Overview of UEC Link Layer Retry (LLR) and Credit-Based Flow Control (CBFC)

Eugene Opsasnick – Broadcom Inc.

LLR and CBFC

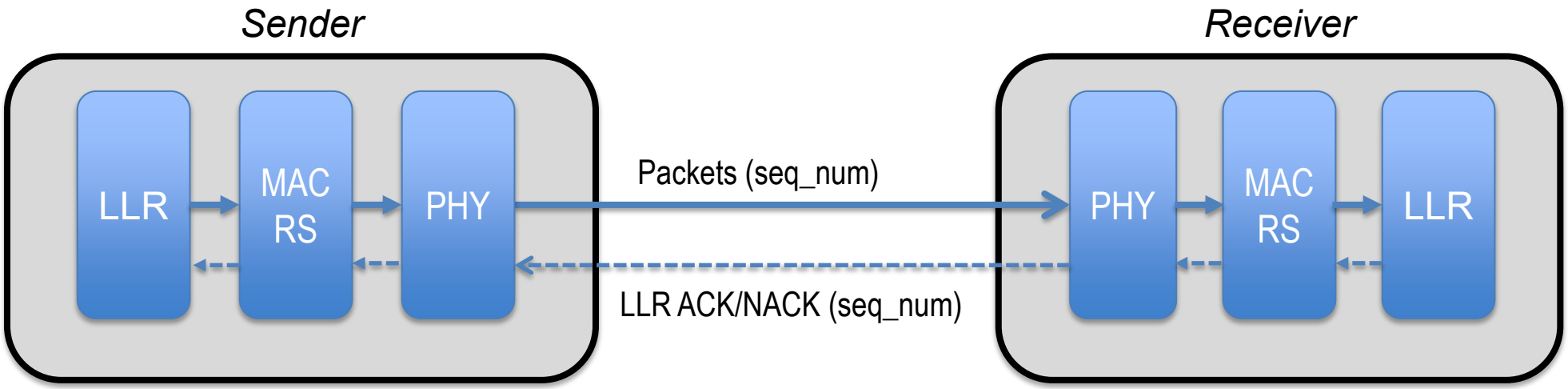
- LLR and CBFC are features that enhance IEEE 802.3 Ethernet at the Link Layer
  - Both are defined in the current Ultra Ethernet Specification, v1.0.2 (www.ultraethernet.org)

- Link Layer Retry (LLR)
  - Can replay packets on a point-to-point link when data corruption occurs on the link
  - Provides faster recovery than end-to-end retransmission protocols as used in TCP
  - Does not prevent packet loss due to network congestion and possible receiver buffer overflow

- Credit-based Flow Control (CBFC)
  - Uses credit system to prevent packet loss due to buffer overflow conditions at the link receiver
  - CBFC has the same objective as Pause and Priority Flow Control (PFC).

LINK LAYER RETRY (LLR)

LLR-enabled Sender and Receiver



LLR Key Concepts - 1

- ACK transmission
  - ACKs are paced and are “cumulative” – not required for every packet
  - ACKs are resent at the specified interval to account for any lost ACK
- NACK transmission
  - Corrupted frames can be NACK’d by receiver as soon as possible
  - Unexpected frames (skipped sequence numbers) also lead to NACK
  - NACK is only sent once to avoid misinterpretation at sender

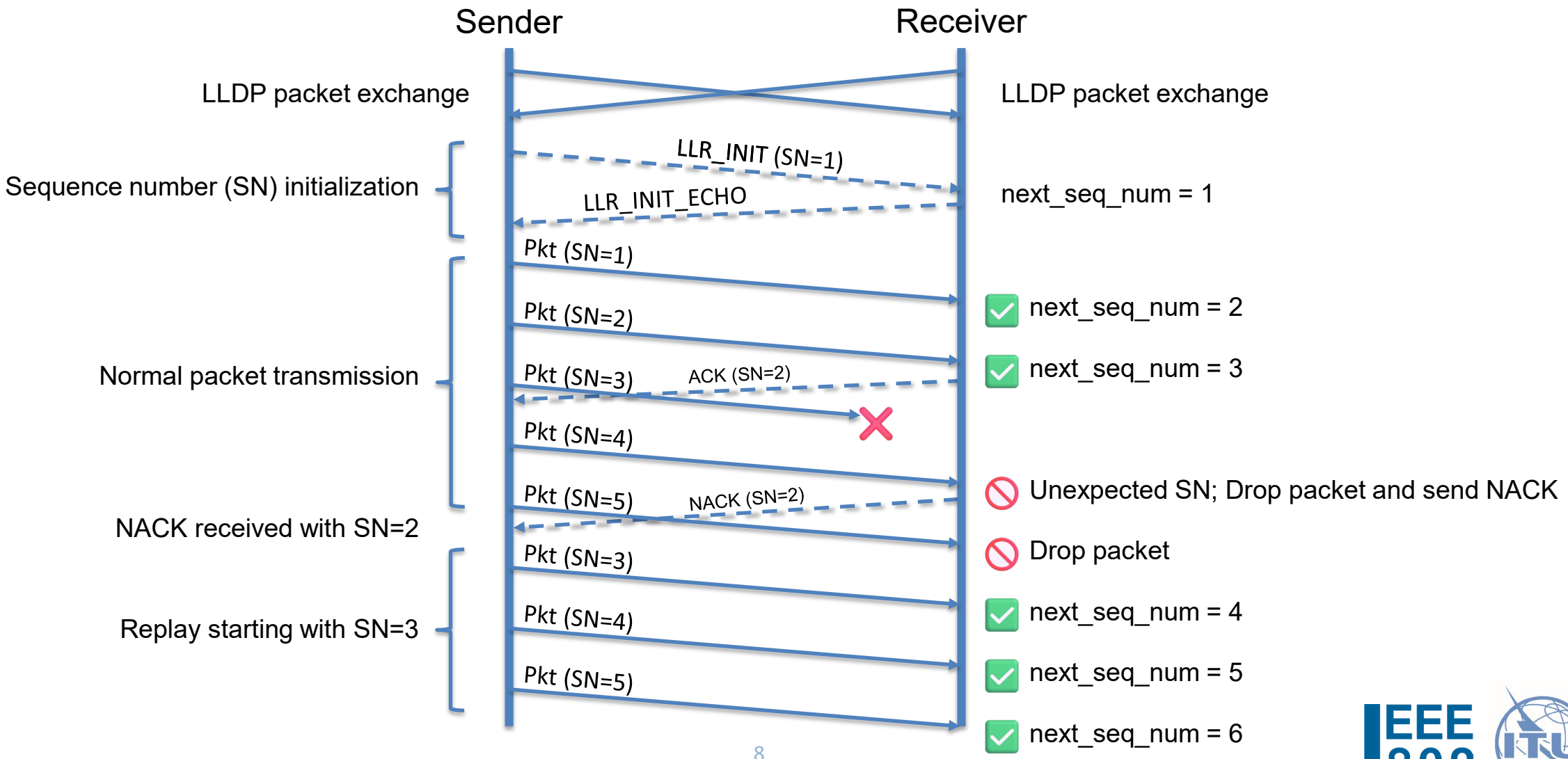
LLR Key Concepts - 2

- Timer-based re-transmission provides protocol robustness for:
  - Lost NACK messages
  - A transmitted packet that is never received at link partner
  - Repeated lost ACKs
- CRC poisoning
  - Sender can apply poisoned CRC to LLR-eligible packets that would fail retransmission and create a livelock condition
    - For example, a packet is corrupted at sender during transmission

LLR Ethernet Extensions

- Modifications to packet preamble - adding a sequence number
- Support for control ordered sets for LLR Messages between the link partners
  - INIT, INIT_ECHO, ACK, NACK
- Control Ordered Sets
  - Uses same format as sequence ordered sets (LF, RF): 8-byte message
  - Distinguished from sequence ordered set by unique 4-bit O code
  - Sent at (almost) any time.
    - Can be sent during IPG, but also allowed to preempt packet transmission

LLR Example


Sender
Receiver
LLDP packet exchange
LLDP packet exchange
Sequence number (SN) initialization
LLR_INIT (SN=1)
LLR_INIT_ECHO
next_seq_num = 1
Pkt (SN=1)
Pkt (SN=2)
next_seq_num = 2
Normal packet transmission
Pkt (SN=3)
ACK (SN=2)
next_seq_num = 3
Pkt (SN=4)
Pkt (SN=5)
NACK (SN=2)
Unexpected SN; Drop packet and send NACK
NACK received with SN=2
Drop packet
Pkt (SN=3)
Pkt (SN=4)
next_seq_num = 4
Replay starting with SN=3
Pkt (SN=5)
next_seq_num = 5
next_seq_num = 6

LLR Reach Limitations

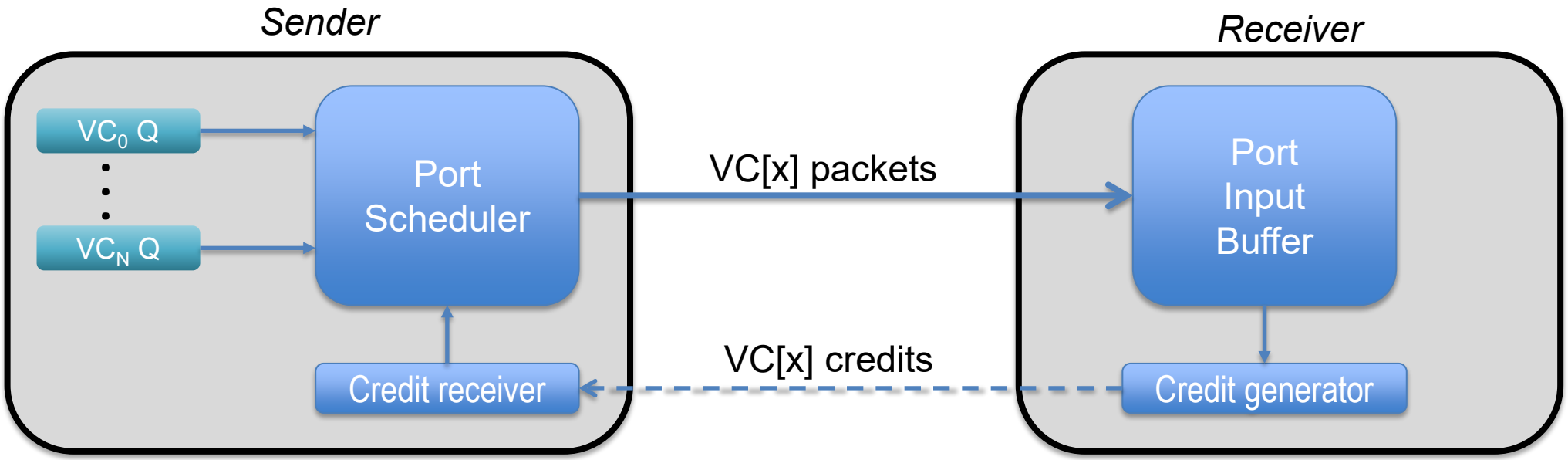
- LLR is targeted at AI scale-up and scale-out networks
- LLR “reach” – the physical distance between sender and receiver
  - Limited by the size of the sender’s replay buffer to maintain full bandwidth
  - Must account for the total round-trip time (RTT), ACK period, and MTU size
- UE Specification does not address a minimum reach
  - Left to implementation choice

CREDIT-BASED-FLOW-CONTROL (CBFC)

What is CBFC and Why it is needed

- CBFC is flow control protocol to enable lossless packet transmission on a point-to-point link between the two link partners
  - Sender only transmits a packet if the receiver has available buffer space
- High-performance networking require reliable packet delivery
  - Can be provided by end-to-end reliable transport layer protocol
  - Can be provided by a lossless fabric without end-to-end retransmissions using CFBC on a hop-by-hop basis

CBFC-enabled Sender and Receiver



CBFC Virtual Channels

- CBFC supports up to 32 VCs
  - LLDP is used to advertise number of VCs supported by receiver and the number of credits available
- VLAN.PCP/DEI or IP.DSCP fields of the packet headers are used to assign a packet to a VC

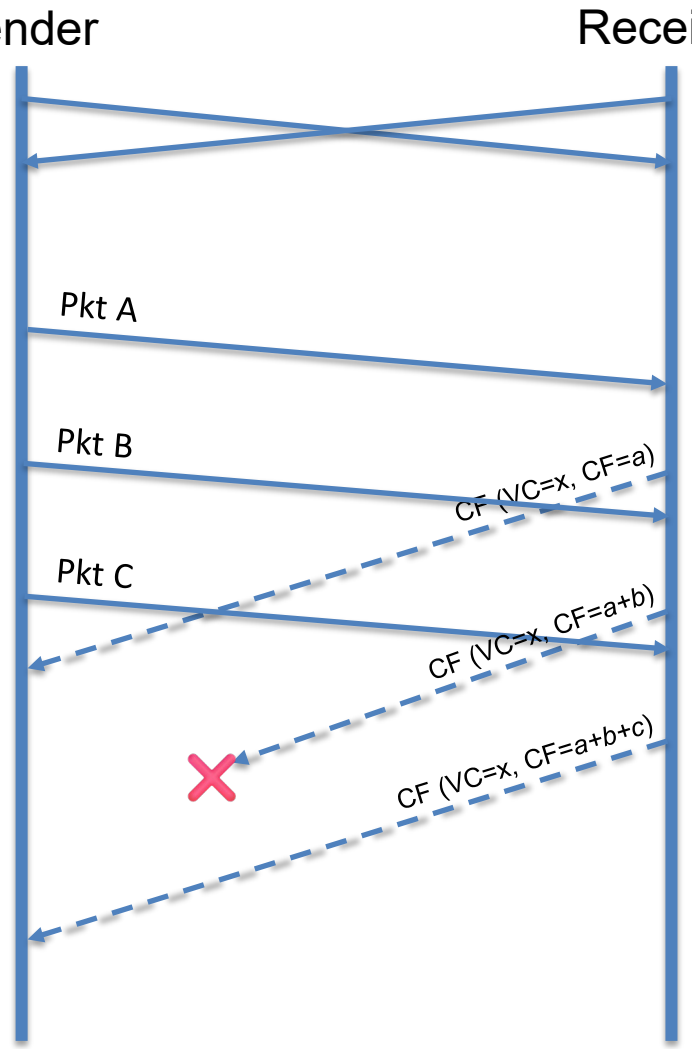
CBFC Key Concepts

- Credits “consumed” at the sender and Credits “freed” by the receiver are tracked on a per-VC basis at both sender and receiver
  - Sender and receiver use cyclic credit counters which “always increment” for resiliency of lost packets or lost credit updates
- Credits are returned to sender with ”Credit Freed” message from receiver
  - CF_Update messages contains the VC number and total number of freed credits
  - A lost CF_Update message is accounted for in the next CF_Update message
- Sender periodically sends a “Credit consumed” message to receiver
  - CC_Update message contains the sender’s current CC_counter values
  - Prevents credit leaks, by allowing receiver to free credits due to lost packets.

CBFC Message Formats

- CF_Update messages use Control Ordered Sets
  - Same format as used for LLR messages (different type field)
  - Contains VC number and CF_counter values for 1 or 2 VCs
  - Must occur frequently enough to keep the link active
- CC_Update messages are a standard 64-byte control packet
  - Uses Ethertype = 0x8808, and UEC OUI value for identification
  - Contains CC_counter values for up to 16 VCs at a time
  - Are sent infrequently

CBFC Example 1



LLDP packet exchange & initialization
S_VC_CC[x] = 0
S_VC_CF[x] = 0
Credit_limit = N

Packet A requires *a* credits and (*a* < N)
S_VC_CC[x] = 0+*a*

Transmit packet B ((S_VC_CC[x] + *b)* < N)
S_VC_CC[x] = *a*+*b*

Transmit packet C ((S_VC_CC[x] + *c)* < N)
S_VC_CC[x] = *a*+*b*+*c*

Credit returned = *a* – S_VC_CF[x] = *a*
S_VC_CF[x] += *a*
Credit_limit = N+*a*

Credits returned = (*a*+*b*+*c*) - S_VC_CF[x] = (*b*+*c*)
S_VC_CF[x] = *a*+*b*+*c*
Credit_limit = (N+*a*+*b*+*c*)

LLDP packet exchange & initialization
R_VC_CC[x] = 0
R_VC_CF[x] = 0

R_VC_CC[x] = 0 + *a*

Pkt A is released from buffer; R_VC_CF[x] = *a*

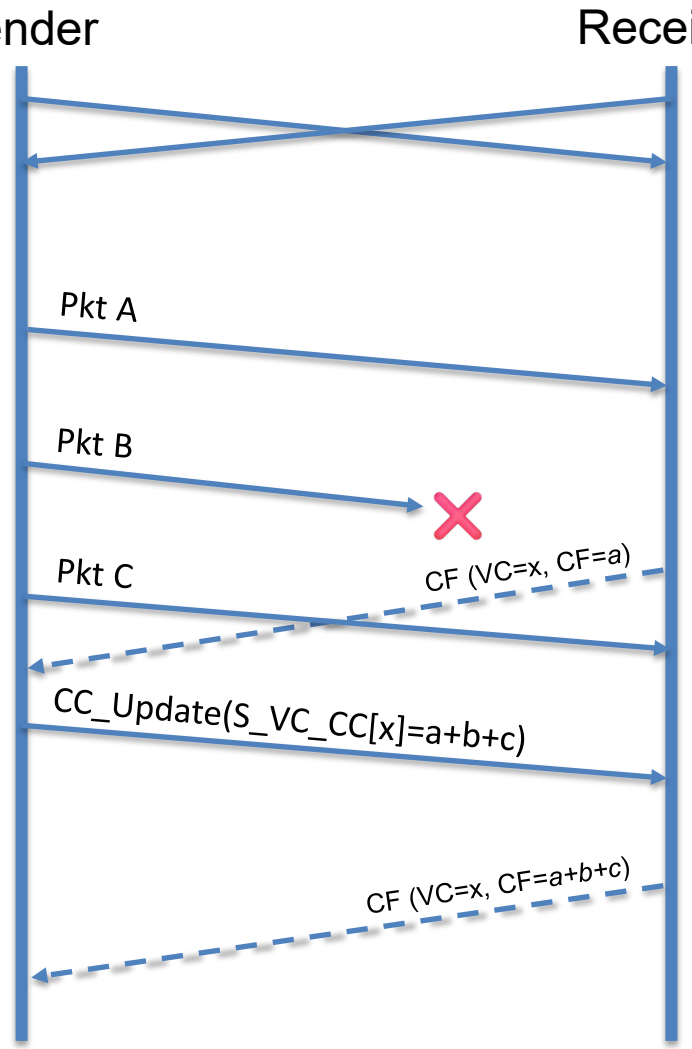
R_VC_CC[x] = *a* + *b*

Pkt B is released from buffer; R_VC_CF[x] = *a*+*b*

R_VC_CC[x] = *a* + *b* + *c*

Pkt C is released from buffer; R_VC_CF[x] = *a*+*b*+*c*

CBFC Example 2



LLDP packet exchange & initialization
S_VC_CC[x] = 0
S_VC_CF[x] = 0
Credit_limit = N

LLDP packet exchange & initialization
R_VC_CC[x] = 0
R_VC_CF[x] = 0

Packet A requires *a* credits and (*a* < N)
S_VC_CC[x] = 0+*a*

R_VC_CC[x] = *a*

Transmit packet B ((S_VC_CC[x] + *b*) < N)
S_VC_CC[x] = *a*+*b*

Transmit packet C ((S_VC_CC[x] + *c*) < N)
S_VC_CC[x] = *a*+*b*+*c*

Pkt A is released from buffer; R_VC_CF[x] = *a*

R_VC_CC[x] = *a* + *c*

Credits retuned = *a* – S_VC_CF[x] = *a*
S_VC_CF[x] += *a*
Credit_limit = N+*a*

S_VC_CC[x] - R_VC_CC[x] = *b* (credit leak)
R_VC_CF[x] += *b* (*a*+*b*)
R_VC_CC[x] = S_VC_CC[x]

Credits returned = (*a*+*b*+*c*) - S_VC_CF[x] = (*b*+*c*)
S_VC_CF[x] = *a*+*b*+*c*
Credit_limit = (N+*a*+*b*+*c*)

Pkt C is released from buffer; R_VC_CF[x] = *a*+*b*+*c*

CBFC and PFC Comparison

- CBFC has some advantages over PFC
  - Can enable more lossless VCs with same amount of input buffer space
  - Sender knows credit usage – can use it for load balancing or adapting routing
  - Not as sensitive to cable length and misconfiguration
    - If cable length is underestimated, PFC can drop packets
    - CBFC would only underutilize the link
  - No packet loss due to lost flow control messages
    - With PFC, if an XOFF message is lost, PFC can drop packets.
  - For low-bandwidth virtual channels, CBFC can simply allocate fewer credits. PFC requires RTT “headroom” to guarantee lossless performance for each priority.

Summary

- LLR and CBFC together enhance Ethernet lossless behavior
  - LLR protects a link when too many bit errors lead to a FEC failure
  - CBFC improves link-level flow control, protecting against packet drops due to receiver buffer congestion
- Main limitation is effective LLR link reach
  - Most suitable for shorter reaches of AI/ML and HPC networks


IEEE
802


ITU