Handling Multiple K-characters

Kevin Bross

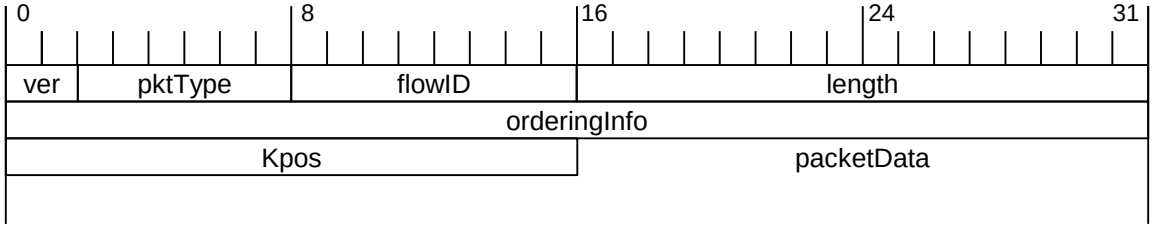
15 April 2016

K-characters

- Both 8b/10b and 64b/66b character encodings have special characters that have no equivalent un-encoded value
  - These special characters are called K-characters
- K-characters are used for special purposes to identify conditions independent of any normal data string
  - For example, CPRI uses K-characters to indicate the start of each hyperframe

Problem Statement

- Structure Agnostic Mapper can only handle 0 or 1 K-characters per packet
  - This works fine for CPRI
  - Other protocols like OBSAI can have multiple K-characters within a given packet
- The "Structure Agnostic" mapper is not really agnostic to the protocols
  - It's currently very CPRI-specific

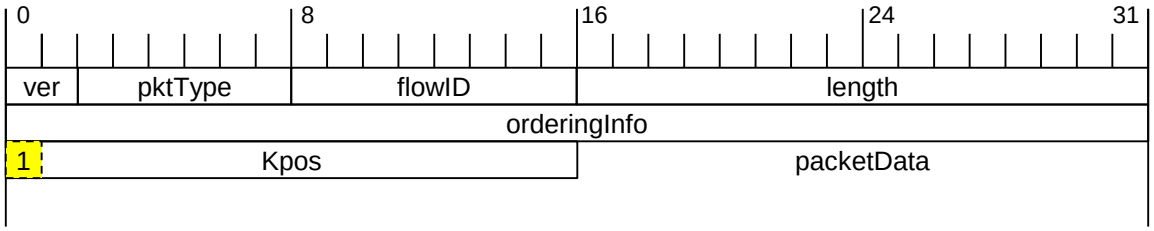


Recommendation

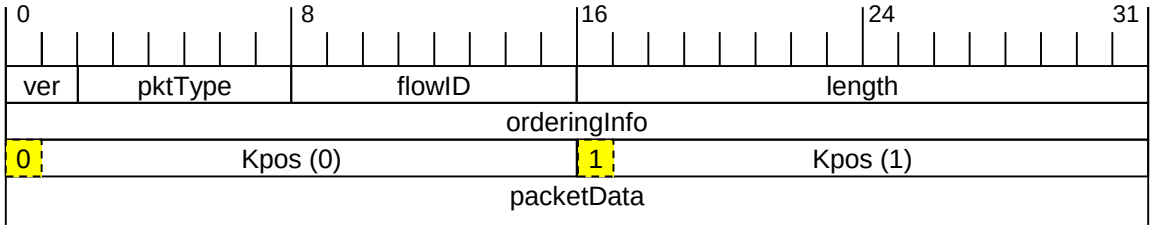
- Provide support to handle multiple K-characters within a packet
  - Enhance current method to indicate position of K-character to allow chaining
- Payload expectation:
  - 1 or more words (16 bits) to indicate Kpos (position of K-character within packet)
    - 0xFF = no K-characters within packet
    - Jumbo packets could use 14 bits for position
    - Chaining bit: MSB clear (0) if another Kpos word immediately follows
  - Start of packetData field shifts out 2 bytes for each additional Kpos field inserted

Example Packets

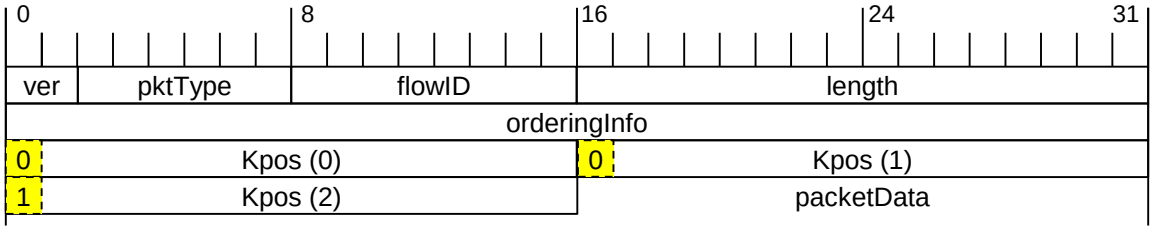
- 0 or 1 K-characters:



- 2 K-characters:

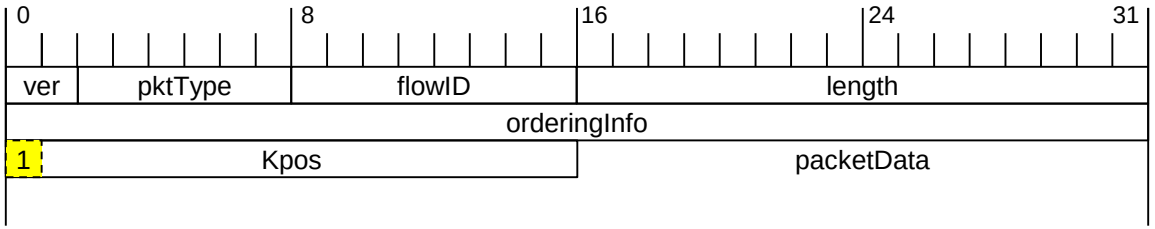


- 3 K-characters:



Summary

- Structure-agnostic mapper is CPRI-specific today
  - Can only handle 0 or 1 K-characters
- Proposal: Use upper bit of Kpos field to indicate chaining of Kpos fields



  - If upper bit is clear (0), another Kpos field immediately follows this Kpos field
  - No impact to packet size (or format) if 0 or 1 K-characters in packetData