CPRI Tunneling Packet Formats

Kevin Bross

December 15, 2015

Tunneling Encoding

- Presentation purpose:
  - Provide recommendations for how to handle tunneling of RoE data in an agnostic manner
  - Describes format for handling streams in raw and 8b/10b-decoded formats
    - Special-purpose characters, such as K-characters like K28.5 are treated as exceptions
  - Focus is on CPRI (especially rates 1-7)
  - Packet formats are proposed

Encoding Modes

- Two primary methods are used for encoding data in today's communications:
  - 8b/10b encoding (20% overhead)
  - 64b/66b encoding (3% overhead)
- Consensus from October meeting:
  - 20% gain in removing 8b/10b was worthwhile
  - 3% gain in removing 64b/66b is not worth it
- Therefore, we need 2 modes:
  - Simple (non-decoded) mode
  - 8b/10b decoded mode

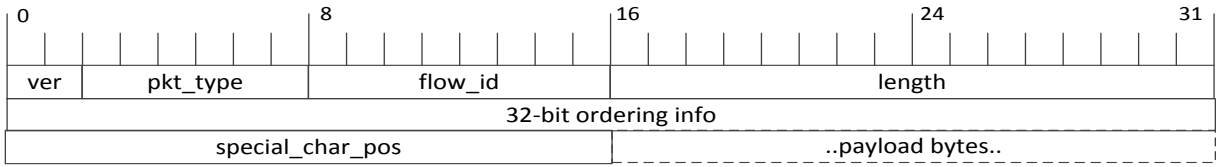
Frequency of Special Characters

- CPRI is primary use case with 8b/10b
- CPRI only uses one special character
  - K28.5 character used for start of frame (SoF)
  - Multiple Ethernet packets between SoF's
- Proposal: support up to one special character per Ethernet packet

- Need to know:
  - Which special character occurred
  - Where special character occurred

Encoding Special Characters (1)

- 16 bits of RoE header added for tunneled RoE data:



  - Need to allocate a pkt_type for tunneled data
- The special_char_pos field indicates the location + 1 of the special character in payload
  - 0 = no special character found
    - Always 0 when simple tunneling is used
  - 1 = first payload character
  - 2 = second payload character, etc.

Encoding Special Characters (2)

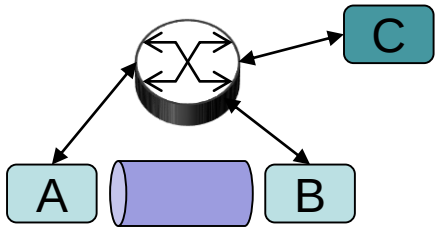
- If special_char_pos is non-zero, that location (1-relative) contains enumerated value indicating the special character
  - Not limited to K28.5 only
- Special 8b/10b characters:

Symbol	Hex
K28.0	0x1C
K28.2	0x5C
K28.4	0x9C
K28.6	0xDC
K23.7	0xF7
K29.7	0xFD

Symbol	Hex
K28.1	0x3C
K28.3	0x7C
K28.5	0xBC
K28.7	0xFC
K27.7	0xF8
K30.7	0xFE

Nodes Involved in Tunneling

- There can be 3 different nodes involved in managing and tunneling data:
  - Endpoint A
  - Endpoint B
  - Management Entity C



- The Management Entity C can be the same node as A or B, but can be separate
  - Allows separate device to control tunnel between two endpoints
- There is no need for RoE tunnel to know which side (A or B) is uplink vs. downlink
- Management packets need a pkt_type

RoE Data Rates

- Protocol does not need to limit itself to a few specific data rates
- Proposal: specify data rates in kbps
  - Allows 32-bit field to handle >10 Gbps
  - Some implementations may support a subset of available rates
- Data rates usually symmetric for RoE
  - Protocol does not need to assume symmetry
- Proposal: specify data rate independently for A → B and B ← A

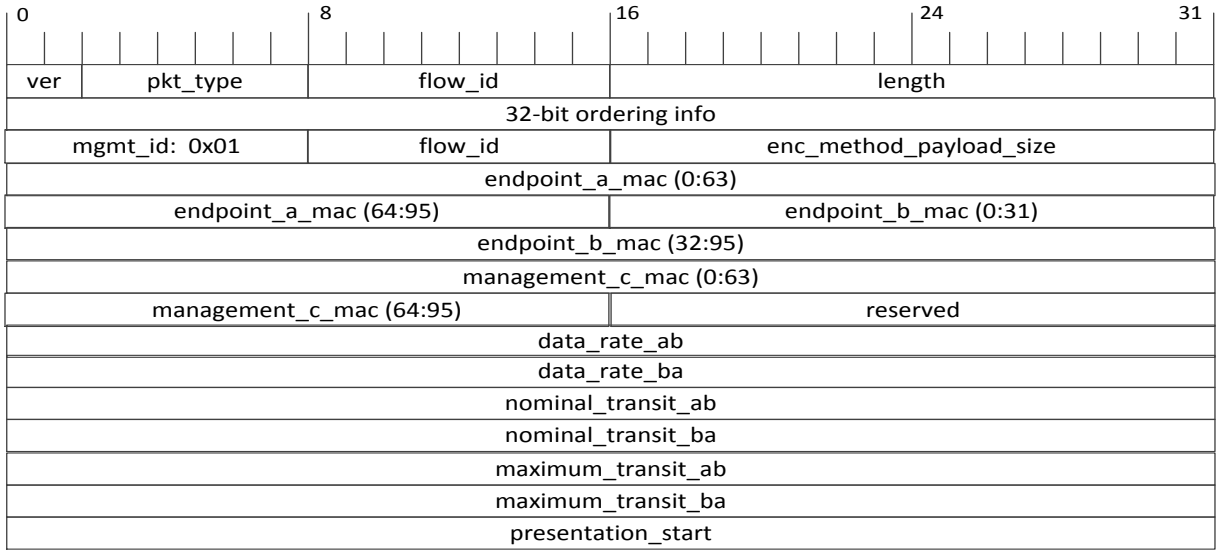
Time Between Nodes

- Protocol needs to know nominal transit time and maximum transit time between nodes
  - IEEE 1588 data may be handled specially and use separate mechanisms than RoE data
  - Difference between nominal and maximum transit time is the nominal buffer time
  - Transit time may be different A → B and B ← A
- Proposal: Tunnel setup protocol should specify nominal and maximum transit time
  - Future contributions may suggest ways to measure nominal time

Initiating a Tunnel

- Management Entity C sends message to Endpoints A and B to start tunnel
- Necessary data:
  - 8b: Management Packet ID: 0x01
  - 8b: Flow ID
  - 16b: Encoding Method and Payload Size
  - 96bx2: Endpoint A Address, Endpoint B Address
  - 96b: Management Entity C Address
  - 32bx2: Data Rate A → B, B ← A
  - 32bx2: Nominal Transit Time A → B, B ← A
  - 32bx2: Maximum Transit Time A → B, B ← A
  - 32b: Starting Presentation Time (at both ends)

Initiating a Tunnel



- Management Packet ID: 0x01
- Flow ID
- Encoding Method and Payload Size
- Endpoint A Address, Endpoint B Address
- Management Entity C Address
- Data Rate A → B, B ← A
- Nominal Transit Time A → B, B ← A
- Maximum Transit Time A → B, B ← A
- Starting Presentation Time (at both ends)

Terminating a Tunnel

- Management Entity C sends message to Endpoints A and B to stop tunnel
- Necessary data:
  - Management Packet ID: 0x02
  - Flow ID