DRAFT

# BSR E1.31-1 – 202x

## Per-slot Priority Extension for ANSI E1.31

CP/2024-1029r0a

**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 guaranties 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:

*[Insert the current Investors table here]*

**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

**Senior Technical Standards Manager**
Karl G. Ruling
ESTA
271 Cadman Plaza PO Box 23200
New York, NY 11202-3200
USA
+1-212-244-1505
karl.ruling@esta.org

**Technical Standards Council Co-Chairpersons**

Mike Garl
Mike Garl Consulting LLC
+1-865-389-4371
mike@mikegarlconsulting.com

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

Michael Lay
Candela Controls, Inc
+1-352-433-2479
michael_m_lay@yahoo.com

**Acknowledgments**
The Control Protocol Working Group members, when this document was approved by the working group on _____, are shown below.

**Voting members:**

**Observer (non-voting) 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**

## ESTA Preamble

### Introduction

#### The Need for This Standard

The sACN standard details DMX-style control over TCP/IP networks. While this provides a fast and efficient mechanism to transport the well-understood DMX protocol, it also introduces a complication not possible in the original DMX standard: multiple sources. In sACN, multiple sources have the ability to provide overlapping sets of universe data, and each source may only be interested in controlling certain subsections of different universes. sACN includes a priority field in each Data Packet, but this applies to the entire set of slots found in the encapsulated DMX data (Property values), and does not cover cases where a controller wants to explicitly control a small subsection of values without disturbing the rest. Using merge and arbitration techniques such as Highest Takes Precedence (HTP) doesn't help when that controller wants to move a subset of the values to a lower DMX level, or when those values do not represent intensity values.

#### Example Scenarios

Consider the following example scenarios, where the need for a Per-Slot Priority mechanism becomes clear:

- A moving light controller wishes to command the pan & tilt of a moving light, while another controller maintains full control over the color mixing and intensity of the same luminaire.
- A theatre has LED houselighting that uses additive color mixing. These are to be controlled by a lighting console during performances, and an architectural controller at all other times.
  The architectural controller requires the ability to specify a minimum white-light lighting level during normal venue operations, while the lighting console requires the ability to reduce the additive color channel values down from "white" in order to mix to the colors desired during performances.
  HTP does not permit the lighting controller to reduce the value of the additive color mixing slots, and raising the universe priority of the lighting console would defeat the minimum normal venue level set by the architectural controller.

#### History

To better handle scenarios, like the above, involving multiple control sources, ETC, Inc (https://www.etcconnect.com) originally obtained the 0xDD Alternate START Code to allow setting a priority per-slot, with the option of having DMX levels for a particular address ignored by a Receiver.

#### Evolution and Purpose of This Document

There are many manufacturers, beyond just the original ETC, who now support per-slot priority using the 0xDD Alternate START Code. With the publication of this standard, ETC will have officially returned that START Code to ESTA. The intent of this document is not to create a new per-slot priority technique, but, instead, to codify that already in use by the industry at the time preceding its publication.

#### 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.

## 1 Normative References

**[DMX] ANSI E1.11-2024**
Entertainment Technology - USITT DMX512-A Asynchronous Serial Digital Data Transmission Standard for Controlling Lighting Equipment and Accessories
This standard is maintained by the Entertainment Services and Technology Association.
ESTA
P.O. Box 23200
Brooklyn, NY 11202-3200
212-244-1505
https://tsp.esta.org
ESTA is an accredited ANSI Standards Developer.

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

**[sACN] ANSI E1.31 — 2018**
Entertainment Technology Lightweight streaming protocol for transport of DMX512 using ACN
This standard is maintained by ESTA.

**[UTF-8] UTF-8 The Unicode Standard, Version 6.3.0**
Mountain View, CA: The Unicode Consortium, 2013. ISBN 978-1-936213-08-5
Chapter 2, Section 2.5, Sub-Heading “UTF-8” defines the UTF-8 encoding.
This standard is maintained by The Unicode Consortium.

**[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

**[UUID] RFC 4122**
P. Leach, M. Mealing, and R. Salz. A universally Unique Identifier (UUID) URN Namespace. July 2005.

## 2 Informative References

**[ACN-DMP] ANSI E1.17 Architecture for Control Networks - Device Management Protocol**
This standard is maintained by ESTA.

## 3 Definitions

**octet.** An eight-bit byte.
**receiver.** (references [sACN], Section 3.6); A component that receives sACN.
**source.** (references [sACN], Section 3.5); A component that sends sACN. Sources are uniquely identified by their CID ([sACN], Section 5.6).
**slot**. (references [DMX], Section 3.24); 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.
**DMX level(s).** (references [sACN], Section 7.7); Data slots from the property values associated with a NULL START Code Data Packet.
**HTP.** "Highest Takes Precedence." A merge and arbitration algorithm whereby the source providing the highest level (largest numerical value) will have control of that slot.
**packet priority field.** (references [sACN], Section 6.2.3); Value of Priority field in an sACN Data Packet.
**packet options field.** (references [sACN], Section 6.2.6);Value of Options field in an sACN Data Packet.
**universe.** The concept of a collection of a maximum of 512 values, corresponding to a single DMX universe.
**address.** A single value in a universe, also known as a channel.
**universal hold last look time.** Employed by some implementers of DMX/sACN, a configurable time that must exceed E131_NETWORK_DATA_LOSS_TIMEOUT ([sACN], Appendix A), after which some Source Loss Behavior may take effect.
**source loss behavior.** Action taken by a receiver to affect a DMX level / Property value in the absence of a controlling source.
Examples of this behavior could be:

- Fade to 0
- Hold last known level for a preconfigured time (possibly forever)

- Fade to some predetermined level

**data packet.** (references [sACN] 4.1); An sACN data packet containing a [DMX] START Code followed by up to 512 slot values.
**NULL START Code packet.** An sACN data packet, where the property values starts with the NULL START Code ([DMX], Section 8.5.1), and each subsequent slot describes the desired dimmer class data ([DMX], Section 8.5.2) for that slot.
**Per-Slot Priority Packet.** An sACN data packet, where the property values starts with the Alternate START Code specified by ASC_PER_SLOT_PRIORITY and each subsequent slot describes the desired priority value for each corresponding slot in an sACN data packet containing NULL START Code data.

## 4 Per-Slot Priority

Devices may choose whether to support Per-Slot Priority. There is no requirement that any sACN device offer this feature. If a device includes support for Per-Slot Priority, this shall be stated in accompanying documentation, as per the requirements of [sACN] 6.2.3.4.

Receivers which do not support Per-Slot Priority shall ignore Data Packets containing the START Code ASC_PER_SLOT_PRIORITY.

Sources which do not support Per-Slot Priority shall not transmit any Data Packets using the START Code ASC_PER_SLOT_PRIORITY.

A Receiver may receive Data Packets from multiple Sources, some of which are sending Per-Slot Priority Packets and some of which are not. When arbitrating these data, the Receiver shall use the Priority field from the Data Packet Framing Layer for any Source deemed not to be sending Per-Slot Priority Packets.

Sources may send "Preview" and "Live" data streams at the same time, by setting the Preview_Data bit of the Options field in the Framing Layer ([sACN, Section 6.2.6). Receivers shall treat Live and Preview as separate Universes, using Preview Per-Channel Priority for the Preview NULL START Code.

### 4.1 Per-Slot Priority Packet

A Per-Slot Priority Packet is an sACN Data Packet (sACN, Section 4.1) that adheres to some additional constraints.

Table 4-1: Per-Slot Priority Packet

| Octet | Field Size | Field Name | Field Description | Field Contents |
|---|---|---|---|---|
| *Root Layer (From [sACN], 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<br>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 |
| *Framing Layer* | | | | |
| 38-39 | 2 | Flags and Length | Protocol flags and length | Low 12 bits = PDU length<br>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 (see 3.1.2.1 Source Name) | UTF-8 [UTF-8] encoded string, null-terminated **(shall be ignored by Receivers)** |
| 108 | 1 | *Priority* | Data priority if multiple sources (see 3.1.2.2 Priority) | 0-200, default 100. **(shall be ignored by Receivers)** |
| 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 (see 3.1.2.3 Options) | Bit 7 = Preview_Data<br>Bit 6 = Stream_Terminated **= 0** (see [sACN] Section 3.1)<br>Bit 5 = Force_Synchronization |
| 113-114 | 2 | Universe | Universe Number | Identifier for a distinct stream of DMX512-A [DMX] Data |
| *DMP Layer* | | | | |
| 115-116 | 2 | Flags and Length | Protocol flags and length | Low 12 bits = PDU length<br>High 4 bits = 0x7 |
| 117 | 1 | Vector | Identifies DMP Set Property Message | VECTOR_DMP_SET_P |

| Octet | Field Size | Field Name | Field Description | Field Contents |
|---|---|---|---|---|
| | | | PDU | ROPERTY<br>(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** | **1** | *First property value* | **DMX512-A START Code** (see 3.1.3.1 First Property Value) | **ASC_PER_SLOT_PRIORITY** |
| **126-637** | **0-512** | *Data Slot Property values* | **Per-Slot Priority Data** (see 3.1.3.2 Data Slot Property Values) | **Per-Slot Priority Data** |

#### 4.1.1 Root Layer
The Root Layer of a Per-Slot Priority Packet is identical to that of a Data Packet. For more information on the Root Layer, see [sACN], Section 5.

#### 4.1.2 Framing Layer
The Framing Layer of a Per-Slot Priority Packet diverges slightly from that of other sACN Packets ([sACN], Section 6. The following fields have additional constraints:

##### 4.1.2.1 Source Name
The Source Name field may be overriden by Sources to include supplemental information that distinguishes Per-Slot Priority Packets from NULL START Code Packets, especially when viewed in packet analyzers. Receivers shall ignore this field.

##### 4.1.2.2 Priority
Receivers supporting Per-Slot Priority do not have need for the Packet Priority Field and may ignore it. Sources shall set this field to the Priority used in the corresponding NULL START Code Packets.

##### 4.1.2.3 Options

**Stream_Terminated**
The Stream_Terminated bit shall always be set to 0, indicating an active stream that is not set to terminate ([sACN], Section 6.2.6). Any termination signals must be sent in other sACN Packets.

**Preview_Data**
The Preview_Data bit shall be set to the same value held in the NULL START Code Data Packets to which the Per-Slot Priority Packet corresponds.

#### 4.1.3 DMP Layer
The DMP Layer of a Per-Slot Priority Packet is the same as a Data Packet, but the DMX512-A START Code shall only be ASC_PER_SLOT_PRIORITY.

##### 4.1.3.1 First Property Value
The leading Property value field (see [sACN], Section 7.7) holds a single octet representing the DMX512-A START Code. This First Property value shall be set to ASC_PER_SLOT_PRIORITY for all Per-Slot Priority Packets.

##### 4.3.1.2 Data Slot Property Values
Any Property values after the First Property value shall be Data Slot Property values. The remaining slots shall contain the Per-Slot Priority Value for the corresponding DMX slot in an sACN Data Packet. Since both [DMX] and [sACN] allow for packets that contain fewer than the full 512 octets, the Per-Slot Priority Property value may, likewise, occupy anywhere from 0 to the full 512 octets.

A Receiver receiving a Per-Slot Priority Packet which contains fewer than 512 DMX slot values, but is expecting more slot values than it has received, shall act on the remaining slot values from that Source as though they were PER_SLOT_PRIORITY_NONE.

### 4.2 Priority Values
A Per-Slot Priority Value indicates the Source’s desired priority for the corresponding DMX slot in an sACN Data Packet, and shall only be applied for the Universe number also transmitted in the same Per-Slot Priority Packet.

Per-Slot Priority Values shall be expressed by a single octet in the range of PER_SLOT_PRIORITY_MIN to PER_SLOT_PRIORITY_MAX. Priority values numerically above PER_SLOT_PRIORITY_MAX shall not be transmitted on the network. Receivers shall ignore any Per-Slot Priority value greater than PER_SLOT_PRIORITY_MAX.

A value of PER_SLOT_PRIORITY_NONE indicates that the corresponding Property Value supplied by this Source shall be ignored by the Receiver.

### 4.3 Requirements for Sources

A Source may begin or cease transmission of Per-Slot Priority Packets at any time it wishes to assert or de-assert its control over a specific slot or set of slots.

*Note: Per-Slot Priority Packets are only one type of Data Packet, and they will exist alongside Data Packets that carry other START Codes. Section 6.2.5 of [sACN] requires that the Sequence Number be incremented for every Data Packet on a given Universe, regardless of whether it is a NULL START Code Packet, a Per-Slot Priority Packet, or any other type of Data Packet.*

Per-Slot Priority Packets shall be transmitted at the same rate as the NULL START Code Packets they apply to.

To reduce repetitive transmissions that may occupy unnecessary network bandwidth, Sources sending Per-Slot Priority Packets shall adhere to the rules for transmission suppression laid out in [sACN] Section 6.6.2, overriding the restriction that those rules shall apply only to NULL START Code data.

*Note: Implementors should be aware that Receivers which do not support Per-Slot Priority may have unpredictable behavior when receiving Data Packets from Sources which do support Per-Slot Priority. To encourage consistent results from such Receivers, implementors of Sources may wish to modify their Data Packets. For example, some extant implementations which send to Receivers that use the HTP algorithm send slots having their priority set to PER_SLOT_PRIORITY_NONE at DMX_LEVEL_ZERO. This ensures that those slots are not considered when calculating the final level. An alternative approach in this same situation could be for the Source to alter the overall Priority of each Data Packet.*

#### 4.3.1 Taking Control

When starting transmission using Per-Slot Priority, a Source shall send interleaved NULL START Code Packets and Per-Slot Priority Packets at the same rate until transmission suppression begins. At least three Per-Slot Priority Packets must be transmitted in less than PER_SLOT_PRIORITY_STARTUP_TIMEOUT.

*Note: Multiple packets are transmitted to ensure that loss of a single packet does not significantly affect the receiver merge result. See [sACN] Sections 6.2.6 and 6.6.2.*

When an already-active Source wishes to take control of one or more slots, it should transmit at least three NULL START Code Packets containing the desired DMX level(s) for the slot(s) before transmitting the first Per-Slot Priority Packet taking control of those slots. This is to ensure that Receivers have valid DMX levels for these newly controlled slots before they begin to act upon them.

*Note: A Source that ceases transmission of Per-Slot Priority Packets, while continuing to transmit NULL START Code Packets will implicitly take control of all slots so transmitted after the E131_NETWORK_DATA_LOSS_TIMEOUT expires in each applicable Receiver. This may appear unpredictable to users, so it is recommended that Sources take control explicitly using Per-Slot Priority packets.*

#### 4.3.2 Releasing Control

When a Source wishes to release control of one or more Slots while retaining control of other slots, it should transmit at least three Per-Slot Priority Packets setting the undesired slot(s) to PER_SLOT_PRIORITY_NONE before sending the first NULL START Code Packet setting these DMX levels to DMX_LEVEL_ZERO. This is to ensure Receivers have registered the release of control and can undertake proper Source Loss Behavior.

*Note: Multiple packets are transmitted to ensure that loss of a single packet does not significantly affect the receiver merge result. See [sACN] Sections 6.2.6 and 6.6.2.*

When a Source wishes to release control of all slots in a Universe, it shall cease transmission of NULL START Code Packets as per the requirements of [sACN] Section 6.2.6.

### 4.4 Requirements for Receivers

When in an operating mode that enables Per-Slot Priority, Receivers shall use the following merge algorithm in place of that defined in [sACN] Section 6.2.3.

For a given Universe number, Sources may be in one of the following states:

Table 4-2: Receiver states

| State | Description |
|---|---|
| Active-Per-Universe | The Source is being merged using the [sACN] Packet Priority Field |
| Active-Per-Slot | The Source is being merged using Per-Slot Priority |
| Pending-Priority | Receiver is waiting for a valid Per-Slot Priority Packet from a new Source |
| Pending-Levels | Receiver has Per-Slot Priority values and is waiting for a valid NULL START Code Packet from the Source |
| Inactive | Source has ceased transmission of both NULL START Code and Per-Slot Priority Packets |

Only Sources that are in the Active-Per-Slot and Active-Per-Universe states shall be considered in the merge.

The most recent NULL START Code Packet from a single Source supersedes any previous NULL START Code Packets from that Source.

The most recent Per-Slot Priority Packet from a single Source supersedes any previous Per-Slot Priority Packets from that Source.

### 4.4.1 Receiver Merge Algorithm

Sources shall be merged in two phases:

1. All Sources in the Active-Per-Universe state
2. All Sources in the Active-Per-Slot state

Once all Sources are merged, any slots with the resultant Priority of PER_SLOT_PRIORITY_NONE shall perform the appropriate Source Loss Behavior with the final DMX levels actioned.

Receivers shall execute their Source Loss Behavior on a per-slot basis. A source lost condition may exist for some DMX levels in a Universe and not for other DMX levels in that same Universe.

#### 4.4.1.1 Active-Per-Universe

The Source that has provided a DMX level for the slot, and has the numerically greatest Packet Priority Field, shall be the controlling Source for that slot. If there are multiple candidate Sources for a slot, each with the same numerically greatest Packet Priority Field value, then the HTP algorithm shall be used to arbitrate—meaning that the numerically largest DMX level shall be used for that slot.

The Slot Priority of each slot shall be set to the value of the Packet Priority that set the DMX level of that slot. If the resultant Slot Priority is equal to PER_SLOT_PRIORITY_NONE, then it shall instead be regarded as if it were PER_SLOT_PRIORITY_MIN.

Slots that have no value from any Active-Per-Universe Source shall be given the DMX level DMX_LEVEL_ZERO and Per-Slot Priority PER_SLOT_PRIORITY_NONE.

This result shall be fed into the Active-Per-Slot phase as an additional Active-Per-Slot Source.

#### 4.4.1.2 Active-Per-Slot

The final controlling Source of each slot shall be the numerically greatest of the Slot Priority result of the Active-Per-Universe result and the Per-Slot Priority values of Active-Per-Slot. If a given slot has multiple Sources with the same greatest Slot Priority, then the HTP algorithm shall be used, meaning that the one with the largest DMX level shall be used.

### 4.4.2 Receiver State Transitions

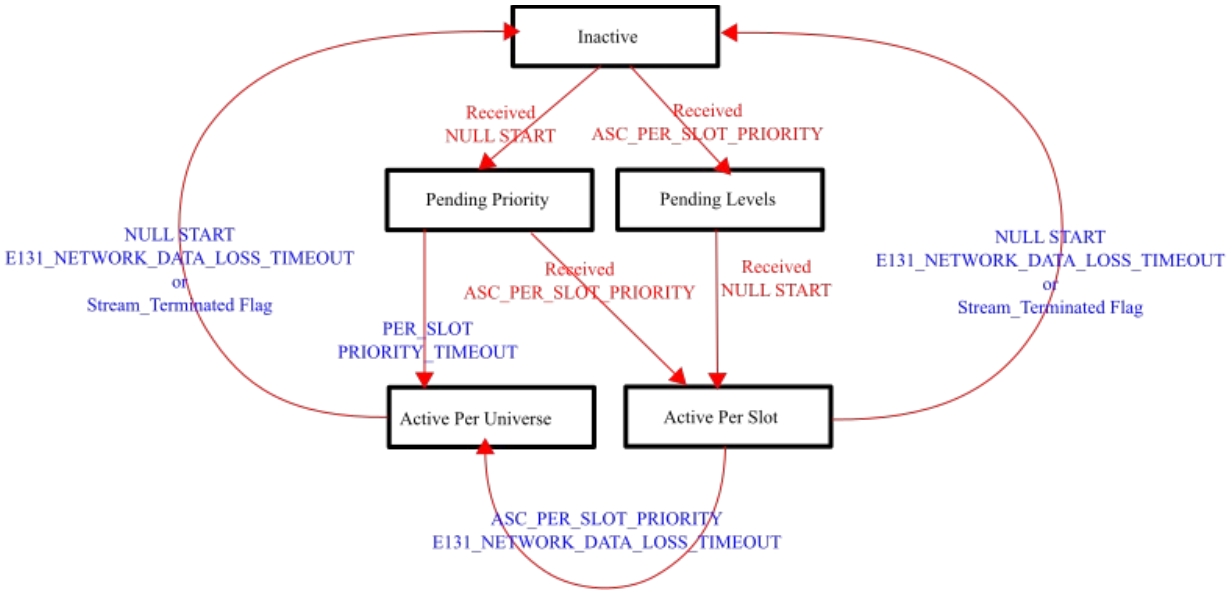


**Figure 4-1: Receiver State Transitions**

All Sources begin in the Inactive state. Receivers do not need to store any information about Inactive Sources.

When a Receiver receives a valid [sACN] Data Packet from a Source, it shall be processed as follows:

- ASC_PER_SLOT_PRIORITY:
  The updated slot priority values for this Source are stored.
  The Per-Slot Priority E131_NETWORK_DATA_LOSS_TIMEOUT timer for this Source is restarted.
  - If the Source is currently in the Inactive state, it enters the Pending-Levels state
  - If the Source is currently in the Pending-Priority state, it enters the Active-Per-Slot state and the PER_SLOT_PRIORITY_STARTUP_TIMEOUT timer is canceled.

- NULL START Code:
  The updated DMX levels and Packet Priority Field for this Source are stored.
  The DMX level E131_NETWORK_DATA_LOSS_TIMEOUT timer for this Source is restarted.
  - If the Source is currently in the Inactive state, it enters the Pending-Priority state.
    The Source's PER_SLOT_PRIORITY_STARTUP_TIMEOUT timer is started.
  - If the Source is currently in the Pending-Levels state, it enters the Active-Per-Slot state.

- NULL START Code, Stream_Terminated Flag Set:
  The Source enters the Inactive state.

Source timer expiration shall transition Sources as follows:

- Per-Slot Priority E131_NETWORK_DATA_LOSS_TIMEOUT timer expires:
  If the Source is currently in the Active-Per-Slot state, it enters the Active-Per-Universe state.

- DMX level E131_NETWORK_DATA_LOSS_TIMEOUT timer expires:
  If the Source is currently in the Active-Per-Slot or Active-Per-Universe state, it enters the Inactive state.

- PER_SLOT_PRIORITY_STARTUP_TIMEOUT timer expires:

If the Source is currently in the Pending-Priority state, it enters the Active-Per-Universe state.

## Appendices

### Appendix A: Defined Parameters (Normative)

Table A-1: Normative

| | |
|---|---|
| ASC_PER_SLOT_PRIORITY | 0xDD |
| DMX_LEVEL_ZERO | 0x00 |
| PER_SLOT_PRIORITY_NONE | 0x00 |
| PER_SLOT_PRIORITY_MIN | 0x01 (1) |
| PER_SLOT_PRIORITY_MAX | 0xC8 (200) |
| PER_SLOT_PRIORITY_STARTUP_TIMEOUT | 1.5 seconds |

Table A-2: Informative

| | |
|---|---|
| E131_NETWORK_DATA_LOSS_TIMEOUT | 2.5 seconds |
| *Note: This value is maintained in [sACN], Appendix A and is provided here as a convenience only* | |

### Appendix B: Examples (Informative)

It is assumed that, unless otherwise stated, all packets in the example tables arrive before the associated timeouts for their type.

For these examples, "NSC" means "NULL START Code".

**Example 1**

This example details the messages from a single Source to a single Receiver. The Source intends to start up, take control of Slot 1 and Slot 2 on the Receiver at a Per-Slot Priority of 100, and then release control of only Slot 1.

The following table shows the messages sent from the Source to the Receiver in order to complete these actions. The messages are shown from the perspective of the Receiver, in the order they have arrived.

Source Startup and Transition to Single Slot

| | | | Network Values | | Receiver Output Values | | |
|---|---|---|---|---|---|---|---|
| **Arrival Order** | **Source** | START Code | Slot 1 | Slot 2 | Slot 1 | Slot 2 | Expected Result |
| **1** | **A** | ASC_PER_SLOT_PRIORITY | 100 | 100 | - | - | Data Loss Behavior |
| **2** | **A** | NSC | 0 | 0 | 0 | 0 | Active-Per-Slot |
| **3** | **A** | ASC_PER_SLOT_PRIORITY | PER_SLOT_PRIORITY_NONE | 100 | - | 0 | Released Slot 1 to Data Loss Behavior |
| **4** | **A** | NSC | 10 | 10 | - | 10 | New Slot 1 level is ignored |
| **5** | **A** | ASC_PER_SLOT_PRIORITY | PER_SLOT_PRIORITY_NONE | 100 | - | 0 | |
| **6** | **A** | NSC | 10 | 10 | - | 10 | New Slot 1 level is ignored |
| **7** | **A** | ASC_PER_SLOT_PRIORITY | PER_SLOT_PRIORITY_NONE | 100 | - | 0 | |
| **8** | **A** | NSC | 10 | 10 | - | 10 | New Slot 1 level is ignored |

**Example 2**

This example details the messages from two Sources A and B to a single Receiver, both using Per Slot Priority.
Source A intends to start up, take control of Slot 1 and Slot 2 on the Receiver at a Per-Slot Priority of 100, and then release control of only Slot 2.
Source B intends to start up without taking control of any slots, then take control of Slot 1 and Slot 2 at a Per-Slot Priority of 100.

The following table shows the messages sent from the Sources to the Receiver in order to complete these actions. The messages are shown from the perspective of the Receiver, in the order they have arrived.

Two Per-Slot Sources

| | | Network Values | | Receiver Output Values | | |
|---|---|---|---|---|---|---|
| **Source** | START Code | Slot 1 | Slot 2 | Slot 1 | Slot 2 | Expected Result |
| **A** | ASC_PER_SLOT_PRIORITY | 100 | 100 | - | - | A is Pending-Levels |
| **B** | NSC | 10 | 10 | - | - | B is Pending-Priority |
| **B** | ASC_PER_SLOT_PRIORITY | 0 | 0 | - | - | B is Active-Per-Slot, but both slots are ignored |
| **A** | NSC | 255 | 255 | 255 | 255 | A is Active-Per-Slot |
| **A** | ASC_PER_SLOT_PRIORITY | 100 | 0 | 255 | - | A has released Slot 2 |
| **B** | ASC_PER_SLOT_PRIORITY | 100 | 100 | 255 | 10 | Slot 1 now merged HTP, and B owns Slot 2 |

**Example 3**
This example details the messages from two Sources A and B to a single Receiver where only one source is using Per Slot Priority.

Source A intends to start up and take control of Slot 1 and Slot 2 on the Receiver at different Per-Slot Priorities. Source B is a pure E1.31 source and does not use Per Slot Priority. It intends to transmit the entire universe at priority 100.

The following table shows the messages sent from the Sources to the Receiver in order to complete these actions. The messages are shown from the perspective of the Receiver, in the order they have arrived.

Merge Per-Slot with Per-Universe

| | | | Network Values | | Receiver Output Values | | |
|---|---|---|---|---|---|---|---|
| **Source** | [sACN] Priority Field | START Code | Slot 1 | Slot 2 | Slot 1 | Slot 2 | Expected Result |
| **A** | *Any* | ASC_PER_SLOT_PRIORITY | 10 | 150 | Data Loss | Data Loss | A is Pending-Levels |
| **A** | *Any* | NSC | 255 | 255 | 255 | 255 | A is Active-Per-Slot |
| **B** | 100 | NSC | 20 | 20 | 255 | 255 | B is Pending-Priority |
| **A** | *Any* | ASC_PER_SLOT_PRIORITY | 10 | 150 | 255 | 255 | |
| **A** | *Any* | NSC | 255 | 255 | 255 | 255 | |
| **B** | 100 | NSC | 20 | 20 | 255 | 255 | |
| ***Above packets repeat…*** | | | | | | | |
| **B** | PER_SLOT_PRIORITY_STARTUP_TIMEOUT expires | | | | 20 | 255 | B is Active-Per-Universe |
| **A** | *Any* | NSC | 255 | 255 | 20 | 255 | |
| **B** | 100 | NSC | 20 | 20 | 20 | 255 | |