CPRI structure aware mapper - clarifications

Jouni Korhonen, Liquan Yuan

8/11/2015

Background

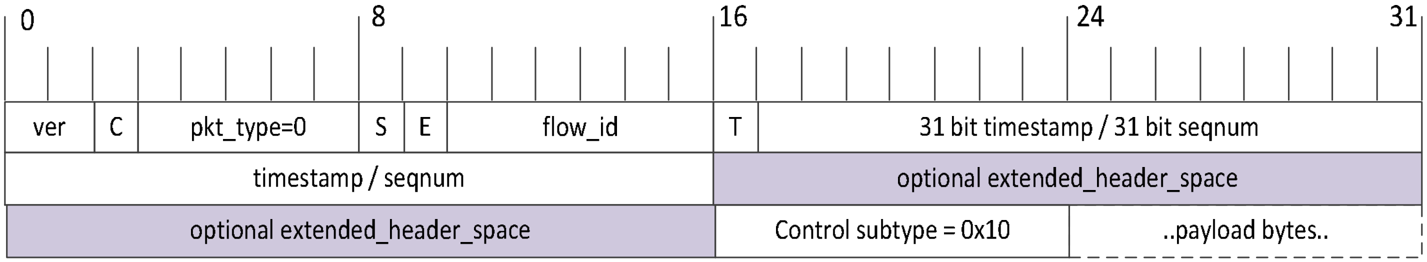
- ❑ Continuing on the agreed baseline tf3_1506_korhonen_9a.pdf this document proposes:
  - How to packetize C&M flows.
  - How to packetize VSD flows.
  - How to packetize Ctrl_AxC flows.
  - How to handle ”synchronization” control words.
  - Start of frame bit usage.
  - How AxC Container shall be structured in a BF.

Fast C&M flow

- The Fast C&M flow ”payload only” is extracted and sent as a native Ethernet packet:
  - No RoE header or RoE EthType.
  - In a contrary to CPRI the FCS is added.
  - The Ethernet packet uses the native to the link framing and PMA/PCS/etc.
  - The Fast C&M data flow shall meet the negotiated bit rate (see CPRI 6.1 spec Table 12).
  - The Fast C&M Ethernet frame header addressing (SA/DA) information is copied to native Ethernet packet.

Slow C&M flow

- The Slow C&M is extracted and encapsulated into a RoE Control packet(s):
  - RoE header with pkt_type=0x00.
  - RoE Control Packet with a subtype=0x10.
- The amount of C&M data is implementation defined but is between 64 to 1500 bytes.
  - The byte ordering shown in CPRI 6.1 spec Table 11 shall be followed.
  - The HDLC FLAG and FCS bytes shall not be included but are synthesized at the receiving end.
  - If the HDLC frame ends non-full byte boundary, the last byte is padded with zero bit until full byte boundary is reached (i.e. shift left and add zero bits).
  - Payload length is derived from the Ethernet frame length.
  - The Slow C&M data flow shall meet the negotiated HDLC bit rate (see CPRI 6.1 spec Table 10).
  - In a case of multiple control packets, the receiver is in charge of ”reassembly”.



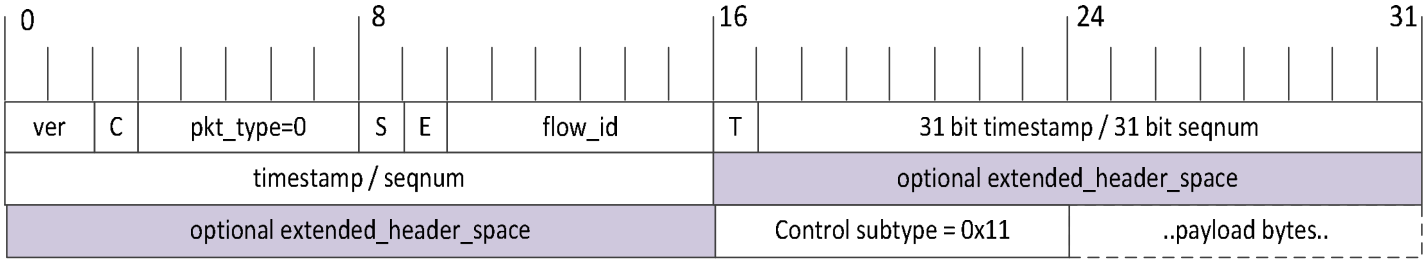
ver | C | pkt_type=0 | S | E | flow_id | T | 31 bit timestamp / 31 bit seqnum

timestamp / seqnum | optional extended_header_space

optional extended_header_space | Control subtype = 0x10 | ..payload bytes..

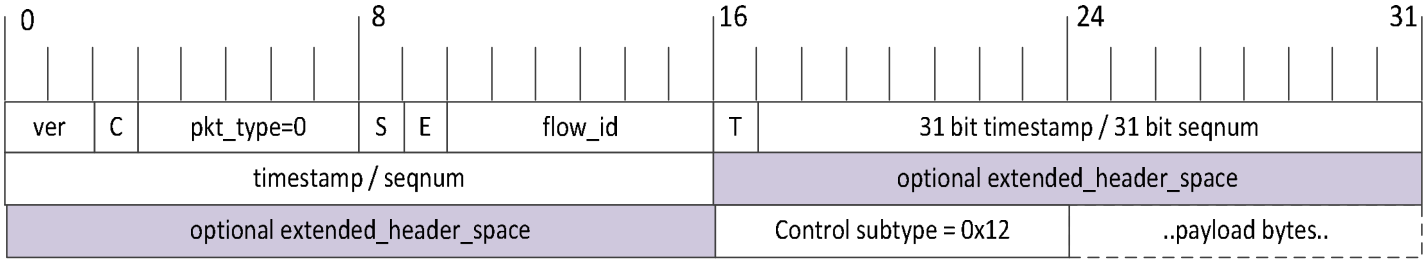
VSD flow

- The VSD is extracted and encapsulated into a RoE Control packet(s):
  - RoE header with pkt_type=0x00.
  - RoE Control Packet with a subtype=0x11.
- The amount of VSD data per RoE packet is implementation defined but is between 64 to 1500 bytes.
  - The byte ordering is big endian, ascending control word order.
  - Payload length is derived from the Ethernet frame length.
  - The RoE encapsulated VSD flow shall meet the bit rate of the CPRI equivalent.
  - In a case of multiple control packets, the receiver is in charge of ”reassembly”.


0
8
16
24
31
ver
C
pkt_type=0
S
E
flow_id
T
31 bit timestamp / 31 bit seqnum
timestamp / seqnum
optional extended_header_space
optional extended_header_space
Control subtype = 0x11
..payload bytes..

Ctrl_AxC flow

- The 16 Ctrl_AxC control words are is extracted and encapsulated into a RoE Control packet(s):
  - RoE header with pkt_type=0x00.
  - RoE Control Packet with a subtype=0x12.
- The amount of Ctrl_AxC data per RoE packet is implementation defined but is between 64 to 1500 bytes.
  - The byte ordering shall follow the CPRI 6.1 spec Section 4.2.7.10 and Figure 23Z first including column Xs=0, then Xs=1, etc.
  - Payload length is derived from the Ethernet frame length.
  - The RoE encapsulated Ctrl_AxC flow shall meet the bit rate of the CPRI equivalent.
  - In a case of multiple control packets, the receiver is in charge of ”reassembly”.



0
8
16
24
31
ver
C
pkt_type=0
S
E
flow_id
T
31 bit timestamp / 31 bit seqnum
timestamp / seqnum
optional extended_header_space
optional extended_header_space
Control subtype = 0x12
..payload bytes..

Start of frame bit usage

- The ”S” bit in the RoE header shall be set to 1 at the start of every new Hyper Frame (HFN). The ”S” bit is set to 0 on other occasions.

AxC Containers within BFs 1/2

- Mapping method 1: IQ sample based shall be supported i.e. IQ Sample based mapping in an AxC Container Block.
  - See CPRI 6.1 spec Section 4.2.7.2.5.
- AxC Container shall implement Option 1 (packed position).
  - See CPRI 6.1 spec Section 4.2.7.2.3.
- Any padding should be avoided.
  - If padding (stuffing bits) is used the sender/receiver shall not assume anything on their ”information content” i.e. set to 0 when sending, ignore on receiving.

AxC Containers within BFs 2/2

- Mapping method 3: Backward compatible shall be supported. See CPRI 6.1 spec Section 4.2.7.2.7.
- AxC Container shall implement Option 1 (packed position).
  - See CPRI 6.1 spec Section 4.2.7.2.3.
- Any AxC Container level padding should be avoided.
  - If padding (stuffing bits) is used the sender/receiver shall not assume anything on their ”information content” i.e. set to 0 when sending, ignore on receiving.
- Stuffing samples shall follow the guidance given in CPRI spec Section 4.2.7.2.7 for (E)-UTRA and GSM.

Motion

- Accept solution listed in tf3_1508_korhonen_cpri_better_mapper_3.pptx pages 3-9 as a baseline for a “better” CPRI mapper.

- John Doe making the motion
- Seconded by Jane Doe

- Technical motion (>=2/3)

- Yes: 0, no: 0, abstain 0

Synchronization control words

- ❑ In this mapper the synchronization word at Z.0.0 is removed/synthesized according to the CPRI line rate/line coding.
- ❑ The HFN byte at Z.64.0 is synthesized with the help of the RoE header S bit i.e. when S=1 -> HFN=0.
- ❑ The BFN as shown in CPRI 6.1 spec Figure 18 is synthesized with the help of the RoE header S bit and internally counted HFN => 150 HFNs -> increment BFN.

Motion

- Accept “synchronization control words” synthezation as described in tf3_1508_korhonen_cpri_better_mapper_3.pptx page 11 as a baseline for a “better” CPRI mapper.

- John Doe making the motion
- Seconded by Jane Doe

- Technical motion (>=2/3)

- Yes: 0, no: 0, abstain 0