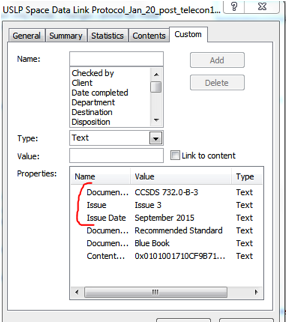
FOREWORD – Do not assume that not commented sections will never get comments in future. 

1. **Correct File Properties Summary to show correct title**.  The current value is set to “AOS Space Data Link Protocol”. This is now fixed in new version.

2. **Correct File Properties Custom to show correct values**.  The current value for Number, Issue, and Date are wrong. This should also fix the wrong footer. This is now fixed in new version.



**3. Correct page Header.** It is currently pointing to “TM SPACE DATA LINK PROTOCOL”  This is now correct and will be changed in new version.

**4. Correct central column for Figure 2-1 Relationship with OSI Layers.**Show the two CCSDS sub layers. This is now correct and will be changed in new version.

**5. In section 2 there is no service mentioned for Higher Protocols Data Units.** This version of USLP proposes to be able to embed directly Higher Protocols Data Units acting basically as encapsulation service (or skipping encapsulation service, if you prefer).

Actually the MAPP Service includes this feature, but it is questionable that the SDU can be considered a Packet.

In general I think that this feature deserves some dedicated discussion with WG Members at the Spring 2016 Meetings. Note that this comment has no position with respect to the validity or not of such new service.

Therefore I suggest:

1.     Adding a row with the new service to “Table 2 1:  Summary of Services Provided by USLP Space Data Link Protocol” marking it e.g. “for future discussion” Agreed.

2.     SLP Chair to present a dedicated presentation for Spring 2016 Meeting to address characteristics and implication of this new feature. Agreed

**6. Clause 3.2.2.2 is mixing data units.** The clause is fully valid for Packets but not for Higher Protocols Data Units. Moreover such clause is putting a requirement on the external data unit and I am not sure this is fully correct. I mean that it is true that “The Packets transferred by this service must be self-delimiting” but I am not sure this formulation is fully correct. In fact the corresponding requirement in 132.0-B simply states: *3.2.2.2 The Packets transferred by this service must have a Packet Version Number (PVN) authorized by CCSDS.  Further, each Packet transferred must conform to the corresponding packet format specified by reference [6]*. In other words it is the upper layer that checks that packet defined there are self-delimiting (i.e. they have a length field).

SANA will only provide a PID if the protocol is registered and the data unit contains a length field in the location specified in the SANA registered protocol.

Note that currently <http://sanaregistry.org/r/packet_version_number/> defines only the Space Packet and the Encapsulation Packet.

Note that defining separately a service for Higher Protocol Data Units would split/cure the problem.

**7. The new service on Higher Protocols Data Units imposes requirements on protocols at higher layers.** As mentioned in the previous comment, it is up to the upper layer that checks that protocol data units defined there are self-delimiting.

Note that currently <http://sanaregistry.org/r/protocol_id/protocol_id.html> defines the  identifies as dedicated to the Encapsulation service. The general philosophy behind that was that those protocol may not be producing self-delimiting units and then the Encapsulation Service fix it as the Encapsulation Packet Header is mandatory and it shall have a length field.   If the protocol does not  provide a length field then it must use the encapsulation or VCA service.

I have the impression that all the Protocol Id’s currently defined contain a length field.



Therefore I doubt that 000 and 111 can be used in USLP.

The following questions are:

a)    Does LTP <http://public.ccsds.org/publications/archive/734x1b1.pdf> defines a length field that can be used by USLP for eventual extraction of LTP Data Units?

b)   Does IP <http://public.ccsds.org/publications/archive/734x1b1.pdf> defines a length field that can be used by USLP for eventual extraction of LTP Data Units?

c)    Does BP <http://public.ccsds.org/publications/archive/734x2b1.pdf> defines a length field that can be used by USLP for eventual extraction of LTP Data Units?

Conversely, I think CFDP includes a PDU Data field length that can be used by USLP for eventual extraction of CFDP Data Units. (And then a question is why not including CFDP among allowed values for PVN?)     PVN for CFDP is unique to CFDP.  PVN for space packets and Encap are co-ordinated.

Having said this, it is clear that an unlimited approach for allowing USLP to carry Higher Protocols Data Units is not possible unless a **new** requirement imposes to  the registry <http://sanaregistry.org/r/protocol_id/protocol_id.html> to only include protocols with self-delimiting data units. As this is not possible, USLP should limit the use of this service to those protocols with self-delimiting data units.   The PIDs will be registered in SANA for use in USLP.  There will be 5 bit PID and 13 bit PIDs (for extended PIDs)

In general see my suggestion under comment #5.

**8. Clause 3.2.2.4 is redundant with respect to clause 3.2.2.3.** This clause states “*3.2.2.4 If blocking of Packets is performed by the service provider, the position and length of the Packet Length Field of the Packets must be known to the service provider in order to extract Packets from Transfer Frames at the receiving end*” and it is a special case of clause “3.2.2.3 The position and length of the Packet Length Field of the Packets must be known to the service provider in order to extract Packets from Transfer Frames at the receiving end”. The redundant clause should be removed and a comment added to the previous clause to state that this clause applies also when blocking of Packets is performed by the service provider.  OK I have fixed it.

**9.  Is the minimum size of the USLP Transfer Frame limited to 4 octets?** The clause “*4.1.1.1 A USLP Transfer Frame shall encompass the major fields, positioned contiguously, in the following sequence: a)            Transfer Frame Primary Header (7 to 14 octets; mandatory)…..*” is in conflict with other clause(s) stating that the limit is 4 octets (see e.g. 4.1.2.6.2).   Change # in a)  and remove Further from the sentence in NOTE.

NOTE  – The Transfer Frame Primary Header can be ~~further~~ limited in size to 4 octets if the End of Transfer Frame Primary Header Flag is set to “1”. See 4.1.2.6 and the USLP Green Book reference [B11] for the motivation of this exceptional case.

Figure 4-1 contains also the value 7 as minimum. This is now fixed.

**10. Figure 4-2 should mark that the VC Frame Count field is optional.** Same approach of Figure 4-1 should be used.  No because this diagram and text both state that the field is 0-56 bits long.

**11. Note to clause 4.1.2.2.2.2 is redundant.** The NOTE could be deleted and the word binary added at the end of the clause to state “*4.1.2.2.2.2                        This 4-bit field shall identify the data unit as a Transfer Frame defined by this Recommended Standard; it shall be set to ‘1100’ binary.*”.   OK, this has been fixed.

**12. The document contains unnecessary change bars (left over from originating doc).** Delete the hard coded bars. OK, this has been fixed.

**13. The Frame Header does not contain the Control Flag needed e.g. by COP-1.** The USP Frame Header contains only a Bypass Flag defined in section 4.1.2.9. Suggest adding a NOTE to that section explain the USLP approach with respect to the Control Flag needed by COP-1. BTW, it looks that the “PROTOCOL IDENTIFIER WITHIN USLP DATA ZONE” is used for this.  A note has been added to 4.1.2.9 explaining this.

**14. There should be a section for each TFDZ Construction Rule.** (Short sub) section/clauses should describe each Rule. The figure is useful but cannot be considered a normative rule.  The statement just before the table is a “shall” statement. It says “The TFDZ Construction Rules shall be interpreted as shown in table 4-3”. CCSDS protocols have state tables that are normative. One cannot easily make everything into a shall statement in that case. Doesn’t it make more sense to summarize this requirement in a table ? Perhaps this is a Tom Gannett question ? I have also provided the requirement statements as a comparison.

**15. Clarify the real use of the “PROTOCOL IDENTIFIER WITHIN USLP DATA ZONE”.** This field is not used only to contain values given by <http://sanaregistry.org/r/protocol_id/protocol_id.html>  and this shall be clarified in section 4.1.4.2.1.4 as well as elsewhere. See also my previous comments on Protocol Id. OK

**16. NOTES in 4.1.4.2.1.4 hide requirements.** Is misleading (in fact some values are not protocol Id’s defined at <http://sanaregistry.org/r/protocol_id/protocol_id.html> ) and the first 3 notes are actually normative clauses.   Agreed.

Moreover this field contains 5 bits and <http://sanaregistry.org/r/protocol_id/protocol_id.html> defines values with 3 bits. Moreover <http://sanaregistry.org/r/extended_protocol_id/extended_protocol_id.html> defines other values with 4 bits. I confirm my suggestion for SLP Chair to carry out a dedicated presentation for Spring 2016 Meeting to address characteristics and implication of the new feature with Protocol Id usage. Agreed. For now we are gathering these IDs but in the end state the USLP book will not contain them, rather SANA will.

**17. Double clause 4.1.4.2.1.6.3.** Clause “*4.1.4.2.1.6.3 The First Header Pointer (used as a Packet pointer) or the Last Valid Octet (used as a MAP\_SDU data pointer) shall delimit the contents of the TFDZ when the TFDF Construction Rules (‘000’ or ‘001’) identify that the TFDZ is transporting either PDU or MAP SDUs. See Section 4.1.4.3.1.4.4, Table 4-3.*” Contains two requirements and should be split. One clause to state *When rule is 000 then…* and the other to state  *When rule is 001 then…* .this section has been rewritten to clarify. We have combined and called the field a “pointer field” which has two subparts: FHP or Last Valid Octet Pointer.

**18. Idle data vs. Last Valid Octet.** To be explained why in this case Idle Packets cannot be used.  This Last Valid Octet Pointer is used for MAP\_SDU and not packets. (Packets require the use of the FHP.)