DRAFT

BSR E1.31 — 202x

Entertainment Technology

Lightweight streaming protocol for transport of DMX512 using ACN

Document number: CP/2014-1009r7c

Approved by the ANSI Board of Standards Review on __________

Notice and Disclaimer

ESTA does not approve, inspect, or certify any installations, procedures, equipment or materials for compliance with codes, recommended practices, or standards. Compliance with an ESTA standard or recommended practice is the sole and exclusive responsibility of the manufacturer or provider and is entirely within their control and discretion. Any markings, identification or other claims of compliance do not constitute certification or approval of any type or nature whatsoever by ESTA.

ESTA neither guarantees nor warrants the accuracy or completeness of any information published herein and disclaim liability for any personal injury, property or other damage or injury of any nature whatsoever, whether special, indirect, consequential or compensatory, directly or indirectly resulting from the publication, use of, or reliance on this document.

In issuing and distributing this document, ESTA does not either (a) undertake to render professional or other services for or on behalf of any person or entity, or (b) undertake any duty to any person or entity with respect to this document or its contents. Anyone using this document should rely on their own independent judgment or, as appropriate, seek the advice of a competent professional in determining the exercise of reasonable care in any given circumstance.

Published by:

Entertainment Services and Technology Association (ESTA)
271 Cadman Plaza PO Box 23200
Brooklyn, NY 11202-3200
USA
Phone: +1-212-244-1505
Email: standards@esta.org

ESTA's Technical Standards Program

The ESTA Technical Standards Program was created to serve the ESTA membership and the entertainment industry in technical standards related matters. The goal of the program is to take a leading role regarding technology and safety within the entertainment industry by creating recommended practices and standards, monitoring standards issues around the world on benefit of our members, and improving communications and safety within the industry. ESTA works closely with the technical standards efforts of other organizations within our industry including ESA, CITT, USITT and VPLT as well as representing the interests of ESTA members to ANSI, UL, ASCE, ICC, and the NFPA. ESTA is an ANSI Accredited Standards Developer.

The Technical Standards Council (TSC) established by ESTA's Board of Directors to oversee and coordinate the Technical Standards Program. Made up of individuals experienced in standards development work from throughout our industry, the Committee approves all projects undertaken and assigns them to the appropriate working group. The Technical Standards Council employs a Technical Standards Manager to coordinate the work of the Committee and its working groups as well as maintaining a "Standards Watch" on behalf of members. Working groups include: Control Protocols, Electrical Power, Event Safety, Floors, Fog and Smoke, Followspot Positions, Photometrics, Rigging, Stage Machinery, and Prop Weapons Safety.

ESTA encourages active participation in the Technical Standards Program. There are several ways to become involved. The easiest way to actively participate is to respond to any of the public reviews advertised on ESTA's public review web page. The next level of participation requires completion of an application to become a working group member; applications are available from the TSP's procedural documents web page. Application as an Observer member affords access to updates on standards development documents. Application as a voting member affords full participation as a consensus voice that helps shape the industry. Requirements for voting membership include responding to letter ballots and attending meetings, but membership in ESTA or any other organization is not a requirement for participation in the TSP. One can also become involved by requesting that the TSC develop a standard or a recommended practice in an area of concern to them.

The Control Protocols Working Group, which authored this standard, consists of a cross section of entertainment industry professionals representing a diversity of interests. ESTA is committed to developing consensus-based standards and recommended practices in an open setting.

Investors in Innovation

The Technical Standard Program (TSP) is financially supported by ESTA and by companies and individuals who make donations to the TSP. Contributing companies and individuals who have helped fund the TSP are recognized as “Investors in Innovation”. These are the Investors in Innovation when this standard was approved by ANSI's Board of Standards Review:

[This is a placeholder for the table current at the time of approval]

Memorial donor: The Estate of Ken Vannice

All donations to the Technical Standards Program benefit the entire program, and are not directed to any specific use or project within the program. Please help support the Technical Standards Program by becoming an Investor in Innovation. Visit our website at http://tsp.esta.org/invest, or contact the ESTA office at 1-212-244-1505 and select "TSP" from the menu.

Contact Information

Technical Standards Manager
Richard J. Nix
ESTA
271 Cadman Plaza PO Box 23200
New York, NY 11202-3200
USA
+1-212-244-1505
richard.nix@esta.org

Technical Standards Council Chairperson
Alan Rowe
I.A.T.S.E Local 728
Phone: 1 310-702-2909
amrowe@iatse728.org

Control Protocols Working Group Co-chairpersons

Milton Davis
Doug Fleenor Design, Inc.
+1-805-481-9599
milton@dfd.com

John David (Javid) Butler
Goddard Design LLC
+1-702-759-2427
javid@goddarddesign.com

Maya Nigrosh
+1-412-901-7446
mnigrosh@alumni.cmu.edu

Acknowledgments

The Control Protocols Working Group members, when this document was approved by the working group on ________, are shown below.

Voting members:

Observer members:

Interest category codes:

CP = custom-market producer	DE = designer
DR = dealer rental company	G = general interest
MP = mass-market producer	U = user

Table of Contents

List of Tables

List of Figures

1. Introduction

1.1 Scope

This standard describes a mechanism to transfer DMX512-A [DMX] packets over an IP network using a subset of the ACN protocol suite. It covers data format, data protocol, data addressing, and network management. It also outlines a synchronization method to help ensure that multiple receivers can process this data concurrently when supervised by the same source. Sources transporting either data or synchronization packets must also advertise, via the Universe Discovery mechanism, what universes they are actively transmitting on.

1.2 Overview and Architecture

This standard can be used to transfer DMX512-A [DMX] packets of all START Codes via an ANSI E1.17 [ACN] supported network. It also defines a method by which this [DMX] data may be synchronized across multiple receivers. A simple packet wrapper approach is used whereby the data is encapsulated in a wrapper following the ACN packet structure. For the use of this standard, the ACN wrapper is carried in UDP [UDP] packets.

The wrapper is structured such that it is both compatible and meaningful to the ANSI E1.17 [ACN] standard. Readers are referred to the ANSI E1.17 [ACN] standard, particularly the “ACN Architecture” and “Device Management Protocol” documents for more information. The “Root Layer Protocol” used in this standard is described in the “ACN Architecture” document.

This standard uses multicast addressing to provide a mechanism to partition traffic for distinct universes of DMX512-A [DMX] and synchronization data. Direct unicast of DMX512-A [DMX] data is also supported.

1.3 Appropriate Use of This Standard

This standard uses a non-reliable IP transport mechanism to stream packets of data from multiple sources to multiple receivers over the ACN network. There is no acknowledgement and therefore no assurance that all packets have been received.

1.4 Classes of Data Appropriate for Transmission

This standard, E1.31, is intended to define a method to carry DMX512-A [DMX] style data and metadata over IP Networks. It is designed to carry repetitive control data from one or more sources to one or more receivers. This protocol is intended to be used to control dimmers, other lighting devices, and related non-hazardous effects equipment.

1.5 Universe Synchronization

This standard defines a mechanism by which data streamed by a single source to many receivers may be synchronized across a network. Through the use of synchronization packets, which are distinct in format from data packets, a source can declare when all of its E1.31 data has been sent and can then be acted upon synchronously.
This methodology may find most of its use in media applications, but can be expanded to a variety of environments.

1.6 Universe Discovery

This standard includes a Universe Discovery packet that sources must provide in order to enumerate the universes upon which they are transmitting. This allows other devices interested in network traffic to monitor which universes are currently active, without the need to join every multicast group to examine their individual transmissions.

1.7 Compliance

Compliance with this standard is strictly voluntary and the responsibility of the implementer. Markings and identification or other claims of compliance do not constitute certification or approval by the E1 accredited standards committee.

2. Normative References

[DMX] ANSI E1.11 Entertainment Technology – USITT DMX512-A Asynchronous Serial Digital Data Transmission Standard for controlling lighting equipment and accessories.

This standard is maintained by Entertainment Services and Technology Association
ESTA
271 Cadman Plaza
PO Box 23200
New York, NY 11202-3200
USA
Phone: 1-212-244-1505
Fax: 1-212-244-1502
standards@esta.org
http://www.tsp.esta.org

ESTA is a standardization body accredited by ANSI to develop, maintain, and withdraw American National Standards.

ANSI
25 West 43rd Street
4th floor
New York, NY 10036
+1-212-642-4900
http://www.ansi.org

[ACN] ANSI E1.17 Entertainment Technology—Architecture for Control Networks

This standard is maintained by ESTA.

[ACN-DMP] ANSI E1.17 Architecture for Control Networks—Device Management Protocol

This standard is maintained by ESTA.

[RDM] ANSI E1.20 Entertainment Technology—Remote Device Management over DMX512 networks.

This standard is maintained by ESTA.

[UTF-8] The Unicode Consortium. The Unicode Standard, Version 5.0.0, defined by:
The Unicode Standard, Version 5.0 (Boston, MA, Addison-Wesley, 2007. ISBN 0-321-48091-0)

[UDP] RFC 0768 UDP User Datagram Protocol

This standard is maintained by the IETF.
IETF Secretariat
c/o NeuStar, Inc.
Corporate Headquarters
46000 Center Oak Plaza
Sterling, VA 20166
Phone: +1-571-434-3500
www.ietf.org

[IGMP2] RFC 2236 IGMPv2 Internet Group Management Protocol Version 2.

This standard is maintained by the IETF.

[IGMP3] RFC 3376 Internet Group Management Protocol, Version 3

This standard is maintained by the IETF.

[MLD] RFC 2710 Multicast Listener Discovery (MLD) for IPv6

This standard is maintained by the IETF.

[MLD2] RFC 3810 Multicast Listener Discovery Version 2 (MLDv2) for IPv6

This standard is maintained by the IETF.

[IPv6Addr] RFC 4291 IP Version 6 Addressing Architecture

This standard is maintained by the IETF.

[IPv6McastScope] RFC 7346 IPv6 Multicast Address Scopes

This standard is maintained by the IETF.

[IPv6AddrText] RFC 5952 A Recommendation for IPv6 Address Text Representation

This standard is maintained by the IETF.

[EmbeddedRP] RFC 3956 Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address

This standard is maintained by the IETF.

[UnicastPrefixMcast] RFC 3306 Unicast-Prefix-based IPv6 Multicast Addresses

This standard is maintained by the IETF.

[ASIPM] RFC 2365 Administratively Scoped IP Multicast.

This standard is maintained by the IETF.

[UUID] RFC 4122 P. Leach, M. Mealling, and R. Salz. *A Universally Unique IDentifier (UUID) URN Namespace*. July 2005.

This standard is maintained by the IETF.

3. Definitions

3.1 Octet: An eight-bit byte within a data packet.

3.2 Universe: A set of up to 512 data slots identified by universe number. Note: In E1.31 there may be multiple sources for a universe. See also: Slot.

3.3 Universe Number: Each E1.31 Data Packet contains a universe number identifying the universe it carries. From an ACN perspective, a receiving device has some number of properties whose value is addressed by the combination of a universe number and a data slot number. From an historical perspective, a receiving device consumes some number of DMX512-A [DMX] data slots.

3.4 Slot: A slot is a sequentially numbered octet in a DMX512-A [DMX] packet. A single Universe contains a maximum of 513 Slots, starting at slot 0. Slot 0 is the DMX512-A [DMX] START Code. Slots 1 through 512 are data slots.

3.5 Source: A stream of E1.31 Packets for a universe is said to be sent from a source. A source is uniquely identified by a number in the header of the packet (see field CID in Table 4-1). A source may output multiple streams of data, each for a different universe. Also, multiple sources may output data for a given universe.

3.6 Receiver: A receiver is the intended target of information from a source. A receiver may listen on multiple universes.

3.7 Active Data Slots: When translating ANSI E1.11 DMX512-A [DMX] to E1.31, the active data slots are defined as ranging from data slot 1 to the maximum data slot in the most recently received packet with the corresponding START Code. Devices originating E1.31 shall define their active data slots using the DMP First Property Address and DMP Property Count fields shown in Table 4-1.

3.8 E1.31 Data Packet: There are several types of E1.31 packets, differentiated by the vectors in their Framing Layer. An E1.31 Data Packet is a packet which carries a DMX512-A [DMX] payload. It is transmitted with the VECTOR_E131_DATA_PACKET vector.

3.9 E1.31 Synchronization Packet: There are several types of E1.31 packets, differentiated by the vectors in their Framing Layer. An E1.31 Synchronization Packet is a packet which contains only universe synchronization information and no additional data. It is transmitted with the VECTOR_E131_EXTENDED_SYNCHRONIZATION vector.

3.10 E1.31 Universe Discovery Packet: There are several types of E1.31 packets, differentiated by the vectors in their Framing Layer. An E1.31 Universe Discovery Packet is a packet which contains a packed list of the universes upon which a source is actively operating. It is transmitted with the VECTOR_E131_EXTENDED_DISCOVERY vector.

3.11 E1.31 Packet: Any of the set of packets containing E1.31 Data Packets, E1.31 Synchronization Packets, and E1.31 Universe Discovery Packets.

4. Protocol Packet Structure Summary

E1.31 is a protocol that lives within the suite of protocols defined by the ANSI E1.17 [ACN] standard. The ACN standard provides a method for layering protocols and for using a simple repeating message structure throughout. The lowest layer ACN protocol is called the Root Layer Protocol (RLP), which wraps E1.31 as well as other protocols such as Session Data Transport (SDT). It is not necessary to implement or understand these other protocols to use E1.31 to send DMX512-A [DMX] data over ACN.

The repeating message structure is called the Protocol Data Unit (PDU) format which is fully defined in "Section 2.2 Protocol Data Units and Blocks – The Standard ACN Message Format" and "Section 2.4 PDU Fields" of the "ACN Architecture Document" of ANSI E1.17 [ACN].

E1.31 defines several distinct packets. E1.31 components must support the E1.31 Data Packet for transportation of DMX512-A data, as well as the E1.31 Universe Discovery Packet for tracking of E1.31 traffic. E1.31 components may support the E1.31 Synchronization Packet. A more complete discussion of Universe Synchronization follows in Section 11. Universe Discovery can be found in Section 12.

4.1 E1.31 Data Packet

The E1.31 Data Packet defines an outer layer PDU wrapper that specifies the sequence number of a packet and that carries a block of data. This data block contains a nested PDU containing a single message of the Device Management Protocol of ANSI E1.17 [ACN] to carry DMX512-A [DMX] data. Each PDU contains a length field which equals the length of the entire PDU, including its header and data block information.

All E1.31 data packets shall carry the vector VECTOR_E131_DATA_PACKET in the Framing Layer in order to indicate that their payload is DMX512-A data.

Table 4-1 describes the E1.31 Data Packet format on UDP [UDP].

Table 4-1: E1.31 Data Packet

Octet	Field Size	Field Name	Field Description	Field Contents
Root Layer (See Section 5)				
0-1	2	Preamble Size	Define RLP Preamble Size.	0x0010
2-3	2	Post-amble Size	RLP Post-amble Size.	0x0000
4-15	12	ACN Packet Identifier	Identifies this packet as E1.17	0x41 0x53 0x43 0x2d 0x45 0x31 0x2e 0x31 0x37 0x00 0x00 0x00
16-17	2	Flags and Length	Protocol flags and length	Low 12 bits = PDU length High 4 bits = 0x7
18-21	4	Vector	Identifies RLP Data as 1.31 Protocol PDU	VECTOR_ROOT_E131_DATA
22-37	16	CID	Sender's CID	Sender's unique ID
E1.31 Framing Layer (See Section 6)				
38-39	2	Flags and Length	Protocol flags and length	Low 12 bits = PDU length High 4 bits = 0x7
40-43	4	Vector	Identifies 1.31 data as DMP Protocol PDU	VECTOR_E131_DATA_PACKET (DMX512-A [DMX] data)
44-107	64	Source Name	User Assigned Name of Source	UTF-8 [UTF-8] encoded string, null-terminated
108	1	Priority	Data priority if multiple sources	0-200, default of 100
109-110	2	Synchronization Address	Universe address on which sync packets will be sent	Universe on which synchronization packets are transmitted
111	1	Sequence Number	Sequence Number	To detect duplicate or out of order packets

112	1	Options	Options Flags	Bit 7 = Preview_Data Bit 6 = Stream_Terminated Bit 5 = Force_Synchronization
113-114	2	Universe	Universe Number	Identifier for a distinct stream of DMX512-A [DMX] Data
DMP Layer (See Section 7)				
115-116	2	Flags and Length	Protocol flags and length	Low 12 bits = PDU length High 4 bits = 0x7
117	1	Vector	Identifies DMP Set Property Message PDU	VECTOR_DMP_SET_PROPERTY (from [ACN-DMP] 13.2)
118	1	Address Type & Data Type	Identifies format of address and data	0xa1
119-120	2	First Property Address	Indicates DMX512-A START Code is at DMP address 0	0x0000
121-122	2	Address Increment	Indicates each property is 1 octet	0x0001
123-124	2	Property value count	Indicates 1+ the number of slots in packet	0x0001 -- 0x0201
125-637	1-513	Property values	DMX512-A START Code + data	START Code + Data

All packet contents shall be transmitted in network byte order (big endian).

4.2 E1.31 Synchronization Packet

The E1.31 Synchronization Packet is used to trigger synchronization. There is no additional data payload. Note: designation of a specific universe as the synchronization address does not preclude that universe from also transporting E1.31 Data.

All E1.31 Synchronization Packets shall carry the vector VECTOR_E131_EXTENDED_SYNCHRONIZATION in the Framing Layer in order to indicate that they carry synchronization information only.

Table 4-2 describes the E1.31 Synchronization Packet format on UDP [UDP].

Table 4-2: E1.31 Synchronization Packet Format

Octet	Field Size	Field Name	Field Description	Field Contents
Root Layer (See Section 5)				
0-1	2	Preamble Size	Define RLP Preamble Size.	0x0010
2-3	2	Post-amble Size	RLP Post-amble Size.	0x0000
4-15	12	ACN Packet Identifier	Identifies this packet as E1.17	0x41 0x53 0x43 0x2d 0x45 0x31 0x2e 0x31 0x37 0x00 0x00 0x00
16-17	2	Flags and Length	Protocol flags and length	Low 12 bits = PDU length High 4 bits = 0x7
18-21	4	Vector	Identifies RLP Data as 1.31 Protocol PDU	VECTOR_ROOT_E131_EXTENDED
22-37	16	CID	Sender's CID	Sender's unique ID
E1.31 Framing Layer (See Section 6)				
38-39	2	Flags and Length	Protocol flags and length	Low 12 bits = PDU length High 4 bits = 0x7
40-43	4	Vector	Identifies 1.31 data as DMP Protocol PDU	VECTOR_E131_EXTENDED_SYNCHRONIZATION (universe synchronization)
44	1	Sequence Number	Sequence Number	To detect duplicate or out of order packets
45-46	2	Synchronization Address	Universe Number	Universe on which synchronization packets are transmitted.
47-48	2	Reserved		Reserved (See Section 6.3.4)

All packet contents shall be transmitted in network byte order (big endian).

4.3 E1.31 Universe Discovery Packet

E1.31 Universe Discovery Packets shall be sent by a source to advertise which E1.31 universes it is actively transmitting on. A set of packets shall be sent once every E131_UNIVERSE_DISCOVERY_INTERVAL, with an indicator as to what page number in the sequence of packets this is, as well as the number of total pages of packets that are intended to be transmitted. The main payload shall consist of a Universe Discovery Layer that contains these page numbers and a sorted list of E1.31 universes-- including synchronization universes, if being transmitted

In the event that the list of universes has changed within E131_UNIVERSE_DISCOVERY_INTERVAL, a source may send up to one additional set of packets to update its information, without waiting for the full time to elapse.

All E1.31 Universe Discovery Packets shall carry the vector VECTOR_E131_EXTENDED_DISCOVERY in the Framing Layer in order to indicate that they are reporting universe discovery data only.

Table 4-3 describes the E1.31 Universe Discovery Packet format over UDP [UDP].

Table 4-3: E1.31 Universe Discovery Packet Format

Octet	Field Size	Field Name	Field Description	Field Contents
Root Layer (See Section 5)				
0-1	2	Preamble Size	Define RLP Preamble Size.	0x0010
2-3	2	Post-amble Size	RLP Post-amble Size.	0x0000
4-15	12	ACN Packet Identifier	Identifies this packet as E1.17	0x41 0x53 0x43 0x2d 0x45 0x31 0x2e 0x31 0x37 0x00 0x00 0x00
16, 17	2	Flags and Length	Protocol flags and length	Low 12 bits = PDU length High 4 bits = 0x7
18-21	4	Vector	Identifies RLP Data as 1.31 Protocol PDU	VECTOR_ROOT_E131_EXTENDED
22-37	16	CID	Sender's CID	Sender's unique ID
E1.31 Framing Layer (See Section 6)				
38-39	2	Flags and Length	Protocol flags and length	Low 12 bits = PDU length High 4 bits = 0x7
40-43	4	Vector	Identifies 1.31 data as Universe Discovery	VECTOR_E131_EXTENDED_DISCOVERY (universe discovery)
44-107	64	Source Name	User Assigned Name of Source	UTF-8 [UTF-8] encoded string, null-terminated
108-111	4	Reserved		Reserved (See Section 6.4.3)
Universe Discovery Layer (See Section 8)				
112-113	2	Flags and Length	Protocol flags and length	Low 12 bits = PDU length High 4 bits = 0x7
114-117	4	Vector	Identifies Universe Discovery data as universe list	VECTOR_UNIVERSE_DISCOVERY_UNIVERSE_LIST
118	1	Page	Packet Number	Identifier indicating which packet of N this is—page numbers start at 0.
119	1	Last Page	Final Page	Page number of the final page to be transmitted.
120-1143	0-1024	List of Universes	Sorted list of up to 512 16-bit universes.	Universes upon which data is being transmitted.

All packet contents shall be transmitted in network byte order (big endian).

5. E1.31 use of the ACN Root Layer Protocol

E1.31 shall use the ACN Root Layer Protocol as defined in the ANSI E1.17 [ACN] “ACN Architecture” document. The fields described here are for E1.31 on UDP [UDP]. Alternative Root Layer Protocol (RLP) content may be defined by further standards in order to transport E1.31 on other protocols. Every type of E1.31 packet shall conform to the same RLP structure.

Table 5-1: E1.31 Root Layer

Octet	Field Size	Field Name	Field Description	Field Contents
Root Layer (See Section 5)				
0-1	2	Preamble Size	Define RLP Preamble Size.	0x0010
2-3	2	Post-amble Size	RLP Post-amble Size.	0x0000
4-15	12	ACN Packet Identifier	Identifies this packet as E1.17	0x41 0x53 0x43 0x2d 0x45 0x31 0x2e 0x31 0x37 0x00 0x00 0x00
16-17	2	Flags and Length	Protocol flags and length	Low 12 bits = PDU length High 4 bits = 0x7
18-21	4	Vector	Identifies RLP Data as 1.31 Protocol PDU	VECTOR_ROOT_E131_DATA or VECTOR_ROOT_E131_EXTENDED
22-37	16	CID	Sender's CID	Sender's unique ID

5.1 Preamble Size

Sources shall set the Preamble Size to 0x0010. Receivers of UDP [UDP]-based E1.31 shall discard the packet if the Preamble Size is not 0x0010. The preamble contains the preamble size field, the post-amble size field, and the ACN packet identifier and has a length of 0x10 octets.

5.2 Post-amble Size

There is no post-amble for RLP over UDP [UDP]. Therefore, the Post-amble Size is 0x0.
Sources shall set the Post-amble Size to 0x0000. Receivers of UDP based E1.31 shall discard the packet if the Post-amble Size is not 0x0000.

5.3 ACN Packet Identifier

The ACN Packet Identifier shall contain the following sequence of hexadecimal characters 0x41 0x53 0x43 0x2d 0x45 0x31 0x2e 0x31 0x37 0x00 0x00 0x00.

Receivers shall discard the packet if the ACN Packet Identifier is not valid.

5.4 Flags & Length

The Root Layer's Flags & Length field is a 16-bit field with the PDU length encoded in the low 12 bits and 0x7 in the top 4 bits.

```
  7   6   5   4   3   2   1   0
+-------------------------------+
| flags=0x7     |  length-hi    |
+-------------------------------+
| length-lo                     |
+-------------------------------+
```

Figure 5-1: RLP Flags and Length

The RLP PDU length is computed starting with octet 16 and counting all remaining octets in the packet. In the case of an E1.31 Data Packet, this includes all of the octets from octet 16 through the last Property Value provided in the DMP layer (NOTE: For a full payload, this count would go through octet 637 and result in a total length of 638). For an E1.31 Synchronization Packet, which has no additional layers, the total length ends at the end of the E1.31 Framing Layer (octet 48, yielding a length of 49). E1.31 Universe Discovery Packet length is computed to the end of the List of Universes field.

5.5 Vector

Sources shall set the Root Layer's Vector to VECTOR_ROOT_E131_DATA if the packet contains E1.31 Data, or to VECTOR_ROOT_E131_EXTENDED if the packet is for Universe Discovery or for Synchronization. Receivers shall discard the packet if the received value is not VECTOR_ROOT_E131_DATA or VECTOR_ROOT_E131_EXTENDED. These values indicate that the root layer PDU is wrapping a specific E1.31 Framing Layer PDU.

5.6 CID (Component Identifier)

The Root Layer contains a CID. The CID shall be a UUID (Universally Unique Identifier) [UUID] that is a 128-bit number that is unique across space and time, compliant with RFC 4122 [UUID]. Each piece of equipment should maintain the same CID for its entire lifetime (e.g. by storing it in read-only memory). This means that a particular component on the network can be identified as the same entity from day to day despite network interruptions, power down, or other disruptions. However, in some systems there may be situations in which volatile components are dynamically created “on the fly” and, in these cases, the controlling process can generate CIDs as required. The choice of UUIDs for CIDs allows them to be generated as required without reference to any registration process or authority. The CID shall be transmitted in network byte order (big endian).

6. E1.31 Framing Layer Protocol

The contents of the E1.31 Framing Layer differ, based on what type of E1.31 information is being transported.

6.1 Flags & Length

Each distinct type of Framing Layer still starts with Flags & Length. The E1.31 Flags & Length field is a 16-bit field with the PDU length encoded in the low 12 bits and 0x7 in the top 4 bits.

```
  7   6   5   4   3   2   1   0
+-------------------------------+
| flags=0x7     |  length-hi    |
+-------------------------------+
| length-lo                     |
+-------------------------------+
```

Figure 6-1: E1.31 Flags and Length

The E1.31 Framing Layer PDU length is computed starting with octet 38 and continuing through the last octet provided by the underlying layer (none, if the E1.31 Framing Layer is followed by no additional payload, as is the case with E1.31 Synchronization Packets). Thus, an E1.31 Data Packet with full payload would have length 638, and an E1.31 Universe Discovery Packet would have length between 120 and 1144, depending on the List of Universes.

6.2 E1.31 Data Packet Framing Layer

Table 6-1: E1.31 Data Packet Framing Layer

38-39	2	Flags and Length	Protocol flags and length	Low 12 bits = PDU length High 4 bits = 0x7
40-43	4	Vector	Identifies 1.31 data as DMP Protocol PDU	VECTOR_E131_DATA_PACKET (DMX512-A [DMX] data)
44-107	64	Source Name	User Assigned Name of Source	UTF-8 [UTF-8] encoded string, null-terminated
108	1	Priority	Data priority if multiple sources	0-200, default of 100
109-110	2	Synchronization Address	Universe address on which sync packets will be sent	Universe on which synchronization packets are transmitted
111	1	Sequence Number	Sequence Number	To detect duplicate or out of order packets
112	1	Options	Options Flags	Bit 7 = Preview_Data Bit 6 = Stream_Terminated Bit 5 = Force_Synchronization
113-114	2	Universe	Universe Number	Identifier for a distinct stream of DMX512-A [DMX] Data

6.2.1 E1.31 Data Packet: Vector

Sources sending an E1.31 Data Packet shall set the E1.31 Layer's Vector to VECTOR_E131_DATA_PACKET. This value indicates that the E1.31 framing layer is wrapping a DMP PDU.

6.2.2 E1.31 Data Packet: Source Name

A user-assigned name provided by the source of the packet for use in displaying the identity of a source to a user. There is no mechanism, other than user configuration, to ensure uniqueness of this name. The source name shall be null-terminated. If the source component implements ACN discovery as defined in EPI 19 [ACN], then this name shall be the same as the UACN field specified in EPI 19 [ACN]. User-Assigned Component Names, as the title suggests, supply a single name for an entire component, so this Source Name field will exist for each unique CID, but may be the same across multiple universes sourced by the same component.

6.2.3 E1.31 Data Packet: Priority

As in [DMX] systems, the most recent E1.31 Data Packet from a single source supersedes any previous packets from that source (for a more thorough examination of sequence numbers, see Section 6.7.2). However, a receiver conforming to this standard may receive data for the same universe from multiple sources, as distinguished by examining the CID in the packet. This is a situation that cannot occur in conventional [DMX] systems. The Priority field is an unsigned, one octet field. The value is used by receivers in selecting between multiple sources of data for a given universe number. Sources that do not support variable priority shall transmit a priority of 100. No priority outside the range of 0 to 200 shall be transmitted on the network. Priority increases with numerical value, e.g., 200 is a higher priority than 100.

For a given universe number, an E1.31 receiver gathering data from multiple sources shall treat data from packets with the highest priority as the definitive data for that universe. The behavior for an E1.31 receiver also doing universe synchronization is undefined, and is beyond the scope of this standard.

6.2.3.1 Multiple Sources at Highest Priority

It is possible for there to be multiple sources, all transmitting data at the highest currently active priority for a given universe. When this occurs, receivers must handle these sources in some way.

A receiver which is only capable of processing one source of data will encounter a *sources exceeded* condition when two or more sources are present.

Many devices are capable of combining, merging or arbitrating between the candidate sources by some algorithm (see below), but such algorithms frequently limit the number of concurrent sources which can be handled due to resource limitations, or encounter situations where there are still multiple candidate sources meeting some specified condition, and then, once again, a *sources exceeded* condition arises which requires resolution.

6.2.3.2 Note on Merge and Arbitration Algorithms

A process of combining data from multiple sources to produce a definitive result is called a merge. A process which selects between candidate sources based on some additional selection criterion is called arbitration.

The single most common merging algorithm, which is usually appropriate to lighting intensity data (e.g. dimmer inputs), is to take the highest (numerically largest) level present from any of the candidate sources slot by slot throughout the universe—Highest Takes Precedence (HTP) merging. A variation of this uses DMX512-A START Code DDh (see [DMX] and http://tsp.esta.org/tsp/working_groups/CP/DMXAlternateCodes.php) to indicate slot-by-slot priority before merging the highest priority data for each slot on an HTP basis.

For other devices such as movement axes in automated luminaires, HTP is often highly inappropriate. In this case, it is common to accept only one candidate source, but arbitration criteria may be applied e.g. based on information in the Source Name field.

6.2.3.3 Note on Resolution of Sources Exceeded Condition

Resolution is required when the number of sources exceeds limitations of the algorithm or of resources available. In the simplest case with no merging or arbitration, this occurs when there is more than one source (at highest active priority).

One resolution mechanism is to stop accepting data from any source. Other mechanisms may choose one or more from the candidate sources by some overload selection scheme.

Designers are very strongly discouraged from implementing resolution algorithms that generate different results from the same source combination on different occasions, because this can make *sources exceeded* conditions hard to detect, makes networks very hard to troubleshoot and may cause unexpected results at critical times. For example, an arbitration scheme which accepts the first source detected at the active highest priority and rejects any subsequent ones is not recommended, as it will produce results dependent on the order of equipment startup and the vagaries of packet timing.

6.2.3.4 Requirements for Merging and Arbitrating

The ability of devices to merge or arbitrate between multiple sources at highest active priority shall be declared in user documentation for the device.

If merging or arbitration is implemented, the maximum number of sources which can be correctly handled shall be declared in user documentation for the device.

If merging or arbitration is implemented the algorithm used shall be declared in user documentation for the device.

6.2.3.5 Requirements for *Sources Exceeded* Resolution

The resolution behavior of equipment under *sources exceeded* conditions shall be declared in user documentation for the device.

Receiving devices conforming are strongly recommended to indicate a *sources exceeded* condition by some means easily detected at the device, e.g., by a flashing indicator, or obvious status message.

Receiving devices may additionally indicate a *sources exceeded* condition by other means such as remote indication initiated by a network message. This is particularly appropriate for devices which may be hard to access.

6.2.3.6 Requirements for Devices with Multiple Operating Modes

Receiving devices which have multiple configurations available to select between different methods for merging and/or *Sources Exceeded* resolution, shall meet the rules above for each configuration separately. Any configurations in which the device is not compliant with this standard should be clearly declared as such, but are otherwise beyond the scope of this specification.

6.2.4 E1.31 Data Packet: Synchronization Address

The Synchronization Address identifies a universe number to be used for universe synchronization.

6.2.4.1 Synchronization Address Usage in an E1.31 Data Packet

E1.31 Synchronization Packets occur on specific universes. Upon receipt, they indicate that any data advertising that universe as its Synchronization Address must be acted upon.

In an E1.31 Data Packet, a value of 0 in the Synchronization Address indicates that the universe data is not synchronized. If a receiver is presented with an E1.31 Data Packet containing a Synchronization Address of 0, it shall discard any data waiting to be processed and immediately act on that Data Packet.

When receiving an E1.31 Data Packet with a nonzero Synchronization Address, any receiver which does not support universe synchronization shall ignore the Synchronization Address and process the received data stream normally.

If the Synchronization Address field is not 0, and the receiver is receiving an active synchronization stream for that Synchronization Address, it shall hold that E1.31 Data Packet until the arrival of the appropriate E1.31 Synchronization Packet before acting on it.

A receiver that supports universe synchronization must not attempt to synchronize any data on a Synchronization Address until it has received its first E1.31 Synchronization Packet containing that address.

6.2.5 E1.31 Data Packet: Sequence Number

In a routed network environment it is possible for packets to be received in a different order to the one in which they were sent. The sequence number allows receivers or diagnostic equipment to detect out of sequence or lost packets.

Sources shall maintain a sequence for each universe they transmit. The sequence number for a universe shall be incremented by one for every packet sent on that universe. Where this increment would exceed the maximum allowed value, the sequence number shall wrap around to zero. There is no implied relationship between the sequence number of an E1.31 Synchronization Packet and the sequence number of an E1.31 Data Packet on that same universe.

6.2.6 E1.31 Data Packet: Options

This bit-oriented field is used to encode optional flags that control how the packet is used.

Preview_Data: Bit 7 (most significant bit)

This bit, when set to 1, indicates that the data in this packet is intended for use in visualization or media server preview applications and shall not be used to generate live output.

Stream_Terminated: Bit 6

This bit is intended to allow E1.31 sources to terminate transmission of a stream or of universe synchronization without waiting for a timeout to occur, and to indicate to receivers that such termination is not a fault condition.

When set to 1 in an E1.31 Data Packet, this bit indicates that the source of the data for the universe specified in this packet has terminated transmission of that universe. Three packets containing this bit set to 1 shall be sent by sources upon terminating sourcing of a universe. Upon receipt of a packet containing this bit set to a value of 1, a receiver shall enter network data loss condition. Any property values in an E1.31 Data Packet containing this bit shall be ignored.

Force_Synchronization: Bit 5

This bit indicates whether to lock or revert to an unsynchronized state when synchronization is lost (See Section 11 on Universe Synchronization and 11.1 for discussion on synchronization states). When set to 0, components that had been operating in a synchronized state shall not update with any new packets until synchronization resumes. When set to 1, once synchronization has been lost, components that had been operating in a synchronized state need not wait for a new E1.31 Synchronization Packet in order to update to the next E1.31 Data Packet.

Bits 0 through 4 of this field are reserved for future use and shall be transmitted as 0 and ignored by receivers.

6.2.7 E1.31 Data Packet: Universe

The Universe is a 16-bit field that defines the universe number of the data carried in the packet. Universe values shall be limited to the range 1 to 63999. Universe value 0 and those between 64000 and 65535 are reserved for future use. E131_DISCOVERY_UNIVERSE is the Universe Discovery universe. See Section 9 for more information.

6.3 E1.31 Synchronization Packet Framing Layer

Table 6-2: E1.31 Synchronization Packet Framing Layer

38-39	2	Flags and Length	Protocol flags and length	Low 12 bits = PDU length High 4 bits = 0x7
40-43	4	Vector	Identifies 1.31 data as Synchronization information	VECTOR_E131_EXTENDED_SYNCHRONIZATION (universe synchronization)
44	1	Sequence Number	Sequence Number	To detect duplicate or out of order packets
45-46	2	Synchronization Address	Universe Number	Universe on which synchronization packets are transmitted
47-48	2	Reserved		Reserved (See Section 6.3.4)

6.3.1 E1.31 Synchronization Packet: Vector

Sources sending an E1.31 Synchronization Packet shall set the E1.31 Layer's Vector to VECTOR_E131_EXTENDED_SYNCHRONIZATION. This value indicates that the E1.31 Framing Layer contains universe synchronization information.

6.3.2 E1.31 Synchronization Packet: Sequence Number

In a routed network environment, it is possible for packets to be received in a different order to the one in which they were sent. The sequence number allows receivers or diagnostic equipment to detect out of sequence or lost packets.

Sources shall maintain a sequence for each universe they transmit. The sequence number for a universe shall be incremented by one for every packet sent on that universe. Where this increment would exceed the maximum allowed value, the sequence number shall wrap around to zero. There is no implied relationship between the sequence number of an E1.31 Synchronization Packet and the sequence number of an E1.31 Data Packet on that same universe.

6.3.3 E1.31 Synchronization Packet: Synchronization Address

The Synchronization Address identifies the universe to which this synchronization packet is directed.

6.3.3.1 Synchronization Address Usage in an E1.31 Synchronization Packet

An E1.31 Synchronization Packet is sent to synchronize the E1.31 data on a specific universe number. A Synchronization Address of 0 is thus meaningless, and shall not be transmitted. Receivers shall ignore E1.31 Synchronization Packets containing a Synchronization Address of 0.

When E1.31 Synchronization Packets are sent via multicast, they shall only be sent to the address which corresponds to their Synchronization Address. Receivers may ignore Synchronization Packets sent to multicast addresses which do not correspond to their Synchronization Address. More information about the correlation between universe numbers and multicast addresses can be found in Section 9.

6.3.4 E1.31 Synchronization Packet: Reserved

Octets 47-48 of the E1.31 Synchronization Packet are reserved for future use. They shall be transmitted as 0 and ignored by receivers.

6.4 E1.31 Universe Discovery Packet Framing Layer

Table 6-3: E1.31 Universe Discovery Packet Framing Layer

38-39	2	Flags and Length	Protocol flags and length	Low 12 bits = PDU length High 4 bits = 0x7
40-43	4	Vector	Identifies 1.31 data as Universe Discovery Data	VECTOR_E131_EXTENDED_DISCOVERY (universe discovery)
44-107	64	Source Name	User Assigned Name of Source	UTF-8 [UTF-8] encoded string, null-terminated
108-111	4	Reserved		

6.4.1 E1.31 Universe Discovery Packet: Vector

Sources sending E1.31 Universe Discovery Packets shall set the E1.31 Layer's Vector to VECTOR_E131_EXTENDED_DISCOVERY. This value indicates that the E1.31 framing layer is wrapping a Universe Discovery PDU.

6.4.2 E1.31 Universe Discovery Packet: Source Name

A user-assigned name provided by the source of the packet for use in displaying the identity of a source to a user. There is no mechanism, other than user configuration, to ensure uniqueness of this name. The source name shall be null-terminated. If the source component implements ACN discovery as defined in EPI 19 [ACN], then this name shall be the same as the UACN field specified in EPI 19 [ACN]. User-Assigned Component Names, as the title suggests, supply a single name for an entire component, so this Source Name field will exist for each unique CID, but may be the same across multiple universes sourced by the same component.

6.4.3 E1.31 Universe Discovery Packet: Reserved

Octets 108-111 of the E1.31 Universe Discovery Packet are reserved for future use. They shall be transmitted as 0 and ignored by receivers.

6.5 Processing by Receivers

There are multiple vectors that may be used in the Framing Layer to indicate E1.31 packets. Receivers shall discard the packet if the received value is not VECTOR_E131_DATA_PACKET, VECTOR_E131_EXTENDED_SYNCHRONIZATION, or VECTOR_E131_EXTENDED_DISCOVERY. Receivers that do not support universe synchronization may ignore packets arriving with vector VECTOR_E131_EXTENDED_SYNCHRONIZATION.

6.6 Framing Layer Operation and Timing - Source Requirements

6.6.1 Transmission Rate

E1.31 sources shall not transmit packets for a given universe number at a rate which exceeds the maximum refresh rate specified in E1.11 [DMX] unless configured by the user to do so.

For a given universe that contains only E1.31 devices—a universe with no devices that convert from E1.31 to DMX512-A—an E1.31 source may be configured to observe higher refresh rates than those specified in the [DMX] standard or above. In order to maintain compatibility with non-E1.31 components, the E1.31 source shall provide a user-configurable option to enable or disable these higher refresh rates.

Note that E1.11 [DMX] places special restrictions on the maximum rate for alternate START Code packets in Section 8.5.3.2 of that document.

6.6.2 Null START Code Transmission Requirements in E1.31 Data Packets

In order to avoid unnecessary use of network bandwidth, transmission of redundant Null START Code data is minimized. For a given universe number, transmitting devices shall transmit Null START Code data only when that data changes, with the following exceptions:

1. Three packets containing the non-changing Property Values (corresponding to DMX512-A slot data) shall be sent before the initiation of transmission suppression.

2. Thereafter, a single keep-alive packet shall be transmitted at intervals of between 800mS and 1000mS. Each keep-alive packet shall have identical content to the last Null START Code data packet sent with the exception that the sequence number shall continue to increment normally.

These requirements do not apply to alternate START Code data.

6.7 Framing Layer Operation and Timing - Receiver Requirements

6.7.1 Network Data Loss

Network data loss is a condition that is defined as either the absence of reception of E1.31 Data Packets from a given source for a period of E131_NETWORK_DATA_LOSS_TIMEOUT or the receipt of a packet containing the Options field, bit 6 set to value of 1 (see Section 6.2.6 Options). Data loss is specific to a universe (see the Universe field in Table 4-1), so data loss may exist for one universe from a source and not for other universes provided by that same source.

When a data loss condition arises, a source specific universe is considered disconnected.

6.7.1.1 Network Data Loss and Universe Discovery

Receivers should be aware that due to jitter or packet loss, pages in a Universe Discovery List of Universes may be dropped or arrive out of order, potentially even mixed in between different runs of pages. The manner in which Receivers respond to jitter or network data loss is outside of the scope of this standard..

6.7.2 Sequence Numbering

Receivers that do not support sequence numbering of packets shall ignore the contents of these fields in the E1.31 Data Packet and the E1.31 Synchronization Packet.

Receivers that do support sequence numbering of packets shall evaluate sequence numbers separately for each E1.31 Packet type, and within each E.131 Packet type, the Receiver shall evaluate sequence numbers separately for each Universe. The Receiver shall then process packets in the order received unless they are discarded according to the algorithm below.

Having first received a packet with sequence number A, a second packet with sequence number B arrives. If, using signed 8-bit binary arithmetic, B - A is less than or equal to 0, but greater than -20, then the packet containing sequence number B shall be deemed out of sequence and discarded.

> Note: This algorithm allows the sequence stream from a source to jump by large amounts without undue delay, as in the case of a reset, without allowing packets received slightly out of order to cause flicker or interfere with predictive algorithms found in many moving light fixtures.

7. DMP Layer Protocol

DMP data only appears in E1.31 Data Packets and shall not be included in E1.31 Synchronization Packets or E1.31 Universe Discovery Packets.

In DMP terms, the DMX512-A packet is treated at the DMP Layer as a Set Property message for an array of up to 513 one-octet virtually addressed properties. A restriction of E1.31 is that the array shall always begin at property zero corresponding in [DMX] nomenclature to START Code. This allows E1.31 implementations which do not process DMP to treat much of the DMP header content as fixed values.

Table 7-1: E1.31 Data Packet DMP Layer

115-116	2	Flags and Length	Protocol flags and length	Low 12 bits = PDU length High 4 bits = 0x7
117	1	Vector	Identifies DMP Set Property Message PDU	0x02
118	1	Address Type & Data Type	Identifies format of address and data	0xa1
119-120	2	First Property Address	Indicates DMX512-A START Code is at DMP address 0	0x0000
121-122	2	Address Increment	Indicates each property is 1 octet	0x0001
123-124	2	Property value count	Indicates 1+ the number of slots in packet	0x0001 -- 0x0201
125-637	1-513	Property values	DMX512-A START Code + data	START Code + Data

7.1 DMP Layer: Flags & Length

The DMP Layer's Flags & Length field is a 16-bit field with the PDU length encoded in the low 12 bits and 0x7 in the top 4 bits.

```
  7   6   5   4   3   2   1   0
+-------------------------------+
| flags=0x7     |  length-hi    |
+-------------------------------+
| length-lo                     |
+-------------------------------+
```

Figure 7-1: DMP Flags and Length

The DMP Layer PDU length is computed starting with octet 115 and continuing through the last property value provided in the DMP PDU (octet 637 for a full payload). This is the length of the DMP PDU.

7.2 DMP Layer: Vector

The DMP Layer's Vector shall be set to VECTOR_DMP_SET_PROPERTY, which indicates a DMP Set Property message by sources. Receivers shall discard the packet if the received value is not VECTOR_DMP_SET_PROPERTY.

7.3 Address Type and Data Type

Sources shall set the DMP Layer's Address Type and Data Type to 0xa1. Receivers shall discard the packet if the received value is not 0xa1.

7.4 First Property Address

Sources shall set the DMP Layer's First Property Address to 0x0000. Receivers shall discard the packet if the received value is not 0x0000.

7.5 Address Increment

Sources shall set the DMP Layer's Address Increment to 0x0001. Receivers shall discard the packet if the received value is not 0x0001.

7.6 Property Value Count

The DMP Layer's Property Value Count is used to encode the number of DMX512-A [DMX] Slots (including the START Code slot).

7.7 Property Values (DMX512-A Data)

The DMP Layer's Property values field is used to encode the DMX512-A [DMX] START Code and data.

The first octet of the property values field shall be the DMX512-A [DMX] START Code.

The remainder of the property values shall be the DMX512-A data slots. This data shall contain a sequence of octet data values that represent a consecutive group of data slots, starting with slot 1, from a DMX512-A packet. The number of data slots encoded in the frame shall not exceed the DMX512-A limit of 512 data slots.

Processing of Alternate START Code data shall be compliant with ANSI E1.11 [DMX] Section 8.5.3.3 -Handling of Alternate START Code packets by in-line devices.

8. Universe Discovery Layer

Universe Discovery data only appears in E1.31 Universe Discovery Packets and shall not be included in E1.31 Data Packets or E1.31 Synchronization Packets.

Table 8-1: E1.31 Universe Discovery Packet Universe Discovery Layer

112-113	2	Flags and Length	Protocol flags and length	Low 12 bits = PDU length High 4 bits = 0x7
114-117	4	Vector	Identifies Universe Discovery data as universe list	VECTOR_UNIVERSE_DISCOVERY_UNIVERSE_LIST
118	1	Page	Packet Number	Identifier indicating which packet of N this is—pages start numbering at 0.
119	1	Last Page	Final Page	Page number of the final page to be transmitted.
120-1143	0-1024	List of Universes	Sorted list of up to 512 16-bit universes.	Universes upon which data is being transmitted.

8.1 Flags and Length

The Universe Discovery Layer's Flags & Length field is a 16-bit field with the PDU length encoded in the low 12 bits and 0x7 in the top 4 bits.

```
  7   6   5   4   3   2   1   0
+-------------------------------+
| flags=0x7     |  length-hi    |
+-------------------------------+
| length-lo                     |
+-------------------------------+
```

Figure 8-1: Universe Discovery Flags and Length

The Universe Discovery Layer PDU length is computed starting with octet 112 and continuing through the last universe provided in the Universe Discovery PDU (octet 1143 for a full payload). This is the length of the Universe Discovery PDU.

8.2 Universe Discovery Layer: Vector

The Universe Discovery Layer's Vector shall be set to VECTOR_UNIVERSE_DISCOVERY_UNIVERSE_LIST, indicating that it contains a list of universes. Receivers shall discard the packet if the received value is not VECTOR_UNIVERSE_DISCOVERY_UNIVERSE_LIST.

8.3 Page

A single source may be transmitting on so many universes that the total number of universes it must include in its List of Universes will span multiple packets. Each one of these packets acts as a “page” of those universes.

The Universe Discovery Layer's Page field is an 8-bit field indicating the page number of this E1.31 Universe Discovery Packet. Page numbers are indexed, starting at 0.

8.4 Last Page

The Universe Discovery Layer's Last Page field is an 8-bit field indicating the number of the final page to be transmitted. Together, these pages carry across the complete list of the universes upon which this source is actively transmitting. Page numbers are indexed starting at 0. Rather than not sending an E1.31 Universe

Discovery packet, sources that are not actively transmitting on any universes may choose to send an E1.31 Universe Discovery Packet with an empty List of Universes, a Page of 0, and a Last Page of 0.

8.5 List of Universes

The Universe Discovery Layer's List of Universes is a packed, numerically sorted list of 16-bit universe addresses. It may be empty if a source is not transmitting on any universes. Otherwise, it shall enumerate all of the universes upon which a source is actively transmitting E1.31 Data or Synchronization information.

9. Operation of E1.31 in IPv4 and IPv6 Networks

This standard has the ability to operate over both IPv4 and IPv6 transports. Components complying with this standard shall indicate, through labelling, manufacturer documentation, or other means, which IP transports they support.

9.1 Association of Multicast Addresses and Universe

This standard uses network multicast addressing in order to direct DMX512-A universes to their specified destination. Universes may be sent directly between network devices using unicast addressing, however no discovery mechanism is specified in this standard to support the dynamic determination of such addresses.

Addressing and partitioning of multicast traffic is achieved by setting the least significant two bytes of the multicast IP address, in both IPv4 and IPv6 transports, to the desired universe number or Synchronization Address. This is described in detail in Sections 9.3.1 and 9.3.2. Sources configured to operate simultaneously in IPv4 and IPv6 shall transmit identical E1.31 Packets regardless of the IP transport used. Receivers configured to operate simultaneously in IPv4 and IPv6 shall not process E1.31 Packets differently based on their IP transport. Receivers seeing the same E1.31 Packet via both IP transports shall only act on one instance of that packet.

9.1.1 Multicast Addressing

The top 256 IPv4 multicast addresses, 239.255.255.0 through 239.255.255.255, are reserved by IANA for scope relative addressing; therefore E1.31 devices shall not transmit on these top 256 addresses. If the corresponding reserved universes are used in the future, they must be transmitted unicast. Additionally, E1.31 devices shall not use universe number 0 as it is reserved for future use. Universe numbers between 64000 and 65535, excluding universe 64214 (which is used for E1.31 Universe Discovery), are reserved for future use and shall not be used by E1.31 devices except as specified in future standards designed to coexist with E1.31 when use of these universes is called for specifically.

Sources can thus transmit data to the required universes without prior knowledge of the network topology. Equally, responders can listen to a predetermined IP address for data representing a specific universe.

An operating mode shall be provided where E1.31 sources shall transmit universes or Synchronization Addresses on the multicast address as defined in Table 4-2. Note: The identity of the universe shall be determined by the universe number in the packet and not assumed from the multicast address. An E1.31 receiver shall also respond to 1.31 data received on its unicast address.

When multicast addressing is used, the UDP destination Port shall be set to the standard ACN-SDT multicast port ACN_SDT_MULTICAST_PORT. For unicast communication the ACN-SDT multicast port shall be used by default, but methods for configuration and use of alternative ports may be provided.

9.2 Multicast Subscription

Receivers supporting IPv4 shall support IGMP V2 [IGMP2] (NOTE: This would include following IGMP V3 [IGMP3] or any subsequent superset of [IGMPv2]'s functionality). Receivers supporting IPv6 shall support MLD V1 [MLD1] (NOTE: This would include following MLD V2 [MDL2] or any subsequent superset of [MLD1]'s functionality). IGMP/MLD are used to communicate multicast address usage to network infrastructure.

9.3 Allocation of Multicast Addresses

9.3.1 Allocation of IPv4 Multicast Addresses

Multicast addresses are from the IPv4 Local Scope and will be managed by routers in conformance with RFC 2365 [ASIPM].

The multicast IP address is defined in Table 9-1 below:

Table 9-1: IPv4 Universe - IP mapping

IP Address Byte	Value
1	239
2	255
3	Universe/Synchronization Address – Hi byte
4	Universe/Synchronization Address – Lo byte

9.3.2 Allocation of IPv6 Multicast Addresses

Multicast addresses are specified as follows:

Table 9-2: IPv6 Multicast Address Format

<table>
<tr><td>Prefix (1 Byte)</td><td colspan="4">Flags (4 bits)</td><td>Scope (4 bits)</td><td>Group ID (112 bits)</td></tr>
<tr><td>0xFF</td><td colspan="4">0x1</td><td>0x8</td><td>00::83:00:UH:UL</td></tr>
<tr><td></td><td colspan="4">Flags Bit-Level Detail</td><td></td><td></td></tr>
<tr><td>Flag</td><td>0</td><td>R</td><td>P</td><td>T</td><td></td><td></td></tr>
<tr><td>Value</td><td>0</td><td>0</td><td>0</td><td>1</td><td></td><td></td></tr>
</table>

The multicast address prefix is fixed at 0xFF.

The flags are set as follows:

0 - The high-order flag bit is reserved and shall be set to 0.

R – The R flag is defined by RFC 3956 [EmbeddedRP] and shall be set to 0.

P – The P flag is defined by RFC 3306 [UnicastPrefixMcast] and shall be set to 0.

T – The T flag indicates a “transient” multicast address that is not permanently assigned by IANA. This bit shall be set to 1 to indicate a non-permanently assigned multicast address.

The scope shall be set to 0x8 to indicate Organization-Local scope per RFC 4291 [IPv6Addr] and RFC 7346 [IPv6McastScope].

The Group ID shall be set to 00::83:00:UH:UL, where UH is the High Byte and UL is the Low Byte of the Universe or Synchronization Address.

Therefore, the final IPv6 Multicast address range, when written according to RFC 5952 [IPv6AddrText], shall be between FF18::83:00:00:00 through FF18::83:00:FA:00.

Stated another way, the multicast IP address is defined in Table 9-3 below:

Table 9-3: IPv6 Universe - IP mapping

IP Address Byte	Value
1	0xFF
2	0x18
3	0x00
4	0x00
5	0x00
6	0x00
7	0x00
8	0x00

9	0x00
10	0x00
11	0x00
12	0x00
13	0x83
14	0x00 (Reserved)
15	Universe/Synchronization Address – Hi byte
16	Universe/Synchronization Address – Lo byte

9.4 IPv4 and IPv6 Support Requirements

To maintain backwards compatibility with E1.31 Receivers implementing previous revisions of this standard, E1.31 Sources shall be able to operate on both IPv4 and IPv6 networks, possibly simultaneously. The state of IPv4 and IPv6 operation on an E1.31 Source shall be configurable by the end user.

To maintain backwards compatibility with E1.31 Sources implementing previous revisions of this standard, E1.31 Receivers shall support IPv4 operation, and may optionally support IPv6 operation. The state of IPv4 and IPv6 operation on an E1.31 Receiver may be configurable by the end user.

10. Translation between DMX512-A and E1.31 Data Transmission

This section pertains only to the E1.31 Data Packet and shall not be used in reference to the E1.31 Synchronization Packet or the E1.31 Universe Discovery Packet.

10.1 DMX512-A to E1.31 Translation

Devices performing translation of incoming DMX512-A [DMX] data to E1.31 network data are subject to the requirements of this Section.

10.1.1 Boot Condition

A DMX512-A [DMX] to E1.31 translator shall not transmit E1.31 Data Packets for a given universe until it has received at least one valid (properly formed) DMX512-A [DMX] input packet for that universe.

10.1.2 Temporal Sequence

A DMX512-A [DMX] to E1.31 translator shall transmit packets in the order in which they are received from the DMX512-A [DMX] source.

10.1.3 Loss of Data

Upon detection of loss of data as defined in DMX512-A [DMX], a source shall terminate transmission in accordance with Section 6.7.1.

10.2 E1.31 to DMX512-A Translation

10.2.1 General

Devices performing translation of incoming E1.31 network data to DMX512-A [DMX] data are subject to the requirements of this Section.

10.2.2 Loss of Data

An operating mode shall be provided, whereupon detection of loss of data, as defined in Section 6.7.1, for all sources of a universe, a source shall immediately stop transmitting DMX512-A [DMX] packets. In addition, a source may supplement this required mode with alternative operating modes, for example, such as those implementing a hold-last-look feature by continuously retransmitting the last valid packet.

11. Universe Synchronization

This standard allows a transport mechanism to synchronize streamed packets of data from single sources to multiple receivers over the ACN network.

Universe synchronization is required for applications where receivers require more than one universe to be controlled, multiple receivers produce synchronized output, or unsynchronized control of receivers may result in undesired visual effects.

Typical receivers which require universe synchronization include media servers, LED panels, and any fixtures with fast-reacting dimming devices.

All configuration for universe synchronization shall be carried out on the source side.

There is no restriction on the number of synchronization addresses allowed on a single network. It is possible to have multiple independent universes configured for E1.31 synchronization existing concurrently.

11.1 Synchronized and Unsynchronized Data

Any receiver that supports synchronization on a specific universe will be either in a “Synchronized” or an “Unsynchronized” state. A receiver is said to be “Synchronized” when it is actively receiving E1.31 Synchronization Packets within the minimum refresh window for DMX512-A [DMX] packets.

11.1.1 When to Begin Synchronizing Data

An “Unsynchronized” receiver shall start performing universe synchronization upon receipt of the first E1.31 Synchronization Packet for that universe.

11.1.2 When to Stop Synchronizing Data

A “Synchronized” receiver shall also stop performing universe synchronization if it does not receive an E1.31 Synchronization Packet on that universe within E131_NETWORK_DATA_LOSS_TIMEOUT. The behavior of the synchronized receivers may be determined by the Force Synchronization Option bit of the E1.31 Data Packet (see Section 6.2.6).

11.2 Synchronization Timings in a Streaming Environment

11.2.1 Arrival of Multiple Packets Before Processing

It is possible that multiple E1.31 Data Packets may arrive in the window of time between E1.31 Synchronization Packets. As E1.31 is a streaming protocol, no guaranteed delivery should be expected. Though the interim data may be lost, an E1.31 receiver shall only synchronize using the definitive E1.31 data for that universe, as determined by the rules outlined in Section 6.2.3. For a single source, this would be the E1.31 Data Packet on that universe with the most recent valid sequence number (see Section 6.7.2). The presence of multiple active synchronization sources on the same synchronization address is beyond the scope of this standard, and may cause unpredictable behavior.

11.2.2 Delays Before Universe Synchronization

It may be advisable for the source to introduce a configurable delay between data packets and the subsequent transmission of the E1.31 Synchronization Packet. In environments with high latency or low bandwidth, this may mitigate some of the perceptible jitter in the system, as it allows receivers time to accept and process all of the data before having to synchronize it. Though this delay of configurable duration is not mandated by this standard, it is recommended.

12. Universe Discovery

E1.31 Universe Discovery enables other components on a network to know which universes are being used to transmit data or synchronization information. Though this specification requires any source intending to comply with E1.31 to implement Universe Discovery, it should be noted that some legacy sources may not support it, meaning that a list of universes cannot ever be guaranteed to be complete.

Universe Discovery is specifically intended to reduce the imposed load on a network that would otherwise be created by a monitoring system joining every single E1.31 multicast group in order to probe its traffic to report this same information.

12.1 Universe Discovery and Termination

Any source that is no longer sending any universe data may stop sending E1.31 Universe Discovery Packets until such time that it resumes transmission of E1.31 Data and/or Synchronization information. It shall also be acceptable for such a source to continue sending E1.31 Universe Discovery Packets that contain an empty List of Universes.

12.2 Termination of Stream Transmission

An E1.31 Data stream has been terminated when either a packet with the Stream_Terminated option bit set is sent, or when E131_NETWORK_DATA_LOSS_TIMEOUT has passed with no data sent.

Sources having terminated transmission on an E1.31 universe need not immediately reflect that change to their List of Universes and may instead wait to report it until no later than the second E131_UNIVERSE_DISCOVERY_INTERVAL has transpired.

Appendix A: Defined Parameters (Normative)

VECTOR_ROOT_E131_DATA	0x00000004	
VECTOR_ROOT_E131_EXTENDED	0x00000008	
VECTOR_DMP_SET_PROPERTY	0x02	(Informative)
VECTOR_E131_DATA_PACKET	0x00000002	
VECTOR_E131_EXTENDED_SYNCHRONIZATION	0x00000001	
VECTOR_E131_EXTENDED_DISCOVERY	0x00000002	
VECTOR_UNIVERSE_DISCOVERY_UNIVERSE_LIST	0x00000001	
E131_E131_UNIVERSE_DISCOVERY_INTERVAL	10 seconds	
E131_NETWORK_DATA_LOSS_TIMEOUT	2.5 seconds	
E131_DISCOVERY_UNIVERSE	64214	
ACN_SDT_MULTICAST_PORT	5568	

Appendix B: An Example of Universe Synchronization For Implementors (Informative)

B.1 Universe Synchronization for Sources

Source A is set to use universe 7962 as its synchronization universe.

Each E1.31 Data Packet that Source A sends that need to be synchronized must be sent with the Synchronization Address set to that universe, no matter what universe the data is intended for. For each E1.31 Data Packet, there is the option of setting the Force_Synchronization bit (See Section 6.2.6) Source A wants to lock and wait for the synchronization packet, so it will set this to 0.

A sample packet may look like this:

Table B-1: Universe Synchronization Example E1.31 Data Packet

Octet	Field Size	Field Name	Field Description	Field Contents
Root Layer				
0-1	2	Preamble Size	Define RLP Preamble Size.	0x0010
2-3	2	Post-amble Size	RLP Post-amble Size.	0x0000
4-15	12	ACN Packet Identifier	Identifies this packet as E1.17	0x41 0x53 0x43 0x2d 0x45 0x31 0x2e 0x31 0x37 0x00 0x00 0x00
16-17	2	Flags and Length	Protocol flags and length	0x72 0x6e
18-21	4	Vector	Identifies RLP Data as 1.31 Protocol PDU	0x00000004
22-37	16	CID	Sender's CID	0xef 0x07 0xc8 0xdd 0x00 0x64 0x44 0x01 0xa3 0xa2 0x45 0x9e 0xf8 0xe6 0x14 0x3e *(example only)*
E1.31 Framing Layer				
38-39	2	Flags and Length	Protocol flags and length	0x72 0x58
40-43	4	Vector	Identifies 1.31 data as DMP Protocol PDU	0x00000002
44-107	64	Source Name	User Assigned Name of Source	Source_A *(example only)*
108	1	Priority	Data priority if multiple sources	100 *(example only)*
109-110	2	Synchronization Address	Universe address on which sync packets will be sent	7962 *(example only)*
111	1	Sequence Number	Sequence Number	154 *(example only)*
112	1	Options	Options Flags	Bit 5 = 0 *(example only)*
113-114	2	Universe	Universe Number	1 *(example only)*
DMP Layer				
115-116	2	Flags and Length	Protocol flags and length	0x72 0x0b
117	1	Vector	Identifies DMP Set Property Message PDU	0x02
118	1	Address Type & Data Type	Identifies format of address and data	0xa1
119-120	2	First Property Address	Indicates DMX512-A START Code is at DMP address 0	0x0000
121-122	2	Address Increment	Indicates each property is 1 octet	0x0001
123-124	2	Property value count	Indicates 1+ the number of slots in packet	0x0201
125-637	1-513	Property values	DMX512-A START Code + data	0x00 + Exciting Data

Once Source A has sent all of the E1.31 Data Packets to the receivers that need to be synchronized in a given interval, those E1.31 Data Packets must be followed by an E1.31 Synchronization Packet to universe 7962 that looks like the following:

Table B-2: Universe Synchronization Example E1.31 Synchronization Packet

Octet	Field Size	Field Name	Field Description	Field Contents
Root Layer (See Section 5)				
0-1	2	Preamble Size	Define RLP Preamble Size.	0x0010
2-3	2	Post-amble Size	RLP Post-amble Size.	0x0000
4-15	12	ACN Packet Identifier	Identifies this packet as E1.17	0x41 0x53 0x43 0x2d 0x45 0x31 0x2e 0x31 0x37 0x00 0x00 0x00
16-17	2	Flags and Length	Protocol flags and length	0x70 0x21
18-21	4	Vector	Identifies RLP Data as 1.31 Protocol PDU	0x00000008
22-37	16	CID	Sender's CID	0xef 0x07 0xc8 0xdd 0x00 0x64 0x44 0x01 0xa3 0xa2 0x45 0x9e 0xf8 0xe6 0x14 0x3e *(example only)*
E1.31 Framing Layer (See Section 6)				
38-39	2	Flags and Length	Protocol flags and length	0x70 0x0b
40-43	4	Vector	Identifies 1.31 data as Synchronization information	0x00000001
44	1	Sequence Number	Sequence Number	236 *(example only)*
45-46	2	Synchronization Address	Universe Number	7962 *(example only)*
47-48	2	Reserved		

Though the E1.31 Synchronization Packet can be sent at any time after its associated E1.31 Data Packets, it is recommended that the source observe a slight pause to allow for processing delays (See Section 11.2.2).

Receivers, upon seeing the E1.31 Synchronization Packet, will then act on their data.

This process can be repeated until a network interruption, or until Source A sets the Synchronization Address in its E1.31 Data Packets to 0. Because Source A chose 0 as its value for Force_Synchronization, it cannot simply stop sending E1.31 Data Packets and wait for the receiver to recognize E131_NETWORK_DATA_LOSS_TIMEOUT has elapsed.

B.2 Universe Synchronization for Receivers

Receiver B supports Universe Synchronization. As E1.31 Data Packets arrive, it examines them to see if the Synchronization Address is set to a nonzero value. At some point, it receives an E1.31 Data Packet with sequence number 154 from Source A that contains a Synchronization Address of 7962. The fifth bit in the Options field of this packet, representing Force_Synchronization is set to 0.

Receiver B then joins the correct multicast address for that universe number, 239.255.31.26, and listens for an E1.31 Synchronization Packet. Since Receiver B supports Universe Synchronization, instead of acting on the original E1.31 Data Packet right away, it will not take any further action until it sees an E1.31 Synchronization Packet on universe 7962. To help minimize latency, it is acceptable for Receiver B to do any necessary pre-processing on this E1.31 Data Packet, as long as that does not affect Receiver B’s output.

When the E1.31 Synchronization Packet arrives from Source A, Receiver B acts on the data. This process continues until Receiver B receives an E1.31 Data Packet with a Synchronization Address of 0. Receiver B may then unsubscribe from the synchronization multicast address.

If, at any time, Receiver B receives more than one E1.31 Data Packet with the same Synchronization Address in it, before receiving an E1.31 Universe Synchronization Packet, it will discard all but the most recent E1.31 Data Packet (for a refresher on sequence numbers, see Section 6.7.2). Those packets are only acted upon when the synchronization command arrives.

Since the the Force_Synchronization bit in the Options field of the E1.31 Data Packet has been set to 0, even if Source A times out the E131_NETWORK_DATA_LOSS_TIMEOUT, Receiver B will stay in its last look until a new E1.31 Synchronization Packet arrives.