



Control Protocols Working Group

ANSI E1.31 - 2018, Lightweight streaming protocol for transport of DMX512 using ACN Reaffirmation Public Review Draft Comment Resolutions

Reference document: ANSI E1.31 - 2018, *Lightweight streaming protocol for transport of DMX512 using ACN* (Document number CP/2014-1009r6a)

ANSI Public review period: 11 August 2023 through 25 September 2023

Question: In your opinion, do you think the requirements of ANSI E1.31 - 2018, *Lightweight streaming protocol for transport of DMX512 using ACN* (DCN CP/2014-1009r6a) are reasonable, and adequately address the intended subject matter?

Please answer the question using one of the options below. Select “Yes”, “Yes, but...” (provide comments to support your opinion), or “No, with reasons” (the document’s requirements are unacceptable or unreasonable).

Responses:

Peter Newman, Open Lighting Project (PN)	Yes, but...
--	-------------

Individual Comments:

No.	Commenter	Ref. section	Comment	Resolution
1	PN	General	I think the standard is generally pretty clear and functional, as proved by the level of take up over the years. I've found a couple of simple typos which I'd hope could be fixed immediately either directly in the document or with some sort of errata addition. There are also some other relatively minor issues/opportunities to improve the clarity of the document which I suspect will need to wait until it's next opened for a full review which I guess may not be immediately.	N/A
2	PN	Section 6.7.1: Network Data Loss	Typo: "receipt of a packet containing the Options field, bit 6 set to value of 1 (see Section 6.6 Options).", section 6.6 doesn't have an options section. I suspect you mean 6.2.6. Also no hyperlink there which I guess would have auto-updated if it was present?	Accept Corrected reference
3	PN	Section 6.2.6: E1.31 Data Packet: Options	Clarification: Probably a lost cause given the standard has been out for ages, but maybe some clarification could be added to flag up to people. "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	Reject You are correct that this is a poor name, however the standard does clearly define the meaning.

No.	Commenter	Ref. section	Comment	Resolution
			synchronized state shall not update with any new packets until synchronization resumes.". This is directly opposite to "Stream_Terminated: Bit 6" and "Preview_Data: Bit 7 (most significant bit)" where a one means the thing is true. Really this is Don't_Force_Synchronization.	
4	PN	Section 6.7.1.1: Network Data Loss and Universe Discovery	<p>Clarification: ""In order to mitigate jitter on a lossy network, sources experiencing a network data loss condition on an E1.31 universe need not immediately reflect that change to their E1.31 Universe Discovery List of Universes"" but 6.7.1 says ""Network data loss is a condition that is defined as either the absence of reception of E1.31 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"" and 4.3 says ""E1.31 Universe Discovery Packets shall be sent by a source to advertise which E1.31 universes it is actively transmitting on.""</p> <p>Given we're sending via UDP and there doesn't seem to be any acknowledgement layer, how is the sender/source supposed to know that a receiver is failing to get it's packets, short of being unplugged from the network which it could detect, but updating it's universe list at this point would be pretty futile!</p>	Accept Corrected typo Sources -> Receivers
5	PN	Section 8: Universe Discovery Layer	Clarification: No mention is given either way of whether it's acceptable (e.g. due to memory constraints) to send e.g. 10 pages of 50 universes instead of one page of 500 universes. I assume given the standard doesn't mention it, it's allowed, but it would be good to flag up this edge case to discovery data consumers, or explicitly rule it out if not intentional.	Reject 8.3 specifies that the Source may divide the list into pages if they do not fit. Source maximum packet size is unspecified, and so receivers are required to handle arbitrary packet spanning.
6	PN	Section B.2: Universe Synchronization for Receivers	Clarification: "it will discard all but the most recent E1.31 Data Packet" should add the "(for a more thorough examination of sequence numbers, see 6.7.2)" clause used elsewhere.	Accept In Principle Added reference
7	PN	Section 3.7: Active Data Slots	Clarification: "most recently received packet" should add the "(for a more thorough examination of sequence numbers, see 6.7.2)" clause used elsewhere.	Reject This paragraph refers to translation from E1.11, which does not have sequence numbers.
8	PN	Section 8.4: Last Page	Typo: "indicating the number of the final page being to be transmitted" needs to drop either "being" or "to be", having both doesn't make sense.	Accept Corrected typo "to be"
9	PN	Section 6.7.1: Network Data Loss	Clarification: "the absence of reception of E1.31 packets from a given source for a period of E131_NETWORK_DATA_LOSS_TIMEOUT", should this not explicitly say data packets, given some legacy stuff may not send discovery (as mentioned in 12)?	Accept Corrected typo

No.	Commenter	Ref. section	Comment	Resolution
10	PN	Section 12: Universe Discovery	<p>Clarification: There are no sequence numbers in Universe Discovery packets, but equally no mention of what to do if you get partial or out of order pages. Does the reception of a lower numbered page reset the data? E.g. if you get pages 0, 1, 3, 2, 0 do you just ignore the first run of out of sequence once? What about if you never get the last page (although I applaud dealing with last page rather than number of pages), or you get 0, 1, then 2, 3 with a gap bigger than the discovery interval in between?</p> <p>I note Simon Newton had put a sequence number into the much earlier draft version of E1.31 he'd been implementing with a note to try and get it in the standard, I think it would have resolved most of these issues as you could follow the same rules as the other packets regarding what to discard.</p>	<p>Reject As this is UDP transport, pages may be received out of order or dropped. Receivers may handle this as they see fit. The standard provides some guidance in section 6.7.1.1.</p>