IEEE STANDARDS ASSOCIATION

IEEE P1914.3 64B66B considerations

Jouni Korhonen, Broadcom

Richard Tse, Microsemi

Background

- Discussion raised on aligning Structure Agnostic (de)mapper line coding procedures to 64b66b coding as well. Currently 1914.3/D1.2 only describes handling of 8b10b coding.
  - See 1914.3/D1.2 sub-clause 7.1 and 6.8/6.9 (mapper params).

- Issue raised during 1914.3/D1.1 TF review by **multiple** reviewers. The comments were **withdrawn** in order to have a baseline proposal for the handling of 64b66b coding:
  - See tf3_1608_comments_resolved.pdf and comments #3, 13, 14, 53, 55, and 58.

- http://www.ieee1904.org/3/meeting_archive/2015/10/tf3_1510_closing.pdf
  - During Oct'2015 we agreed to add a tunneling mapper.. See http://www.ieee1904.org/3/meeting_archive/2015/10/tf3_1510_bross_agnostic_1.pdf
  - However, there was no motion on this – just gentlemen's agreement.

Reasoning

- There's still unnecessary overhead from the 64b66b line coding:
  - ~3% is not much but still enough to complicate the transport of (popular) CPRI option 10 using structure agnostic RoE.

- Describing the removal/insertion of 64b66b would align the specification, since 8b10b line coding is already there.

- Structure aware mapper mode has to handle 64b66b line coding anyway, so the required functionality is already there in the mapper.

Some numbers..

Assuming CPRI option 10 (24330.24Mb/s, word size 48 octets).

- One BF is 768 octets w/o 64b66b and 792 octets w/ 64b66b.
- Ethernet overhead 38 octets (preamble, DA/SA, no VLANs, FCS and IPG) + RoE overhead 8 octets -> total 46 octets overhead per RoE packet.
- 1 BF per RoE packet -> even w/o 64b66b won't fit into 25G Ethernet link (25.006Gb/s bw needed).
- 1.5 BF per RoE packet w/ 64b66b -> ~25.27Gb/s.
- 1.5 BF per RoE packet **w/o 64b66b -> ~24.54Gb/s**.
- Why 1.5 BF payload size? First "nice" size that is below 1500 octets..

- Summary: CPRI option 10 usable only without 64b66b coding.

RoE with or without 64B66B coding

CPRI 10 rate			
with 66B		24,330.24	Mb/s
without 66B		23,592.96	Mb/s
ENET frame size			
	BF/pkt		
BF with 66B	1	792	bytes
BF w/o 66B	1	768	bytes
BF with 66B	1.5	1188	bytes
BF w/o 66B	1.5	1152	bytes
BF with 66B	2	1584	bytes
BF w/o 66B	2	1536	bytes
ENET properties			
ENET RoE OH		46	bytes
ENET capacity (-100ppm)		24,997.50	Mb/s
Required ENET bandwidth			
	BF/pkt		
BF w 66B	1	25,743.36	Mb/s
BF w/o 66B	1	25,006.08	Mb/s
BF with 66B	1.5	25,272.32	Mb/s
BF w/o 66B	1.5	24,535.04	Mb/s
BF with 66B	2	25,036.80	Mb/s
BF w/o 66B	2	24,299.52	Mb/s

CPRI 100G rate			
with 66B		97,320.96	Mb/s
without 66B		94,371.84	Mb/s
ENET frame size			
	BF/pkt		
BF with 66B	1	3168	bytes
BF w/o 66B	1	3072	bytes
BF with 66B	0.5	1584	bytes
BF w/o 66B	0.5	1536	bytes
BF with 66B	0.375	1188	bytes
BF w/o 66B	0.375	1152	bytes
ENET properties			
ENET RoE OH		46	bytes
ENET capacity (-100ppm)		99,990.00	Mb/s
Required ENET bandwidth			
	BF/pkt		
BF w 66B	1	98,734.08	Mb/s
BF w/o 66B	1	95,784.96	Mb/s
BF with 66B	0.5	100,147.20	Mb/s
BF w/o 66B	0.5	97,198.08	Mb/s
BF with 66B	0.375	101,089.28	Mb/s
BF w/o 66B	0.375	98,140.16	Mb/s

Proposal

Add 64b66b coding remove/add to the Structure Agnostic Mapper.

Changes needed in 1914.3/D1.2:

- Sub-clause 7.1 paragraph b)
  - Add 64b66b text similar to 8b10b removal.
  - Add a mention that FEC is also processed as a part of decoding.
- Sub-clause 7.1.2.3
  - Add required 64b66b text.
- Sub-clause 7.1.2.5
  - Add 64b66b text similar to 8b10b removal.
- Tables 6-4 and 6-5:
  - Rename 8b10 aware mode to e.g., ”agnostic mode”
  - Increase 8b10 aware mode bits from 1 to 2
  - Add .64b66stripMode to 8b10 aware mode (”agnostic mode”)
  - modify packet length mapper type to cover 64b66b and fractional ”BF” sizes. (octet counts for all agnostic mapper cases)
  - Rename .8b10stripMode to e.g., ”.agnosticMode” or similar.

About synchronization 1/4

- P1914.3/D1.2 states for Start of Frame: ”Bit 0 is the Start of frame (SoF) marker and is an indication of a radioframe boundary. When the SoF bit is set to 1, this indicates the start of the payload contained within the packet is the start of the radio frame.”

- The same principle shall be used when sequence numbers are used. The sequence number pattern or event has to be described for the agnostic mapper.

- Proposal: the overflow of p-counter and p-counter being 0 marks the SoF boundary.
  - Document in subclause 7.1.2.3.

About synchronization 2/4

- When timestamps are used with the agnostic mapper the synchronization event and data content alignment for the RoE payload is already defined:
  - On the sync event the start of the payload contained within the packet is the start of the radio frame.
  - In e.g., CPRI case this would always be the Sync Byte (or rather the Sync Control Word).

- When sequence numbers are used with the agnostic mapper the synchronization event and data content alignment for the RoE payload shall be the same as with timestamps.

About synchronization 3/4

- The procedures are the same for both 8b10b and 64b66b cases:
  - The (de)mapper has to know the synchronization event of the framing protocol it deals with.
  - The (de)mapper knows how to (de,en)code properly the line coding as long as the RoE payload is also aligned properly with the synchronization event.
  - The (de)mapper has to provide the RoE (de)packetizer a properly aligned data. The RoE (de)packetizer just relies on the sync event and properly aligned data it exchanges with the (de)mapper.

- The assumption is that the (de)mapper which deals with an arbitrary framing protocol like CPRI or OBSAI cannot be entirely transparent to the stream it handles. It knows a somel of the framing protocol it handles – on the other hand RoE is agnostic.
  - We discussed this in tf3_1606_korhonen_sof_2.pdf
  - IEEE P1914.3 PAR mentions ” The structure-agnostic encapsulation is only a container for the encapsulated <u>digitized radio transport frames</u>.” so framing is known.

About synchronization 4/4 and special character handling

- Continuing the procedures for both 8b10b and 64b66b cases – **a simplified example**:
  - The ”digitized radio transport frame” flow is CBR.
  - The (de)mapper knows the ”digitized radio transport frame” size – an octet counter_1.
  - The (de)mapper knows how many octets there are between potential synchronization events – an octet counter_2, which is a multiple of counter_1.
  - The (de)mapper knows how many octets there are in a radio frame – an octet counter_3, which is a multiple of counter_2.
  - .lenPacket has to be chosen so that some n*.lenPacket equals the octets in a radio frame.
- Special events:
  - When counter_2 wraps -> the octet(s) is on a start of a sync mark -> handle potential 8b10b or 64b66b special characters.
  - When counter_3 wraps -> sync event on a start of frame.

Example numbers.. case CPRI (again)

- Assuming CPRI option 10.
  - Frame size (counter_1) set to 768 (1 BF)
  - Sync events size (counter_2) set to 196608 (1 HF, 256 BFs)
  - Radio frame size (counter_3) set to 29491200 (150 HFs i.e., 38400 BFs)
  - .LenPacket set to 1152 (1.5 BFs)
- When
  - counter_2 % 196608 = 0 -> handle sync mark (remove or substitute based on known masks & values)
  - counter_3 % 29491200 = 0 -> start of frame (either set SoF in timestamp or p-counter shall be zero)
- SoF after 25600 RoE packets.. and sync mark at the beginning of the RoE payload.

Example.. the substitution

- Issue.. how to regerate the line coding?
- Based on the counters the demapper knows where to look for special octet sequences (one or more octets)
- If the predefined comparision sequence matches
  - Do substitution (swap bytes etc)
  - Advise PCS accordingly (K-characters, 64b66b sync headers, ..)

- What if the comparison and counters do not match?
  - You lost sync!
  - Then do whatever needed to resync..

What if..

- The sync mark, for example, CPRI HF Sync Byte is not nicely aligned at the beginning of the RoE payload (does not concern start of frame case – it must be aligned)? This can happen when .lenPacket is not an integer multiple of the frame size.
  - How the line coding add/remove then works?
- The mapper knows what to do because it tracks the radio transport framing protocol sync points anyway.. based on known special charaters.
- The demapper (fifo) counts octets it received from RoE depacketizer.. and the place where counter_2 % ”value” = 0 then that’s the place for special character substitution..

Example 1/2 – sync mark handling

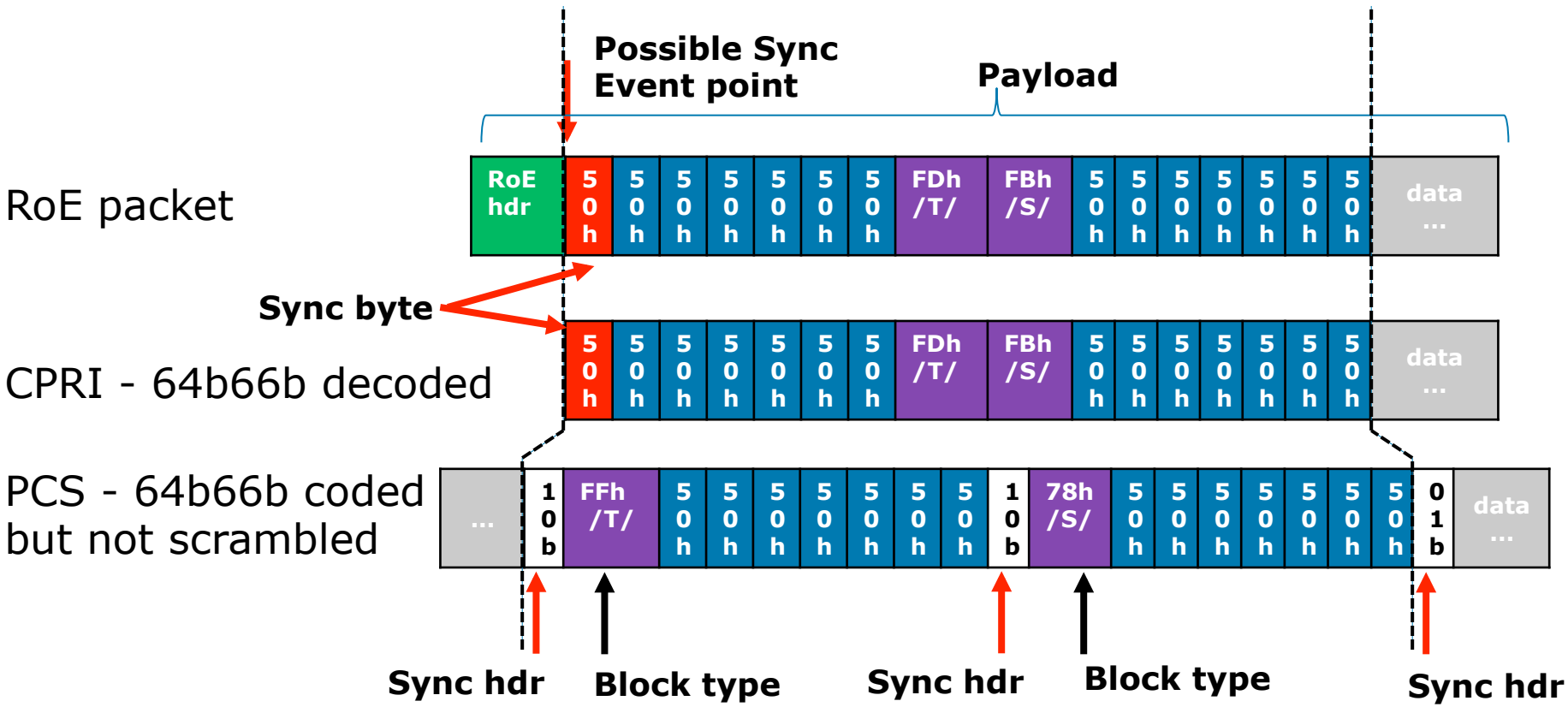
- **Example** from CPRI and a case 64b66b line coding removed (this is just an example illustration using CPRI).
- On the synchronization event the CPRI flow in the mapper has to start with the **Hyperframe Synchronization control word**. In a case of 64b66b that would be the 16 octets as defined in CPRI V7.0 Table 9A.
- Sync Event would be delivered once a radio frame not every hyper frame.

Sync. Control Word	#Z.0.0	Sync Byte	50h
	#Z.0.1	Filling Bytes	50h
	…		
	#Z.0.6		
	#Z.0.7	Terminate Control Character	/T/
	#Z.0.8	Start Control Character	/S/
	#Z.0.9	Filling Bytes	50h
	…		
	#Z.0.15		

Mappers know to track this "synchronization sequence":

50h 50h 50h 50h 50h 50h 50h /T/ /S/ 50h 50h 50h 50h 50h 50h 50h

Example 2/2 – sync mark handling



straw-man poll #_1_

- Agree as a baseline the changes needed for 64b66b line coding add/ removal in structure agnostic mode as listed in pages 6 and 7 of tf3_1610_korhonen_64b66b_2.pdf.

- Mover: Jouni Korhonen
- Seconder: Richard Tse

- (voting members - 1)
- Yes: _9_ No: _0_ Abstain: _4_ (technical motion needs >= 2/3)

- (whole room)
- Yes: _12_ No: _0_ Abstain: _6_ (technical motion needs >= 2/3)

Motion #_2_

- Agree as a baseline the changes needed for 8b10b & 64b66b line coding add/removal in structure agnostic mode as listed in pages 6 and 7 of tf3_1610_korhonen_64b66b_3.pdf.

- Mover: Jouni Korhonen
- Seconder: Richard Tse

- (voting members - 1)
- Yes: _7__ No: ___ Abstain: __2_ (technical motion needs >= 2/3)

(the motion needs to be ratified in the email reflector due the lack of quorum during the meeting)