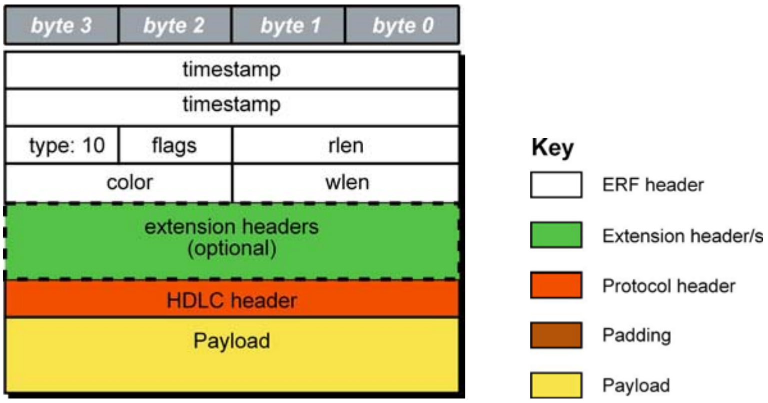
Bits	Attributes
0-10:	Connection number (0-2047). 512 connections are supported by DAG 3.7T card.
11-15:	Reserved.
16-19:	Physical port (0-15) cell was captured on. Physical ID is interpreted from the firmware perspective. For example, if a cable is plugged into port 0, examining the ERF MC Header field will give a Physical ID of 11. This is a little counter-intuitive and reflects the internal processing required. From the software/user perspective, this could be interpreted as the Logical ID, and as such, we can convert from the Logical to Physical ID using the provided `dagutil` function, `dagutil_37t_line_get_logical` which will return the Software Physical ID/Firmware Logical ID. In this case, assuming data is coming in on a cable plugged into port 0, we will convert 11 back to 0. For the 7.1S this field is always 0.
20:	CRC checked.
21:	CRC error.
22:	Length checked.
23:	Length error.
24-27:	Reserved.
28:	1st Cell. This is the first cell received since this connection was configured.
29-31:	Reserved.

Field	Description
ATM header (4 bytes)	Protocol Header. This does not include the 8-bit HEC.
Payload (bytes of AAL5 frame)	Payload = rlen - ERF header (16 bytes) - Extension headers (optional) - Protocol header (8 bytes)

ERF 10. TYPE_COLOR_HDLC_POS

Type	Bit 7	1 = Extension header present. See Extension Headers (page 34).
	Bits 6:0	Type 10
Short description	TYPE_COLOR_HDLC_POS	
Long description	Type 10 Colored PoS HDLC Record	
Use	This record format is for data links, incorporating filter results. The record format is the same type as the Type 1 POS_HDLC (page 8) record, with the exception that the *lctr* field is reassigned as *color*. Requires Endace Coprocessor and appropriate firmware.	

The TYPE_COLOR_HDLC_POS record is shown below:



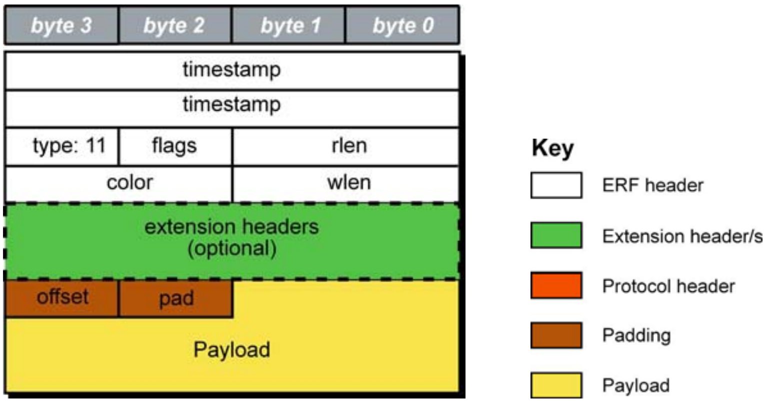
The following is a description of the TYPE_COLOR_HDLC_POS record format:

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>color
(2 bytes)</td><td>The color field is a hardware generated tag indicating the result of a filtering or classification operation.
This field is divided into the following:
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0:</td><td>Set if the record should have been sent to receive stream 0.</td></tr>
<tr><td>1:</td><td>Set if the record should have been sent to receive stream 2.</td></tr>
<tr><td>2-15:</td><td>A 14-bit unsigned integer that corresponds to the filter rule this packet matched.</td></tr>
</table></td></tr>
<tr><td>HDLC header
(4 bytes)</td><td>Protocol header. Length may vary depending on protocol.</td></tr>
<tr><td>Payload
(bytes of record)</td><td>Payload = rlen - ERF header (16 bytes) - Extension headers (optional) - Protocol header (4 bytes)</td></tr>
</table>

ERF 11. TYPE_COLOR_ETH

Type	Bit 7	1 = Extension header present. See Extension Headers (page 34).
	Bits 6:0	Type 11
Short description	TYPE_COLOR_ETH	
Long description	Type 11 Colored Ethernet Record	
Use	This record format is for the Ethernet links [802.3], incorporating filter results. The record format is the same type as the Type 2 TYPE_ETH (page 9) record, with the exception that the *lctr* field is reassigned as *color*. Requires Endace Coprocessor and appropriate firmware.	

The TYPE_COLOR_ETH variable length record is shown below:



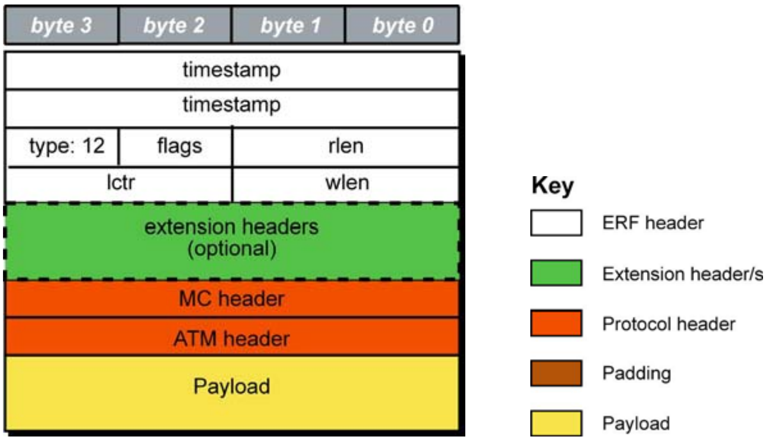
The following is a description of the TYPE_COLOR_ETH record format:

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>color
(2 bytes)</td><td>The color field is a hardware generated tag indicating the result of a filtering or classification operation.
This field is divided into the following:
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0:</td><td>Set if the record should have been sent to receive stream 0.</td></tr>
<tr><td>1:</td><td>Set if the record should have been sent to receive stream 2.</td></tr>
<tr><td>2-15:</td><td>A 14-bit unsigned integer that corresponds to the filter rule this packet matched.</td></tr>
</table></td></tr>
<tr><td>offset
(1 byte)</td><td>Number of bytes <b>not</b> captured from the start of the frame. This is typically used to skip link layer headers when they are not required in order to save bandwidth and space.
<b>Note: This field is currently not implemented; contents should be disregarded.</b></td></tr>
<tr><td>Pad
(1 byte)</td><td>The Ethernet frame begins immediately after the pad byte so that the layer 3 [IP] header is 32-bit aligned. This is typically used to skip link layer headers when they are not required in order to save bandwidth and space.</td></tr>
<tr><td>Payload
(bytes of record)</td><td>Payload = rlen - ERF header (16 bytes) - Extension headers (optional)
- Padding (2 bytes)</td></tr>
</table>

ERF 12. TYPE_MC_AAL2

Type	Bit 7	1 = Extension header present. See Extension Headers (page 34).
	Bits 6:0	Type 12
Short description	TYPE_MC_AAL2	
Long description	Type 12 Multi-channel AAL25: Multi-channel AAL2 Frame Record	
Use	This record format is for channelized links is the same as the normal ERF Types but capture interface is always zero.	

The TYPE_MC_AAL2 record is shown below:



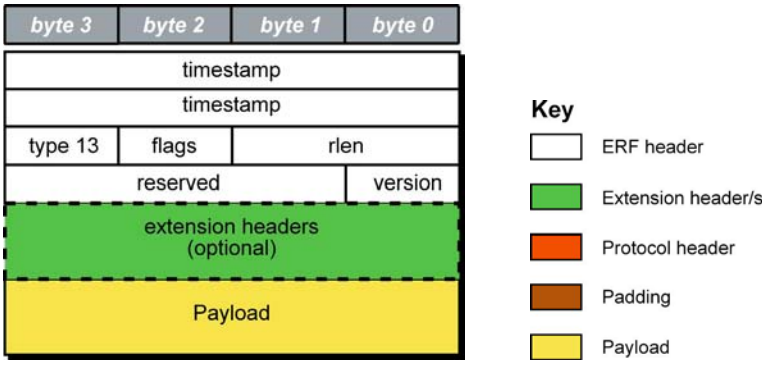
The following is a description of the TYPE_MC_AAL2 record format:

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>flags
(1 byte1)</td><td>This field is the same as normal ERF types but capture interface is always zero.
• Fixed length mode not supported.
• RX Error is set if any MC Header Error bit is set.</td></tr>
<tr><td>MC header
(4 bytes)</td><td>Protocol header. This field is divided into the following:
<table>
<tr><th>Bits</th><th>Attribute</th></tr>
<tr><td>0-9</td><td>Connection number (0-1023).
512 connections are supported by DAG 3.7T card.</td></tr>
<tr><td>10-12</td><td>Reserved for possible extra connection numbers</td></tr>
<tr><td>13-15</td><td>Reserved for indication of AAL2 type (a value of 0x0 indicates a SSSAR packet).</td></tr>
<tr><td>16-19</td><td>Physical port (0-15) cell was captured on.
Physical ID is interpreted from the firmware perspective. For example, if a cable is plugged into port 0, examining the ERF MC Header field will give a Physical ID of 11. This is a little counter-intuitive and reflects the internal processing required. From the software/user perspective, this could be interpreted as the Logical ID, and as such, we can convert from the Logical to Physical ID using the provided dagutil function, dagutil_37t_line_get_logical which will return the Software Physical ID/Firmware Logical ID. In this case, assuming data is coming in on a cable plugged into port 0, we will convert 11 back to 0. For the 7.1S this field is always 0.</td></tr>
<tr><td>20</td><td>Reserved</td></tr>
<tr><td>21</td><td>1st Cell. This is the first cell received since this connection was configured.</td></tr>
<tr><td>22</td><td>MAAL Error (errnum as specified in ITU I.363.2 is copied to the data part of this record)</td></tr>
<tr><td>23</td><td>Length Error</td></tr>
<tr><td>24-31</td><td>Channel Identification Number (cid)</td></tr>
</table></td></tr>
<tr><td>ATM header
(4 bytes)</td><td>Protocol header. This does not include the 8-bit HEC.</td></tr>
<tr><td>Payload
(bytes of AAL5 frame)</td><td>Payload = rlen - ERF header (16 bytes) - Extension headers (optional) - Protocol header (8 bytes)</td></tr>
</table>

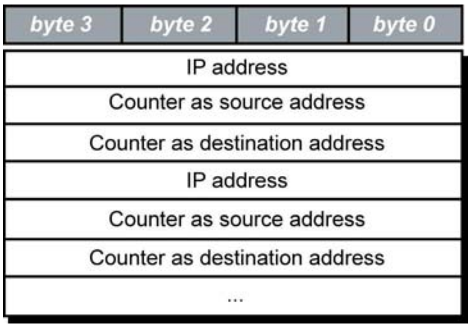
ERF 13. TYPE_IP_COUNTER

Type	Bit 7	1 = Extension header present. See Extension Headers (page 34).
	Bits 6:0	Type 13
Short description	TYPE_IP_COUNTER	
Long description	Type 13 IP Counter ERF Record	
Use	This record format counts IP address records.	

The TYPE_IP_COUNTER record is shown below:



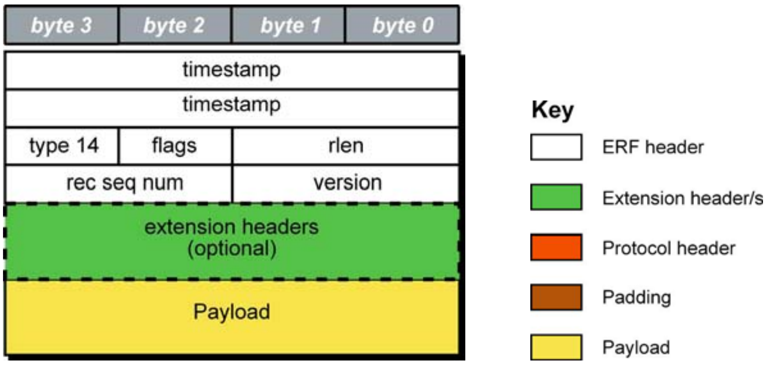
The following is a description of the TYPE_IP_COUNTER record format:

Field	Description
Version (1 byte)	4 bits to identify the version of the counter record used.
Payload (bytes of record)	If version = 1 the following is the record format:  byte 3 byte 2 byte 1 byte 0 IP address Counter as source address Counter as destination address IP address Counter as source address Counter as destination address ...

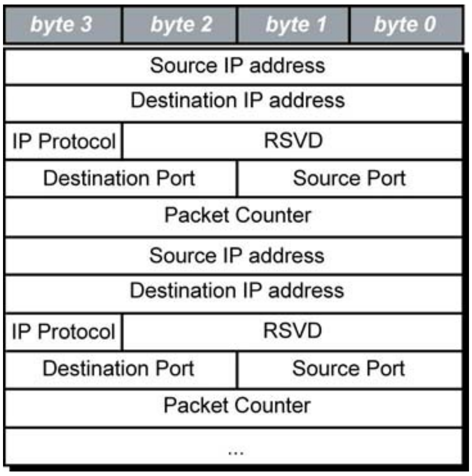
ERF 14. TYPE_TCP_FLOW_COUNTER

Type	Bit 7	1 = Extension header present. See Extension Headers (page 34).
	Bits 6:0	Type 14
Short description	TYPE_TCP_FLOW_COUNTER	
Long description	TCP Flow Counter ERF Record	
Use	This record format counts TCP flow records	

The TYPE_TCP_FLOW_COUNTER record is shown below:



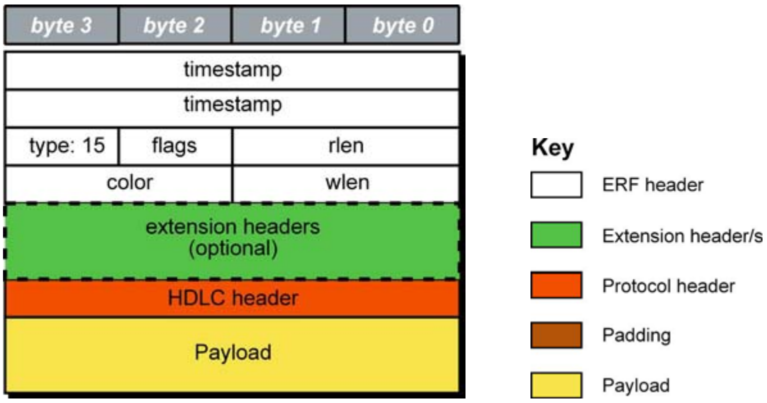
The following is a description of the TYPE_TCP_FLOW_COUNTER record format:

Field	Description
rec seq num (2 bytes)	This is the record counter so the user can tell how many flow records have been received so far.
version (2 bytes)	4 bits to identify the version of the counter record used.
Payload (bytes of record)	If version = 1 the following is the record format:  byte 3 byte 2 byte 1 byte 0 Source IP address Destination IP address IP Protocol RSVD Destination Port Source Port Packet Counter Source IP address Destination IP address IP Protocol RSVD Destination Port Source Port Packet Counter ...

ERF 15. TYPE_DSM_COLOR_HDLC_POS

Type	Bit 7	1 = Extension header present. See Extension Headers (page 34).
	Bits 6:0	Type 15
Short description	TYPE_DSM_COLOR_HDLC_POS	
Long description	Type 15 DSM Color HDLC PoS Record	
Use	This record format is for HDLC data links, incorporating filter results. The record format is the same type as the Type 10 TYPE_COLOR_HDLC_POS (page 17) record, with the exception that the *lctr* field is reassigned as *DSM* type *color*.	

The TYPE_DSM_COLOR_HDLC_POS variable length record is shown below:



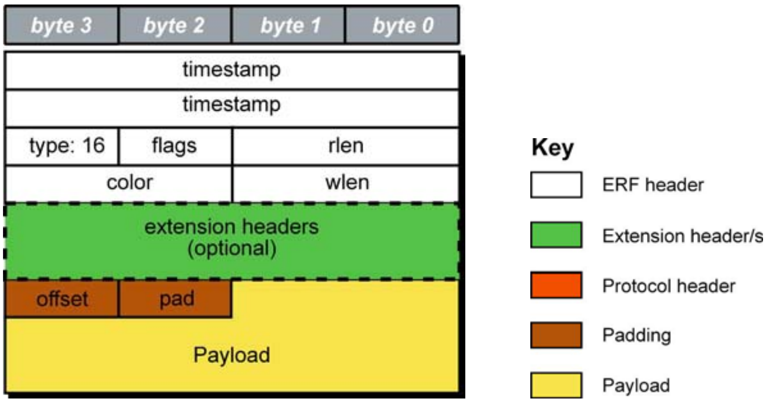
The following is a description of the TYPE_DSM_COLOR_HDLC_POS record format:

Field	Description
color (2 bytes)	The color field is a hardware generated tag indicating the result of a filtering or classification operation. This field is divided into the following: **Bits** / **Description** 0-5: Receive stream number (0-63) 6-13: Filter match bits (bit6 = filter0, bit7 = filter1 and so on). 14: hlb0 (CRC calculation) output bit. 15: hlb1 (parity calculation) output bit.
HDLC header (4 bytes)	Protocol header. Length may vary depending on protocol.
Payload (bytes of record)	Payload = rlen - ERF header (16 bytes) - Extension headers (optional) - Protocol header (4 bytes)

ERF 16. TYPE_DSM_COLOR_ETH

Type	Bit 7	1 = Extension header present. See Extension Headers (page 34).
	Bits 6:0	Type 16
Short description	TYPE_DSM_COLOR_ETH	
Long description	Type 16 DSM Color Ethernet Record	
Use	This record format is for Ethernet [802.3] data links, incorporating filter results. The record format is the same type as the Type 2 TYPE_ETH (page 9) record, with the exception that the *lctr* field reassigned as *DSM* type *color.*	

The TYPE_DSM_COLOR_ETH record is shown below:



The following is a description of the TYPE_DSM_COLOR_ETH record format:

Field	Description
Color (2 bytes)	The color field is a hardware generated tag indicating the result of a filtering or classification operation. This field is divided into the following: Bit: 0-5 — Receive stream number (0-63) Bit: 6-13 — Filter match bits (bit6 = filter0, bit7 = filter1 and so on). Bit: 14 — hlb0 (CRC calculation) output bit. Bit: 15 — hlb1 (parity calculation) output bit.
Offset (1 byte)	Number of bytes **not** captured from the start of the frame. This is typically used to skip link layer headers when they are not required in order to save bandwidth and space. **Note: This field is currently not implemented; contents should be disregarded.**
Pad (1 byte)	The Ethernet frame begins immediately after the pad byte so that the layer 3 [IP] header is 32-bit aligned. This is typically used to skip link layer headers when they are not required in order to save bandwidth and space.
Payload (bytes of record)	Payload = rlen - ERF header (16 bytes) - Extension headers (optional) - Padding (2 bytes)

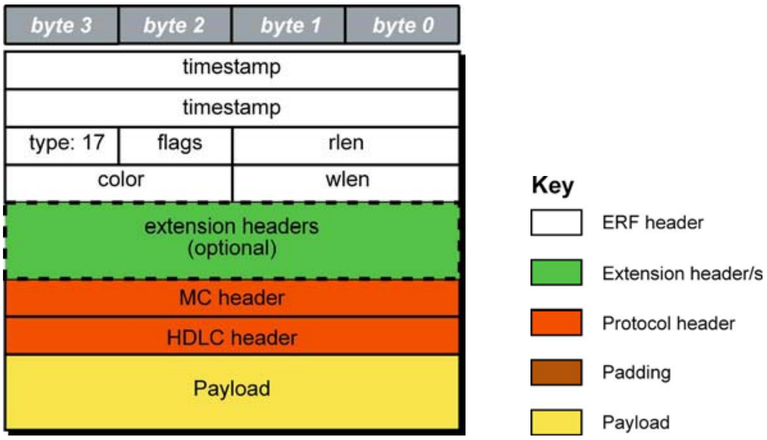
Color field subdivision:

Bit	Description
0-5	Receive stream number (0-63)
6-13	Filter match bits (bit6 = filter0, bit7 = filter1 and so on).
14	hlb0 (CRC calculation) output bit.
15	hlb1 (parity calculation) output bit.

ERF 17. TYPE_MC_HDLC_POS

Type	Bit 7	1 = Extension header present. See Extension Headers (page 34).
	Bits 6:0	Type 17
Short description	TYPE_COLOR_MC_HDLC_POS	
Long description	Type 17 Multi-channel HDLC Frame with Color Record	
Use	This record format is for channelized HDLC data links, incorporating filter results. The record format is the same type as the Type 5 TYPE_MC_HDLC (page 12) record, with the exception that the *lctr* field reassigned as *color*.	

The TYPE_COLOR_MC_HDLC_POS record is shown below:



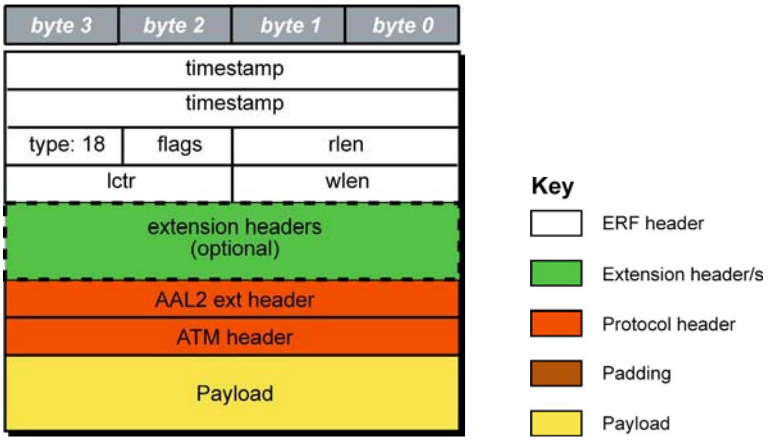
The following is a description of the TYPE_COLOR_MC_HDLC_POS record format:

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>flags
(1 byte)</td><td>Same as normal ERF Types but capture interface is always zero.
• Fixed length mode not supported.
• RX Error is set if any MC header Error bit is set.</td></tr>
<tr><td>Color
(2 bytes)</td><td>This field is divided into the following:
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>0-1</td><td>Stream number of the record, this should match the stream that the packet record was received on.</td></tr>
<tr><td>2-15</td><td>Filter rule match, user defined value that is used to indicate which filter rule matched the packet record.</td></tr>
</table></td></tr>
<tr><td>MC header
(4 bytes)</td><td>Protocol Header. This field is divided into the following:
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>0-9</td><td>Connection number (0-511).</td></tr>
<tr><td>10-15</td><td>Reserved</td></tr>
<tr><td>16-23</td><td>Reserved</td></tr>
<tr><td>24</td><td>FCS Error</td></tr>
<tr><td>25</td><td>Short Record Error (<5 Bytes)</td></tr>
<tr><td>26</td><td>Long Record Error (>2047 Bytes)</td></tr>
<tr><td>27</td><td>Aborted Frame Error</td></tr>
<tr><td>28</td><td>Octet Error. The closing flag wasn't octet aligned after bit unstuffing.</td></tr>
<tr><td>29</td><td>Lost Byte Error. The internal datapath had an unrecoverable error.</td></tr>
<tr><td>30</td><td>1st Rec. This is the first record received since this connection was configured.</td></tr>
<tr><td>31</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>HDLC header
(4 bytes)</td><td>Protocol Header. Length may vary depending on protocol.</td></tr>
<tr><td>Payload
(bytes of packet)</td><td>Payload = rlen - ERF header (16 bytes) - Extension headers (optional) - Protocol header (8 bytes)</td></tr>
</table>

ERF 18. TYPE_AAL2

Type	Bit 7	1 = Extension header present. See Extension Headers (page 34).
	Bits 6:0	Type 18
Short description	TYPE_AAL2	
Long description	Type 18 Reassembled AAL2 Frame Record	
Use	This record is for reassembled ATM AAL2 frames.	

The TYPE_AAL2 record is shown below:



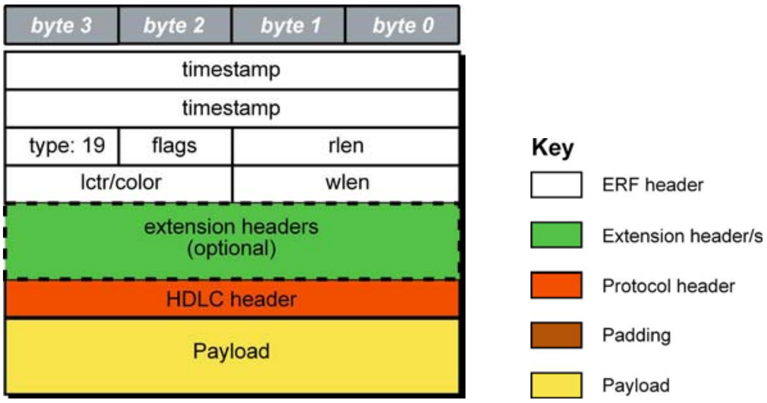
The following is a description of the TYPE_AAL2 record format:

Field	Description
flags (1 byte)	This field is divided into the following: **Bit** / **Description** 0 — MAAL Error Indication, will be set if the frame has a MAAL error otherwise it is cleared. 1 — 1st Frame Indicator, will be set if this is the first frame reassembed on the Interface/Channel/VPI/VCI/CID. 2-7 — Reserved
AAL2 ext header (4 bytes)	Protocol Header. This field is divided into the following: **Field** / **Description** 0-7 — Channel Identification Number (cid) 8-15 — MAAL Error (errnum as specified in ITU I.363.2 is copied to the data part of this record) 16-23 — AAL2 flags, see above. 24-31 — Reserved
ATM header (4 bytes)	Protocol Header. This does not include the 8-bit HEC.
Payload (bytes of AAL2 frame)	Payload = rlen - ERF header (16 bytes) - Extension headers (optional) - Protocol header (8 bytes)

ERF 19. TYPE_COLOR_HASH_POS

Type	Bit 7	1 = Extension header present. See Extension Headers (page 34).
	Bits 6:0	Type 19
Short description	TYPE_COLOR_HASH_POS	
Long description	Type 19 Colored PoS HDLC record with Hash load balancing.	
Use	This record format is for data links, incorporating filter results. The record format is the same type as the Type 1 POS_HDLC (page 8) record, but with IPF color and hash value instead of the loss counter field.	

The TYPE_COLOR_HASH_POS record is shown below:



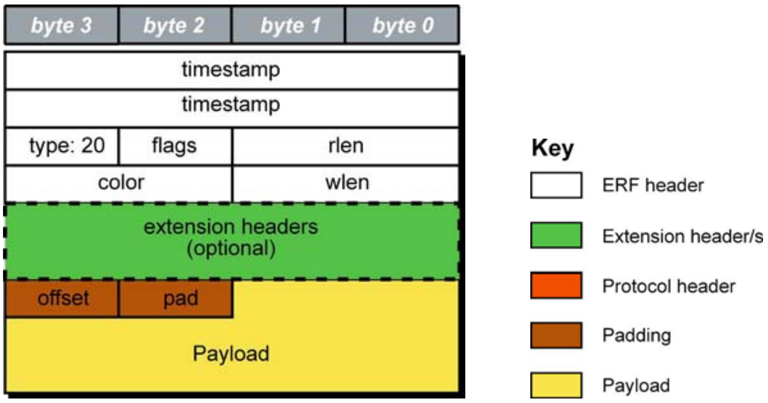
The following is a description of the TYPE_COLOR_HASH_POS record format:

Field	Description
color (2 bytes)	The color field is a hardware generated tag indicating the result of a filtering or classification operation. This field is divided into the following: Bit / Description 0-3 / Hash Value 4-16 / IPF Color
HDLC header (4 bytes)	Protocol header. Length may vary depending on protocol.
Payload (bytes of record)	Payload = rlen - ERF header (16 bytes) - Extension headers (optional) - Protocol header (4 bytes)

ERF 20. TYPE_COLOR_HASH_ETH

Type	Bit 7	1 = Extension header present. See Extension Headers (page 34).
	Bits 6:0	Type 20
Short description	TYPE_COLOR_HASH_ETH	
Long description	Type 20 Colored Ethernet variable length record with hash load balancing.	
Use	This record is like Type 2 TYPE_ETH (page 9), but with IPF color and hash value instead of the loss counter field.	

The TYPE_COLOR_HASH_ETH record is shown below:



The following is a description of the TYPE_COLOR_HASH_ETH record format:

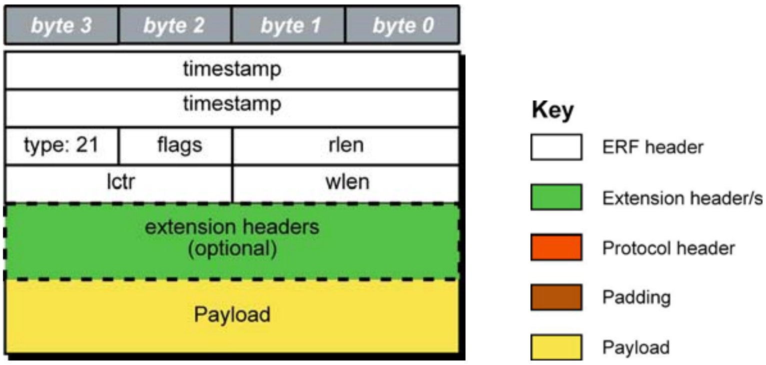
Field	Description
color (2 bytes)	The color field is a hardware generated tag indicating the result of a filtering or classification operation. This field is divided into the following: Bit 0-3: Hash Value Bit 4-16: IPF Color
Offset (1 byte)	Number of bytes that were not captured from the start of the frame. This is typically used to skip link layer headers when they are not required in order to save bandwidth and space. This field is currently not implemented; contents should be disregarded.
Pad (1 byte)	The Color Ethernet frame begins immediately after the pad byte so that the layer 3 [IP] header is 32-bit aligned. This is typically used to skip link layer headers when they are not required in order to save bandwidth and space.
Payload (bytes of record)	Payload = rlen - ERF header (16 bytes) - Extension headers (optional) - Protocol header (2 bytes)

Bit	Description
0-3	Hash Value
4-16	IPF Color

ERF 21. TYPE_INFINIBAND

Type	Bit 7	1 = Extension header present. See Extension Headers (page 34).
	Bits 6:0	Type 21
Short description	TYPE_INFINIBAND	
Long description	Type 21 Infiniband Variable Length Record.	
Use	This record format captures Infiniband data. Used in conjunction with EH 3. Classification (page 36).	

The TYPE_INFINIBAND record is shown below:



byte 3	byte 2	byte 1	byte 0
timestamp			
timestamp			
type: 21	flags	rlen	
lctr		wlen	
extension headers (optional)			
Payload			

Key

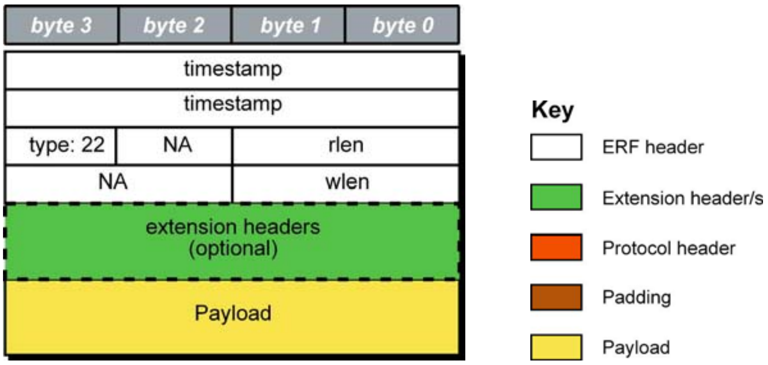
- □ ERF header
- 🟩 Extension header/s
- 🟥 Protocol header
- 🟫 Padding
- 🟨 Payload

Field	Description
Payload (bytes of record)	Payload = rlen - ERF header (16 bytes) - Extension headers (optional)

ERF 22. TYPE_IPV4

Type	Bit 7	1 = Extension header present. See Extension Headers (page 34).
	Bits 6:0	Type 22
Short description	TYPE_IPV4	
Long description	Type 22 IPV4 Variable Length Record.	
Use	This is a layer III single packet record.	

The TYPE_IPV4 record is shown below:



The following is a description of the TYPE_IPV4 record format:

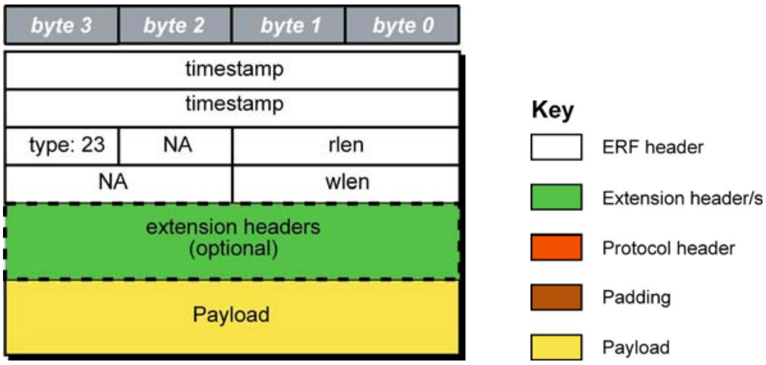
This is a layer-III ERF record. Payload consists of a single IPV4 packet. Layer-II information such as MPLS Tags, VLAN Tags and MAC addresses, POS Headers etc are not present.

Field	Description
Payload (bytes of record)	Payload = rlen - ERF header (16 bytes) - Extension headers (optional)

ERF 23. TYPE_IPV6

Type	Bit 7	1 = Extension header present. See Extension Headers (page 34).
	Bits 6:0	Type 23
Short description	TYPE_IPV6	
Long description	Type 23 IPV6 Variable Length Record	
Use	This is a layer III single packet record.	

The TYPE_IPV6 record is shown below:



The following is a description of the TYPE_IPV6 record format:

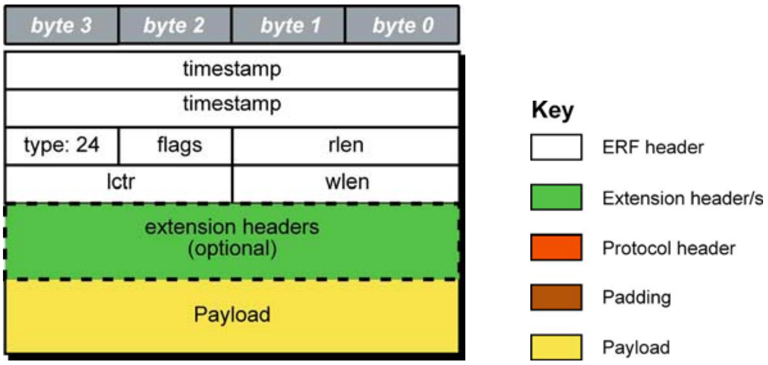
This is a layer-III ERF record. Payload consists of a single IPV6 packet. Layer-II information such as MPLS Tags, VLAN Tags and MAC addresses, POS Headers etc are not present.

Field	Description
Payload (bytes of record)	Payload = rlen - ERF header (16 bytes) - Extension headers (optional)

ERF 24. TYPE_RAW_LINK

Type	Bit 7	1 = Extension header present. See Extension Headers (page 34).
	Bits 6:0	Type 24
Short description	TYPE_RAW_LINK	
Long description	Type 24 Raw link data, typically SONET or SDH Frame	
Use	Used in Raw Capture image for SONET/SDH. Used with Extension Header 5 (page 38).	

The TYPE_RAW_LINK record is shown below:



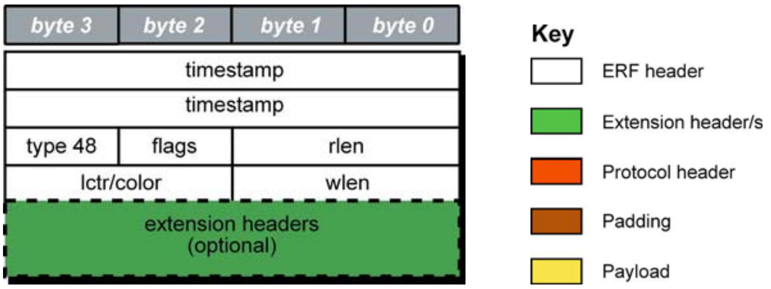
The following is a description of the TYPE_RAW_LINK record format:

Field	Description
Payload (bytes of record)	Payload = rlen - ERF header (16 bytes) - Extension headers (optional)

ERF 48. TYPE_PAD

Type	Bit 7	1 = Extension header present. See Extension Headers (page 34).
	Bits 6:0	Type 48
Short description	TYPE_PAD	
Long description	Type 48 Pad record	
Use	This record type is for pad records in DAG-II (and anywhere else that needs it).	

The TYPE_PAD record is shown below:



The following is a description of the TYPE_PAD record format:

Field	Description
timestamp (4 bytes)	All zeroes
type (1 byte)	48 (0x30)
flags (1 byte)	A value of 0
rlen (2 bytes)	16 in the first version (Currently, all pad records are 16 bytes for simplicity. This could change in the future, as other uses are made of these records.)
loss counter/color (2 bytes)	A value of 0
wlen (2 bytes)	A value of 0

Extensible Record Format Timestamps

Overview

The Extensible Record Format (ERF) incorporates a hardware generated timestamp of the packet's arrival.

The format of this timestamp is a single little-endian 64-bit fixed point number, representing whole and fractional seconds since midnight on the first of January 1970.

The high 32-bits contain the integer number of seconds, while the lower 32-bits contain the binary fraction of the second. This allows an ultimate resolution of 2^{-32} seconds, or approximately 233 picoseconds.

Another advantage of the ERF timestamp format is that a difference between two timestamps can be found with a single 64-bit subtraction.

It is not necessary to check for overflows between the two halves of the structure as is needed when comparing UNIX time structures, which are also available to Windows users in the Winsock library.

DAG card resolutions

Different DAG cards have different actual resolutions. This is accommodated by the lowermost bits that are not active being set to zero. In this way the interpretation of the timestamp does not need to change when higher resolution clock hardware is available.

Example code

The following is example code showing how a 64-bit ERF timestamp (erfts) can be converted into a struct timeval representation (tv).

```
unsigned long long lts;
struct timeval tv;
 lts = erfts;
 tv.tv_sec = lts >> 32;
 lts = ((lts & 0xffffffffULL) * 1000 * 1000);
 lts += (lts & 0x80000000ULL) << 1;      /* rounding */
 tv.tv_usec = lts >> 32;
 if(tv.tv_usec >= 1000000) {
 tv.tv_usec -= 1000000;
 tv.tv_sec += 1;
 }
```

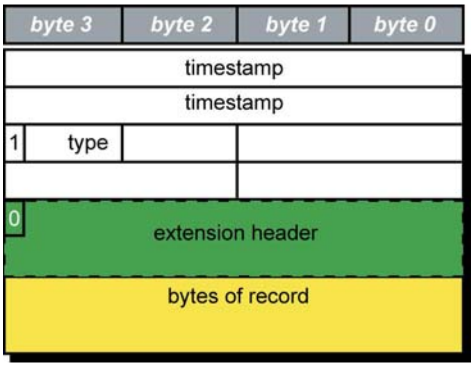
Extension Headers (EH)

Introduction

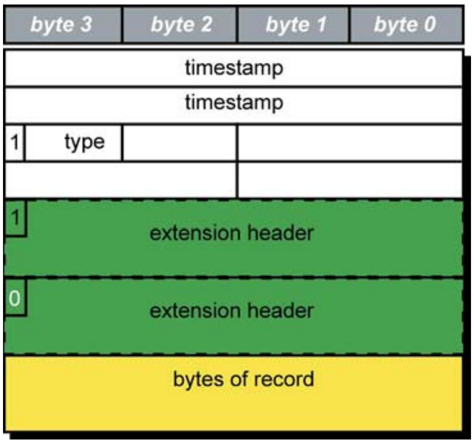
The addition of an Extension Header into the ERF record allows extra data relating to the packet to be transported to the host. The extension header allows certain features to be added independently of ERF types, for example, features shared by different ERF records do not have to be implemented separately. This results in automatic support across ERF types.

Bit 7 of the ERF type field is used to indicate that Extension Header's are present. If set to '1' Extension Headers are present. The Extension Header type field indicates the type and format of the Extension Header. It also indicates whether further Extension Headers are present. If bit 7 of the Extension Header is set to '1' further Extension Headers exist in the record. The Extension Headers are 8 bytes in length.

The following diagram shows presence of an Extension Header in addition to the ERF record.



The following diagram shows presence of two Extension Headers with Bit 7 of the first Extension Header set to '1'.



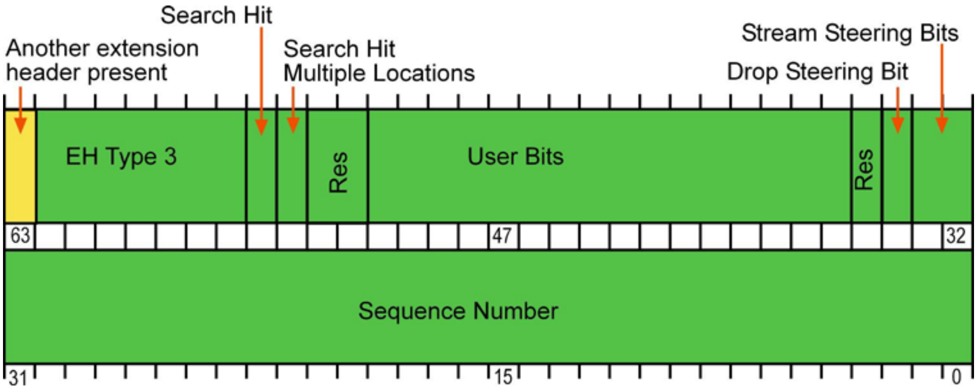
Extension Headers Types

Number	Type	Description
0:	Reserved	Reserved.
1:	Reserved	Reserved.
2:	Reserved	Reserved.
3:	Classification	Used to report filter and steering results. Used in conjunction with ERF 21. TYPE_INFINIBAND (page 28)
4:	Intercept_ID	ID attached to intercepted packet.
5:	Raw_Link	Extra information for ERF 24. TYPE_RAW_LINK (page 31) records.

EH 3. Classification

Type	Bit 7	Extension header present
	Bits 6:0	Type 3
Short description	Classification	
Long description	-	
Use	Used with ERF 21. TYPE_INFINIBAND (page 28). Entries marked Metadata are derived by firmware. Entries marked SRAM are stored in the TCAM Associated SRAM.	

Note: The following is provisional and subject to change.



The following details the make up of the Classification Extension Header:

Bit	Length	Meaning
63	1	More Extension Headers present (1 = more)
62:56	7	0x03 - Assigned type code.
55	1	Search Hit, rest of bits are meaningful.
54	1	Search Hit Multiple Locations, lowest-numbered shown.
53:52	2	Reserved.
51:36	16	User Bits.
35	1	Reserved.
34	1	Drop Steering Bit. May have Stream Steering bits set too.
33:32	2	Stream Steering Bits. Binary encoded.
31:0	32	Sequence Number from Blackbird framer chip.

Note: For NinjaProbe 40G1 this is the format at the output of the RXOne chip, and therefore the input to the Steering logic and also the Software.

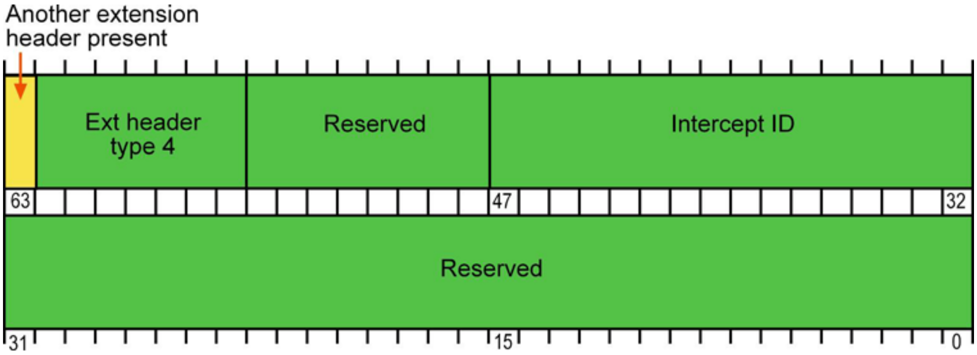
Bits 31:0 are optional for InfiniBand and can be sequence number or set to 0. TCAM Associated SRAM Data (colour) for InfiniBand and NinjaProbe 40G1 used for classification.

Bit	Length	Meaning
31:20	12	Reserved set to 0
19:4	16	Tag (user classification(data))
3	1	Reserved
2	1	Drop Steering Bit. May have Stream Steering bits set too.
1:0	2	Stream Steering Bits. Binary encoded.

EH 4. Intercept ID

Type	Bit 7	Extension header present
	Bits 6:0	Type 4
Short description	Intercept ID	
Long description	ID attached to intercepted packet.	
Use	Used to identify packet as associated with a unique ID.	

The Intercept_ID record is shown below:



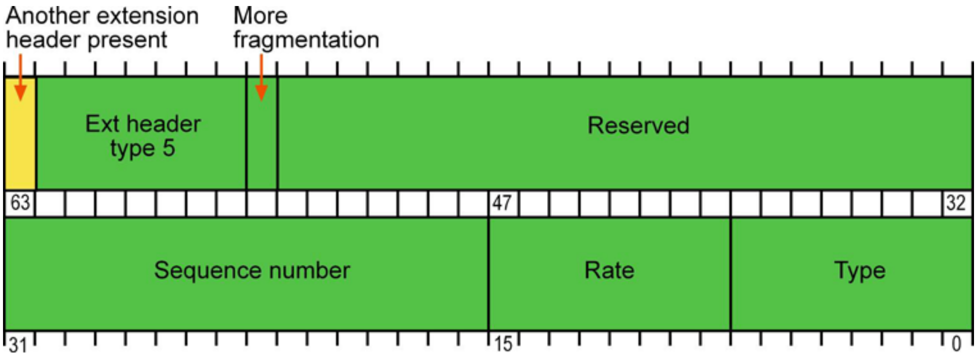
The following details the make up of the Intercept_ID Extension Header:

Bit	Length	Meaning
63	1	More Extension Headers present (1 = more).
62:56	7	0x04 - Assigned type code.
55:48	8	Reserved.
47:32	16	InterceptID. Integer. Unique ID.
31:0	16	Reserved.

EH 5. Raw_Link

Type	Bit 7	Extension header present
	Bits 6:0	Type 5
Short description	Raw_Link	
Long description	Extra information for TYPE_RAW_LINK records	
Use	Used in Raw Capture image for SONET/SDH. Used with ERF 24. TYPE_RAW_LINK (page 31).	

The Raw_Link record is shown below:



The following details the make up of the Raw_Link Extension Header:

Bit	Length	Meaning
63	1	More Extension Headers present (1 = more).
62:56	7	0x05 - Assigned type code.
55	1	More fragmentation. (0 = Start of Frame, 1 = More Fragmentation)
54:32	39	Reserved.
31:16	16	Sequence number (starting at 0)
15:8	8	Rate. • 0 = reserved • 1 = OC3 • 2 = OC12 • 3 = OC48 • 4 = OC192 As defined in the SONET control register.
7:0	8	Type. • 0 = SONET mode • 1 = SDH • others are reserved for future use. As defined in the SONET control register.

Version History

Version	Date	Reason
1 - 2	-	Previous versions
3	October 2005	
4	August 2007	Added new data formats and updated existing data formats.
5	November 2007	Added Extension Headers 3,4 and records 19,20,22,23.
6	December 2007	Added ERF Type 21 and updated ERF types per DAG card
7	February 2008	Added ERF type 24 and EH 5. Defined Payload field in ERF types.
8	June 2008	Corrected ERF types per card information for the 5.4 and 5.4A DAG cards.