



Version field options

Ashley Butterworth
Apple Inc

The Problem

- As we noted during the Clause 4 V1 header modifications there are some questions around what does the version field mean and do we need to put diagrams everywhere
- Which area does it define?
 - Is it the version of the whole AVTPDU and encapsulated protocol?
 - Is it the version of the AVTPDU's header, that is everything between the version field and the stream_data_payload/control_data_payload/alternative_data_payload field?
- Is the definition shared by all AVTPDU formats (stream, control and alternative) or is it per format.
- How does this apply to stream formats like CRF which use the alternative header

What does version define

- If it defines the AVTPDU's header, that is everything between the version field and the stream_data_payload/control_data_payload/alternative_data_payload field (my currently preferred interpretation) then some protocols may not have a way to expand (e.g. CRF)
- If it defines the whole AVTPDU including the subtypes payload then the version number may mean different things for different subtypes even though they share the same format of AVTPDU (stream, control or alternative).
- Is the definition of the version field entirely in clause 4 or do subtypes get to define the version field values (excluding the subtypes ability to limit what options are available)

Shared definition

- Is the definition of a version number shared across all of the AVTPDU types?
- There are only 8 possible values for version (3-bit number).
- We are currently defining a set of header changes that only apply to a Stream AVTPDU. It seems like a bad idea to burn a version number across everything when there would be no change to Control and Alternative headers in V1
- Separate definitions allow us to add extra to the Stream header without using version numbers on the other 2

CRF and NTSCF

- CRF and NTSCF are an interesting case. Both use the Alternative AVTPDU format since they don't have a presentation timestamp but both include (or optionally include) timestamps that use gPTP.
- They need a place for the gPTP GM ID and expanded sequence number.
- If the version field is defined by clause 4 then we have an option to include an expanded sequence number (maybe 20 bits of extension to the existing sequence numbers???) and GM ID and have it common to any subtype. Additionally in this case CRF and NTSCF have no version number of their own to be able to expand their header
- If the version field is defined by the subtype then CRF and NTSCF can add fields

CRF and NTSCF (cont)

- CRF and NTSCF have an alternative for the GM ID but not the extended sequence number, both have message/event types within the PDU which could provide another version which includes the GM ID in the message/event itself.
- The downside of this is that it leads to message bloat and we now have an exponential growth in message/event types.

AVTPDU version and subtypes

- At present it is expected that all subtypes using the Stream AVTPDU type will support version 1 headers (nothing seems to need to exclude it)
- Currently each subtype includes a diagram with a full V0 AVTPDU in it. Do we need to include both the V0 and V1 diagrams?
- AVTPDU Stream Diagrams occur in subclauses
 - 5.2.1, 5.3.1, 5.4.3.1, 5.4.4, 5.4.5.1, 7.2.1.1, 7.3.1.1, 7.4.1.1, 8.3.1, 8.4.1, 8.5.1, 8.6.1, 9.3.1, 11.2.1, 12.2.1, 14.1.1.1

-