FEC Summary

Ed Boyd, Xingtera

FEC Summary

	Encoder Delay	Decoder Delay	Overall Efficiency*	Complexity
Medium Only	0	Med	80.4%	Easy
Long-Short	Long	Long	87.4%	Medium
Long-Short Parity at End	0	Long	87.4%	Medium
Long-Med-Short	Long	Long	87.4%	Difficult
L-M-S Parity at End	0	Long	87.4%	Difficult
L-M-S + K/2	Long + Short/2	Long + Short/2	87.4%	Most Difficult

*128 Users at 1Gbps upstream

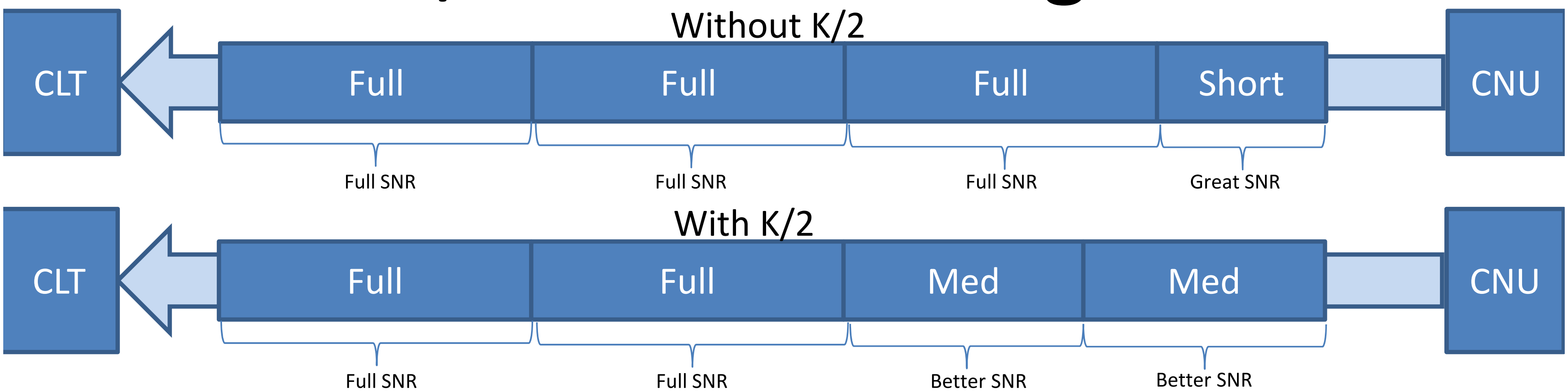
Calculating FEC Overhead

- The CLT must calculate the burst size from the REPORT value.
- 64/66 Overhead
  - Vectors_Bits = 160*65/64*REPORT_Value
- FEC Overhead
  - Medium Only
    - Data_FEC_Bits = Vector_Bits + ROUNDUP(Vector_Bits/4940)*(900+40)
    - Simple Single Divider
  - L-S
    - Data_FEC_Bits = Vector_Bits + INT(Vector_Bits/14300)*(1800+40) + ROUNDUP(REMAINDER(Vector_Bits/14300)/800)*(280+40)
    - Requires 2 Dividers
  - L-M-S
    - TAIL_BITS = TABLE_LOOKUP(REMAINDER(Vector_Bits/14300)
    - Data_FEC_Bits = Vector_Bits + INT(Vector_Bits/14300)*(1800+40) + TAIL_BITS
    - Requires a Divider and a Table Lookup

FEC Summary

K/2

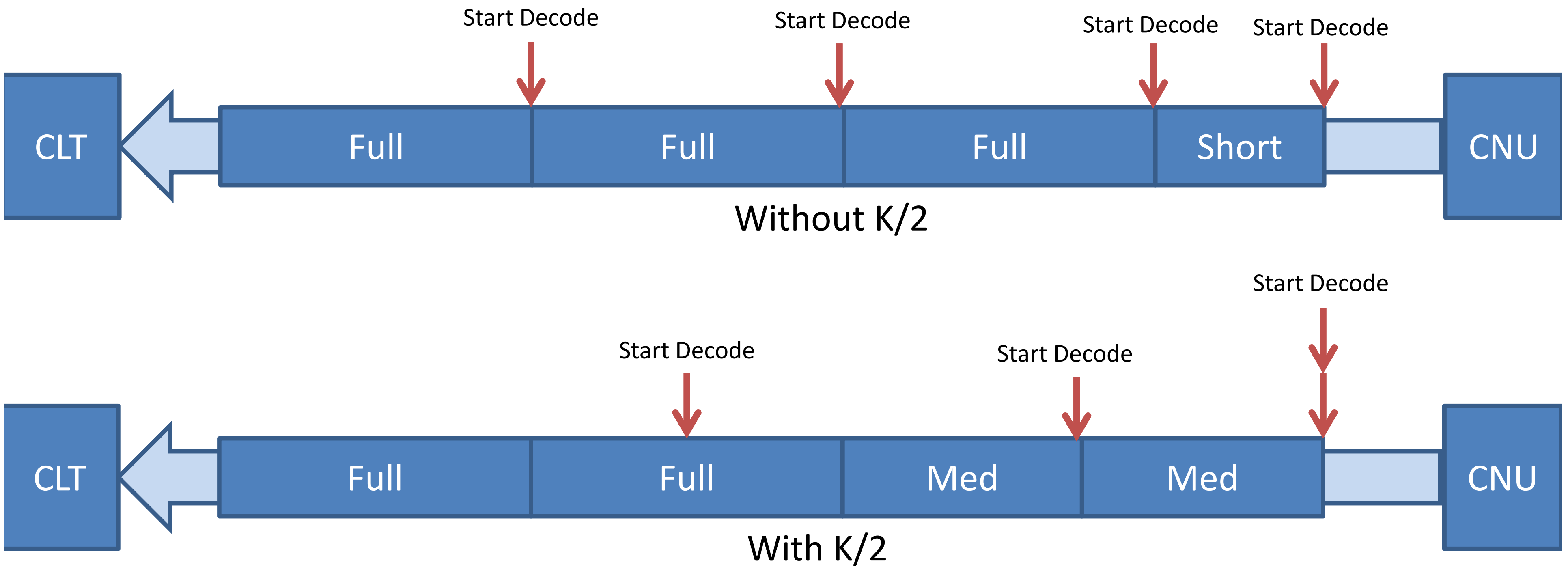
K/2 SNR Advantage?



- Without K/2, Final Short codeword has much better SNR than Full blocks.
- With K/2, last 2 blocks have better SNR.
- Overall SNR is still limited by Full Block size SNR since improvement only on last block on certain block sizes.

K/2 Does not improve overall SNR

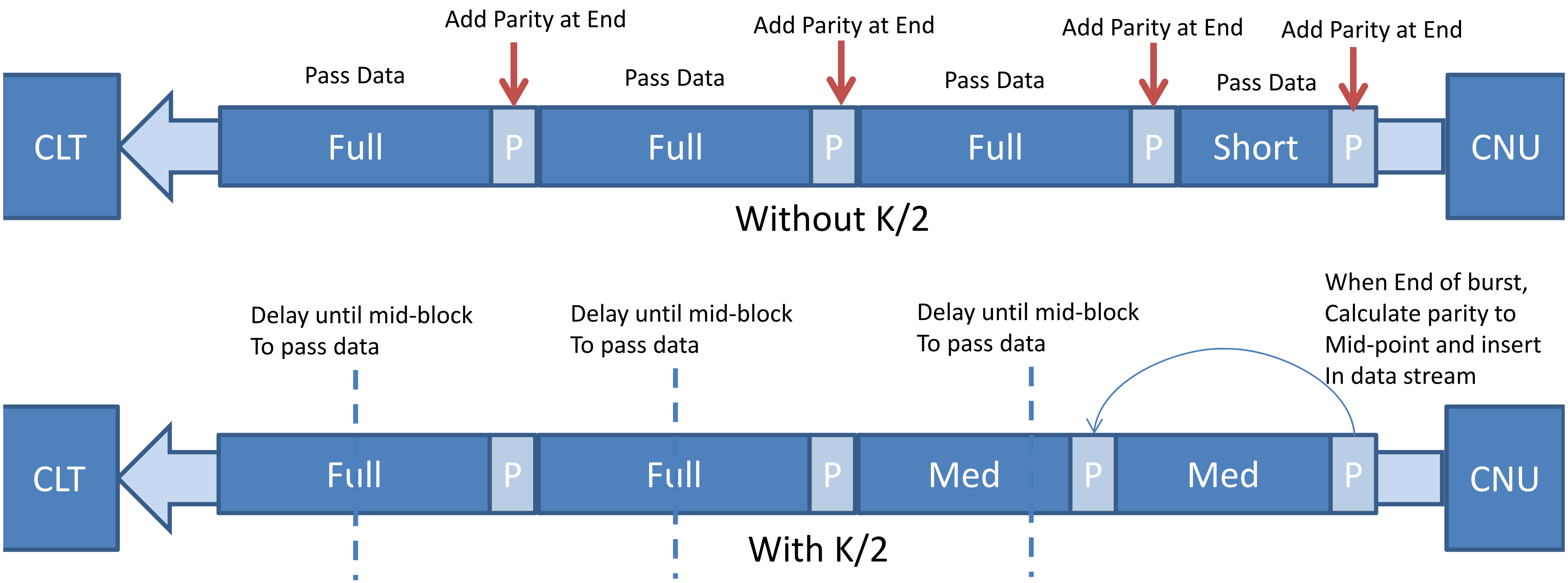
K/2 Decode Delay



- Without K/2, decoding starts after full codeword of data or End of burst marker.
- With K/2, decoding is delayed until half of next codeword or end of burst marker.
- With K/2, decoding the final 2 blocks starts at last code word.

K/2 Increases Decoder Delay

K/2 Encode Delay



- Without K/2, transmit data is not delayed and parity is always after data.
- With K/2, transmitter must delay data until mid-point of next block to determine where parity will be inserted.
- With K/2, parity calculation can't start until end of burst for last 2 blocks and must be inserted in non-end location.

K/2 Increases Encoder Delay and Complexity

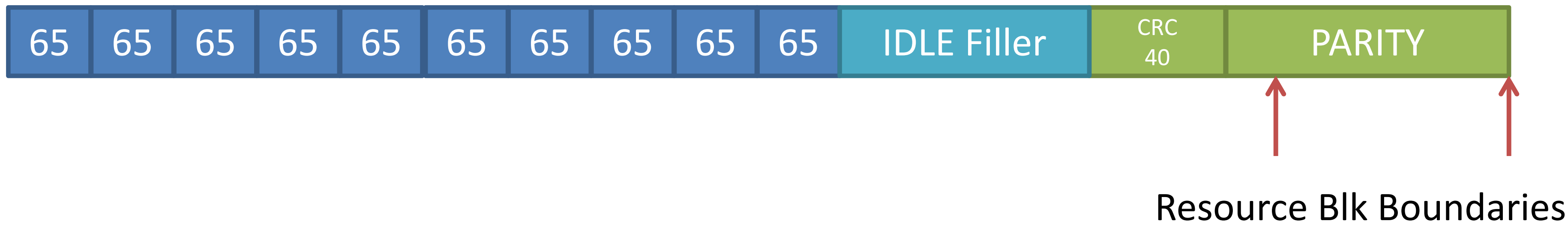
FEC Summary

65 BIT VECTOR ALIGNMENT

Payload Alignment

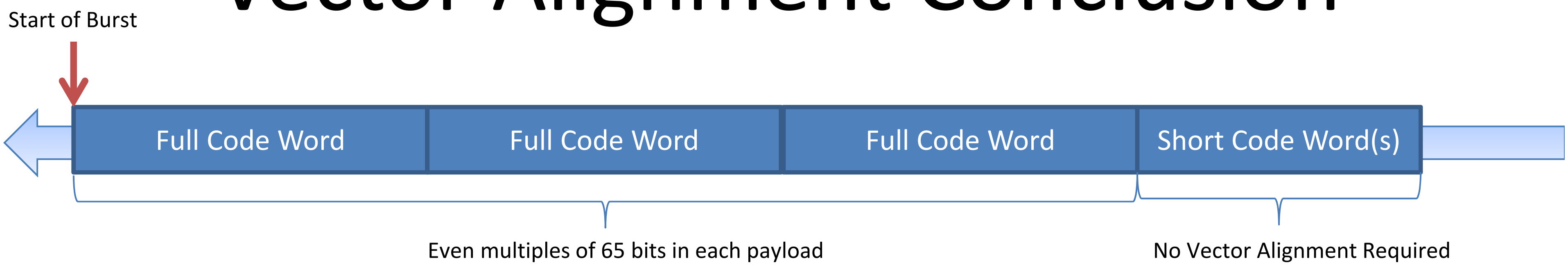
- In the EPoC downstream, the FEC payload carries an even multiple of 65 bit vectors.
  - Allows FEC alignment to set vector alignment.
- Should the EPoC upstream use payload sizes aligned to the 65 bit vectors?
  - Alignment makes it easier to discard a bad FEC block.
  - Alignment Efficiency Penalty
    - Short Efficiency: 70.9% vs 71.4% (75% without CRC-40)
    - Med Efficiency: 84% vs 84.2% (84.8% without CRC-40)
    - Long Efficiency: 88.6% vs 88.6% (88.9% without CRC-40)

Final FEC Block of Burst



- Bursts will not naturally end at Resource Block Boundaries (RBB) so idles must added to the end of the burst.
- Adding data after the parity would not allow end-marker to identify the FEC size.
- IDLE characters must be inserted before the CRC-40 and Parity. (The required data may change the type or amount of FEC parity).
- The IDLE filler inserted to reach the RBB may cause the FEC parity to increase or an additional FEC block added.
- We should NOT require the final FEC block payload to be a multiple of 65 bit vectors so the FEC ends at the RBB. (No alignment issue at end of burst)

Vector Alignment Conclusion



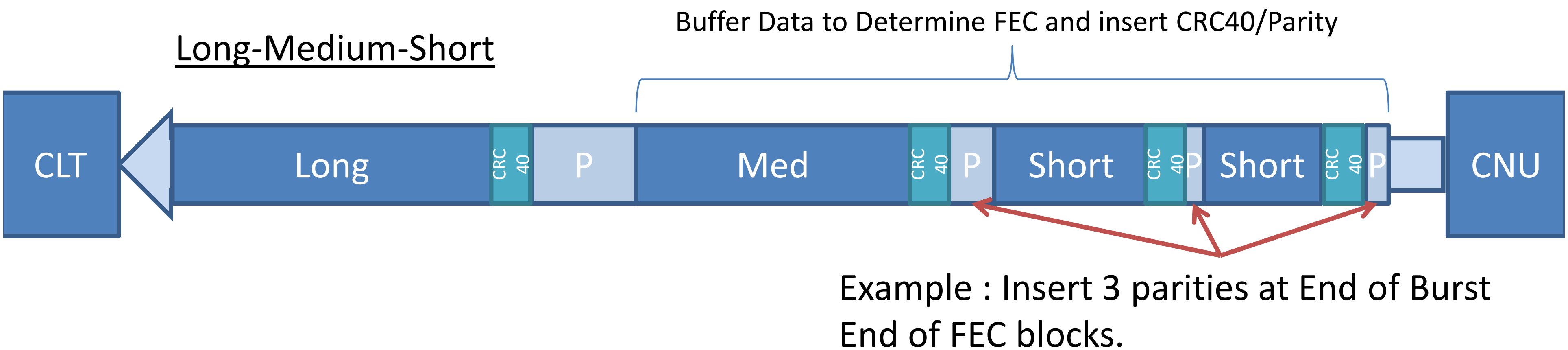
- Alignment of vectors for full size Long code words has a minor (<.1%) impact on efficiency.
- It is easier to discard FEC code words and realign with the vector of the next FEC block if payload is multiples of 65 bits.
- No added complexity since it is the size definition from downstream.
- The End of burst FEC Block(s) will not require vector alignment so the FEC can align with end of the Resource Block.
- There is no benefit for realignment at the end of the burst and the alignment penalty is higher on short code words.

Vector Alignment on Start/Middle Code Words

FEC Summary

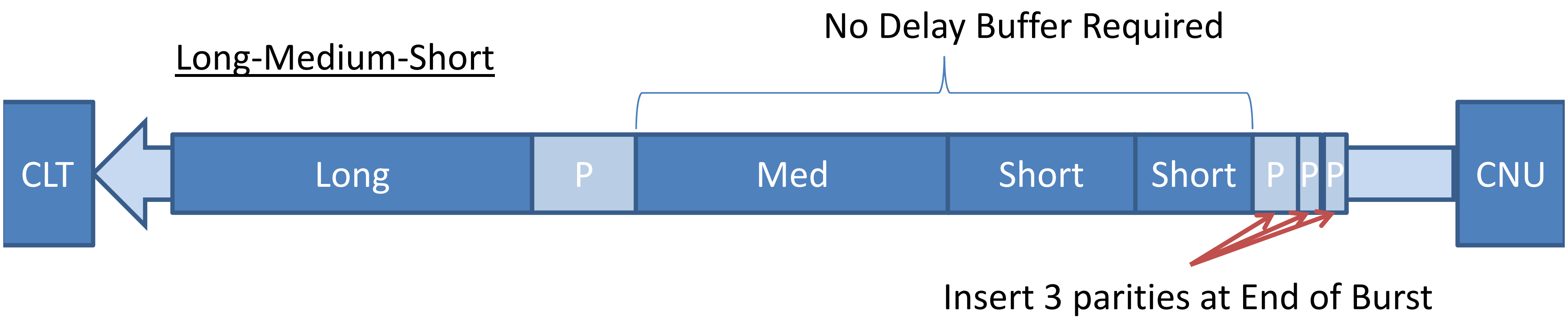
LONG MEDIUM SHORT

Multiple Code Word Complexity



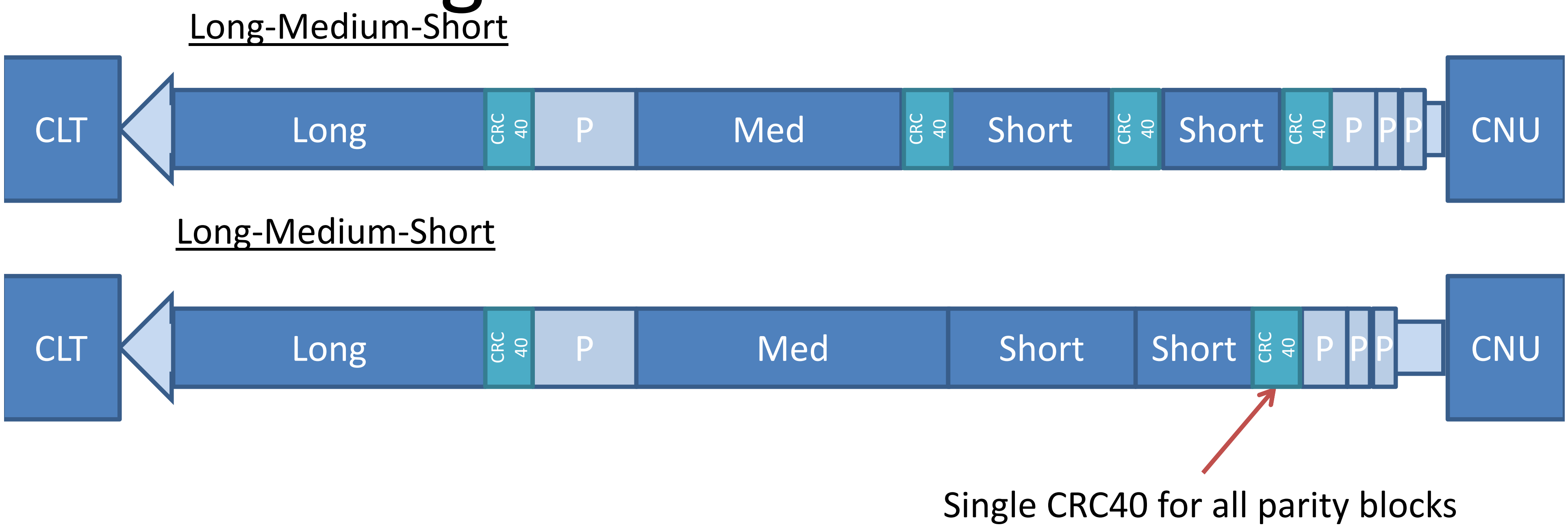
- Medium only has no transmit buffering delay and parity only inserted at the end.
- LMS requires that transmitter buffer data so it can insert the parity/CRC-40 between multiple different size blocks of data.
- K/2 not considered.

Parity at the End



- If parity for 1 or more blocks is always transmitted at the end, transmit data doesn't need to be delayed.
- Multiple Sized Encoders need to calculate parity on multiple data block sizes at the end of the burst.
- K/2 not considered in this slide.

Single CRC40 at the End



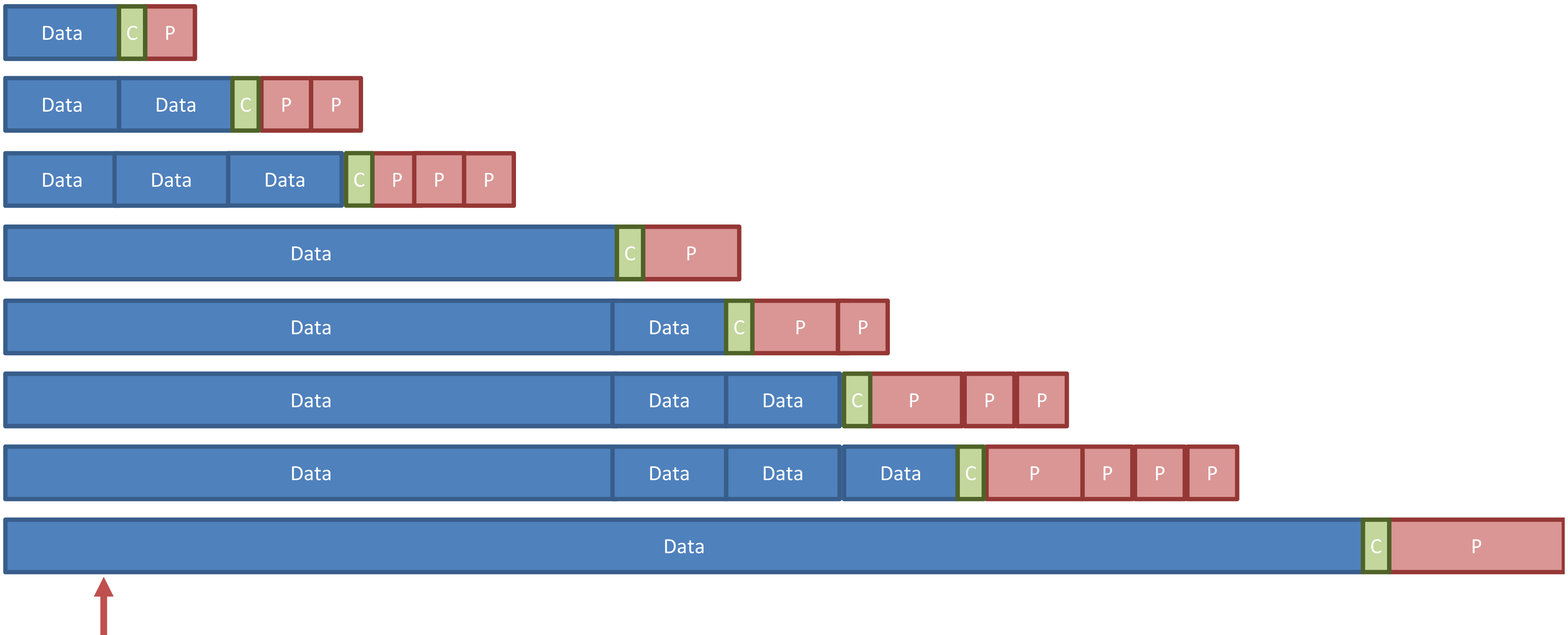
- Inserting CRC-40 at the end of the code words would require transmit buffering and the shifting of data.
- The end of burst is less data than a full long code word so a single CRC-40 would be simpler and more efficient.

Multiple Code words (L-M-S) [@End]

- Assume Parity at the end and a single CRC-40 for a long code word or end of burst blocks
- The tail of a burst can use 1 or more smaller code words to shorten the parity required.
- The code word sizes can be determined by the number of bits in the block.
- Code word alignment is not required on the end of the burst.
- The Look up table below shows the most efficient code words sizes and required parity for any block size.

Min Bits	Max Bits	Long	Medium	Short	CRC40	Parity Bits	Overhead
1	800	0	0	1	40	280	320 bits
801	1640	0	0	2	40	560	600 bits
1641	2480	0	0	3	40	840	880 bits
2481	5000	0	1	0	40	900	940 bits
5001	5840	0	1	1	40	1180	1260 bits
5841	6680	0	1	2	40	1460	1500 bits
6681	7520	0	1	3	40	1740	1780 bits
7521	14300	1	0	0	40	1800	1840 bits

Multiple Code words (L-M-S)



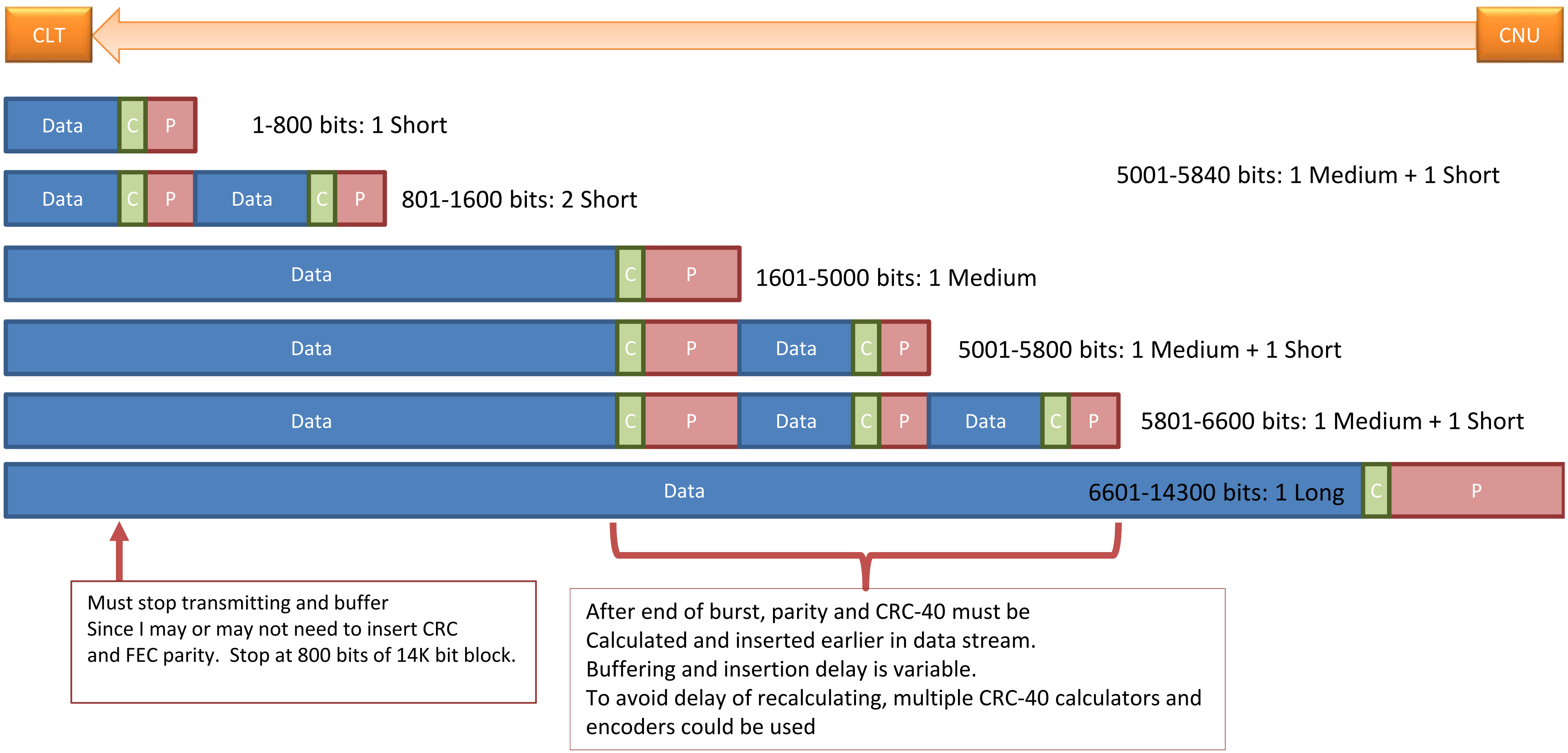
Same Data sent out regardless of size of block.
Single CRC-40 and block of parity avoids variable delay buffer and complexity.

Multiple Code words (L-M-S)

- The tail of a burst can use 1 or more smaller code words to shorten the parity required.
- The code word sizes can be determined by the number of bits in the block.
- Code word alignment is not required on the end of the burst.
- The Look up table below shows the most efficient code words sizes and required parity for any block size.

Min Bits	Max Bits	Long	Medium	Short	CRC40	Parity Bits	CRC+Parity
1	800	0	0	1	40	280	320 bits
801	1600	0	0	2	80	560	640 bits
1601	5000	0	1	0	40	900	940 bits
5001	5800	0	1	1	80	1180	1260 bits
5801	6600	0	1	2	120	1460	1580 bits
6601	14300	1	0	0	40	1800	1840 bits

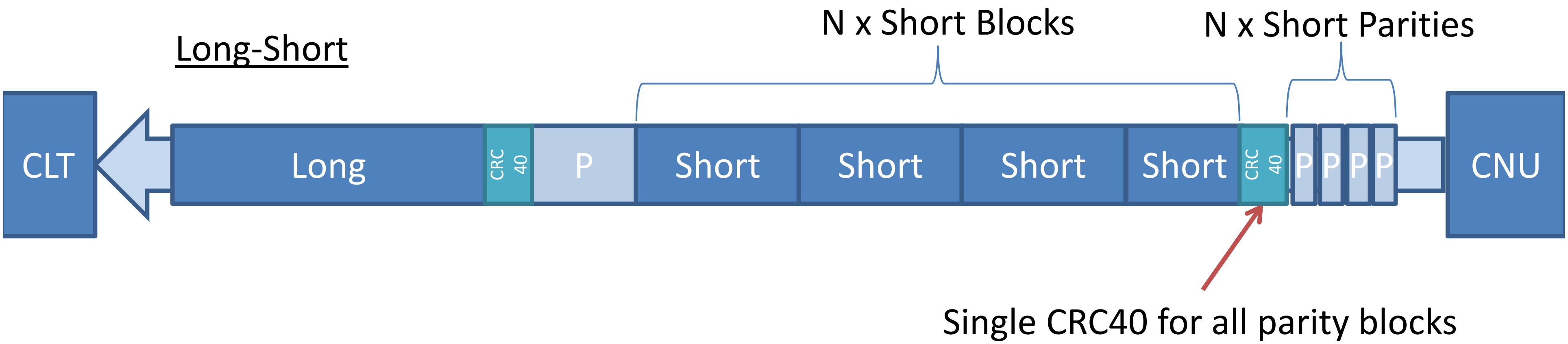
Multiple Code words (L-M-S)



FEC Summary

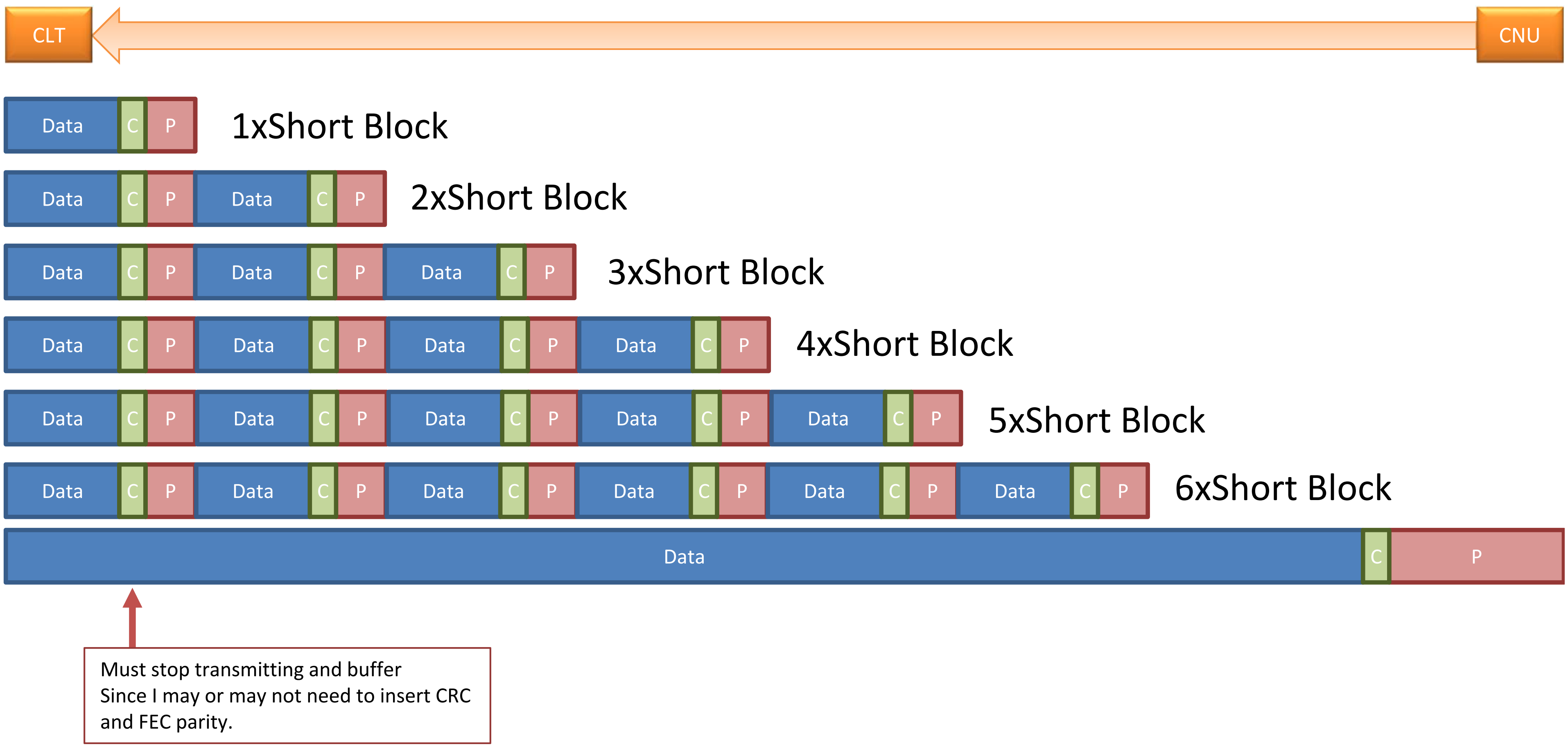
LONG-SHORT

Long-Short (all at end)

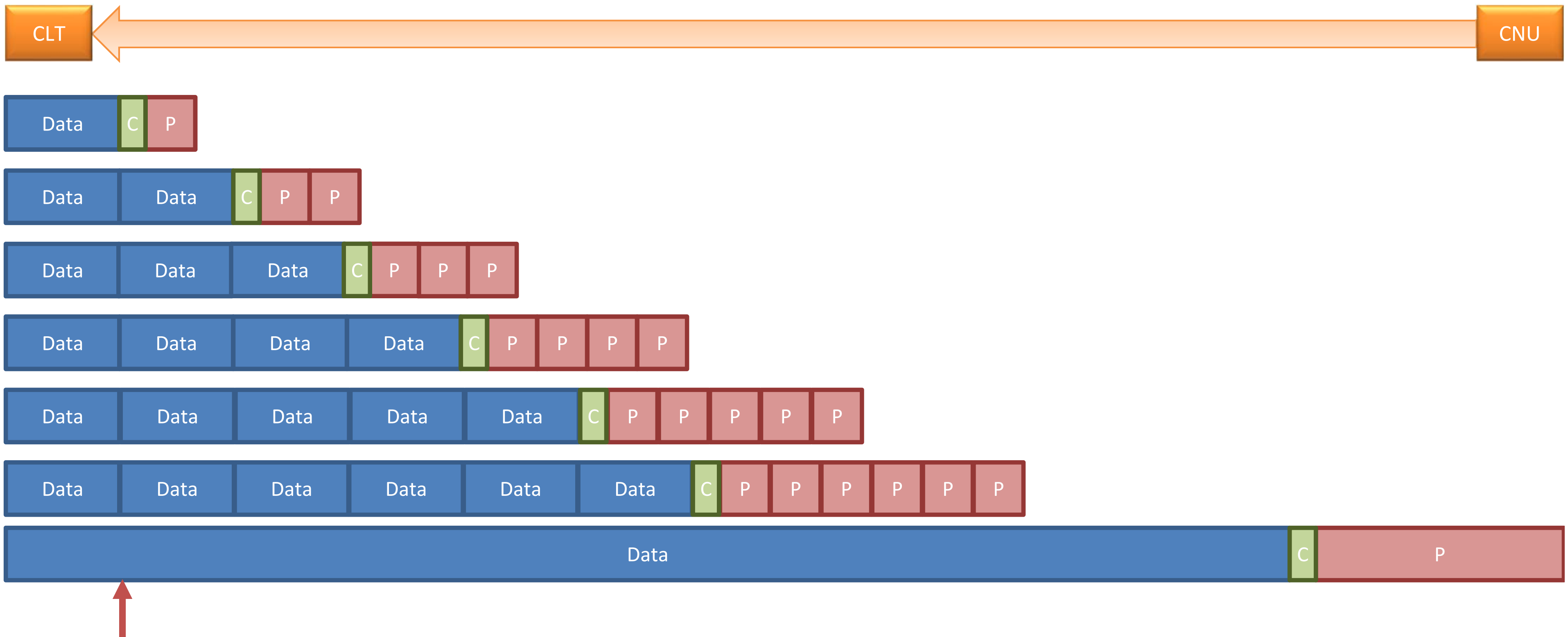


- Transmitter running 2 encoders (short and long) would have no additional delay or jitter.
- If CRC-40 is bad, all end of burst blocks are lost. (Still Less Data lost than bad Long block)

Multiple Code words (L-S) [All at End]



Multiple Code words (L-S)



Same Data sent out regardless of size of block.
Single CRC-40 and block of parity avoids variable delay buffer and complexity.
FEC size increases simply with block size.

Multiple Code words (L-S)

- If the Medium code word size is not used, the following look up table could be used to select the parity.
- Code word alignment is not required at the end of burst.
- Assumes Parity at the End with Single CRC-40.

Min Bits	Max Bits	Long	Short	CRC40	Parity Bits
1	800	0	1	40	280
801	1640	0	2	40	560
1641	2480	0	3	40	840
2481	3320	0	4	40	1120
3321	4160	0	5	40	1400
4161	5000	0	6	40	1680
5001	14300	1	0	40	1800

FEC Summary

IMPACT OF FEC/PARITY AT END

Parity/CRC-40 at End

- PROs
  - Parity/CRC-40 at End is more efficient.
  - Parity/CRC-40 at End has lower delay on transmitter.
  - Parity/CRC-40 at End simplifies transmitter. (To be explained)

- Concerns Raised on Call
  - Parity/CRC-40 at End lowers the ability to detect errors. – Will show that it isn't an issue.
  - Parity/CRC-40 increases delay on receiver. – Will show that it isn't an issue.

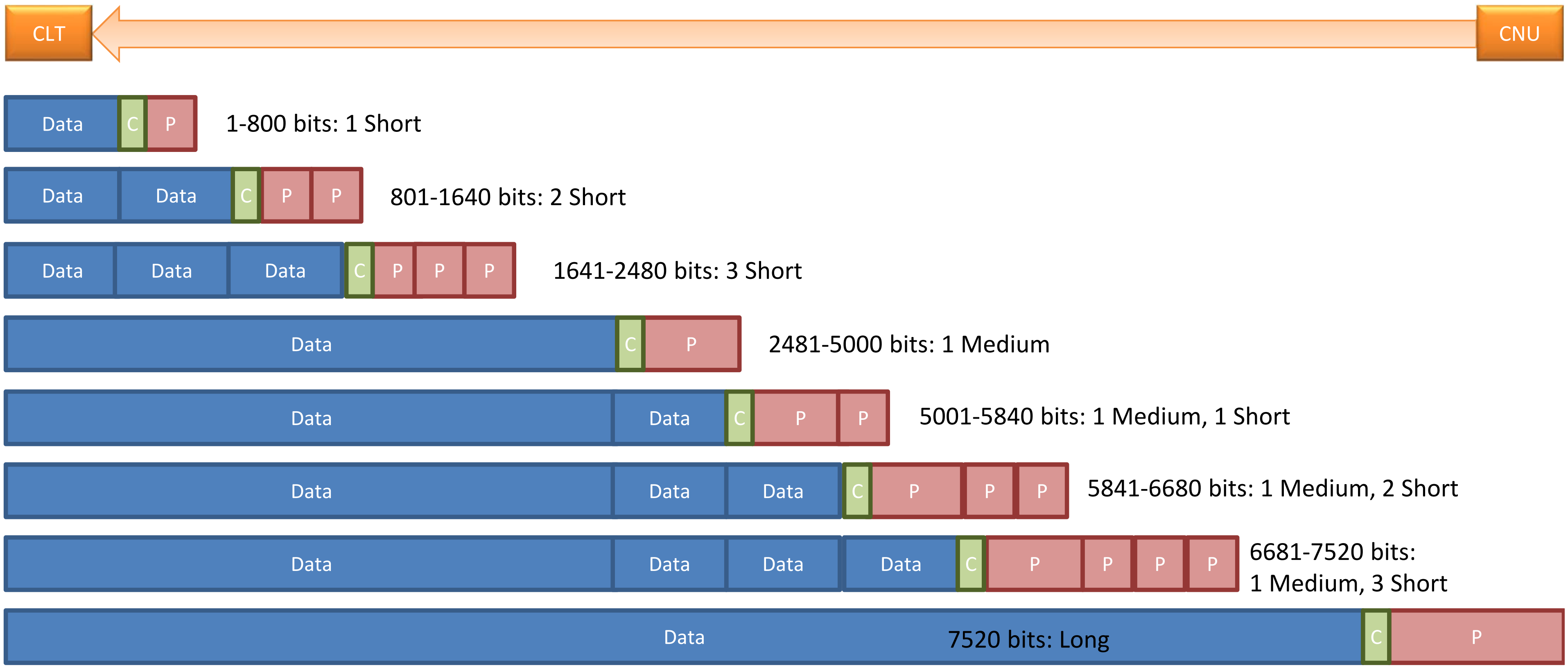
If Parity/CRC-40 at End is less complex, lower delay, and more efficient, we should use it.

Multiple Code words (L-M-S) [@End]

- Assume Parity at the end and a single CRC-40 for a long code word or end of burst blocks
- The tail of a burst can use 1 or more smaller code words to shorten the parity required.
- The code word sizes can be determined by the number of bits in the block.
- Code word alignment is not required on the end of the burst.
- The Look up table below shows the most efficient code words sizes and required parity for any block size.

Min Bits	Max Bits	Long	Medium	Short	CRC40	Parity Bits	CRC40 + Parity
1	800	0	0	1	40	280	320 bits
801	1640	0	0	2	40	560	600 bits
1641	2480	0	0	3	40	840	880 bits
2481	5000	0	1	0	40	900	940 bits
5001	5840	0	1	1	40	1180	1260 bits
5841	6680	0	1	2	40	1460	1500 bits
6681	7520	0	1	3	40	1740	1780 bits
7521	14300	1	0	0	40	1800	1840 bits

Multiple Code words (L-M-S) [@End]



Multiple Code words (L-M-S)

- The tail of a burst can use 1 or more smaller code words to shorten the parity required.
- The code word sizes can be determined by the number of bits in the block.
- Code word alignment is not required on the end of the burst.
- The Look up table below shows the most efficient code words sizes and required parity for any block size.
- The multiple CRC-40's make 2 of the sizes more inefficient than a single block.

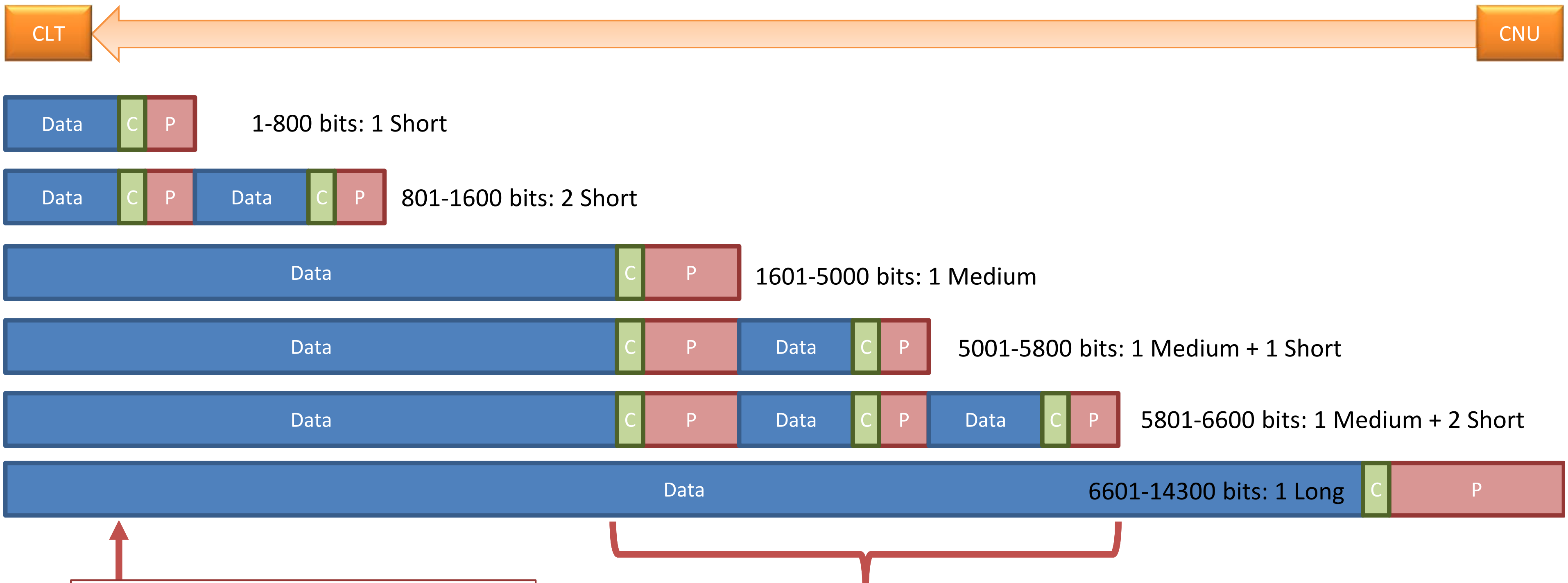
Min Bits	Max Bits	Long	Medium	Short	CRC40	Parity Bits	CRC+Parity
1	800	0	0	1	40	280	320 bits
801	1600	0	0	2	80	560	640 bits
1601	2400	0	0	3	120	840	960 bits
2401	5000	0	1	0	40	900	940 bits
5001	5840	0	1	1	80	1180	1260 bits
5841	6680	0	1	2	120	1460	1580 bits
6681	7520	0	1	3	160	1740	1900 bits
7521	14300	1	0	0	40	1800	1840 bits

Multiple Code words (L-M-S)

- The tail of a burst can use 1 or more smaller code words to shorten the parity required.
- The code word sizes can be determined by the number of bits in the block.
- Code word alignment is not required on the end of the burst.
- The Look up table below shows the most efficient code words sizes and required parity for any block size.

Min Bits	Max Bits	Long	Medium	Short	CRC40	Parity Bits	CRC+Parity
1	800	0	0	1	40	280	320 bits
801	1600	0	0	2	80	560	640 bits
1601	5000	0	1	0	40	900	940 bits
5001	5800	0	1	1	80	1180	1260 bits
5801	6600	0	1	2	120	1460	1580 bits
6601	14300	1	0	0	40	1800	1840 bits

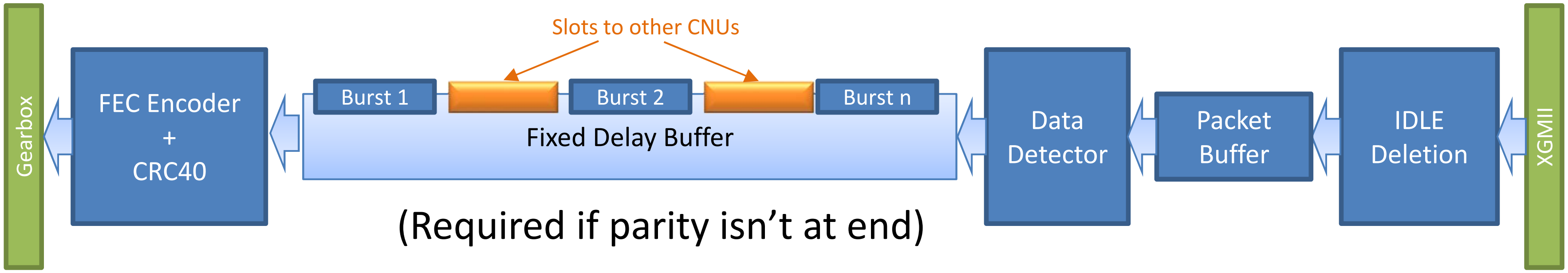
Multiple Code words (L-M-S)



Must stop transmitting and buffer
Since I may or may not need to insert CRC
and FEC parity. Stop at 800 bits of 14K bit block.

After end of burst, parity and CRC-40 must be
Calculated and inserted earlier in data stream.
Buffering and insertion delay is variable.
To avoid delay of recalculating, multiple CRC-40 calculators and
encoders could be used

PCS Transmit Buffer



- If Parity/CRC40 is not at the end, a fixed delay transmit buffer is required in the PCS. [Shown between Data Detector and FEC Encoder]
- The buffer allows the FEC encoder to know the end of burst tail size and place CRC-40 and parity .
- Size of the Buffer
  - The delay will need to be 6,600 bits at the lowest data rate.
  - 6,600 bits @ 250Mbps = 26.4us (Max Delay at low speed)
  - 26.4us @ 1Gbps = 26,400 bits (Max Size at high speed)
- # of Bursts in the buffer
  - Bursts can be 1 to 1.5 us for polling or single packet.
  - Buffer may contain 1 burst or up to 20+ bursts.

Buffer adds complexity and adds delay; we should avoid it.

CRC-40 Coverage

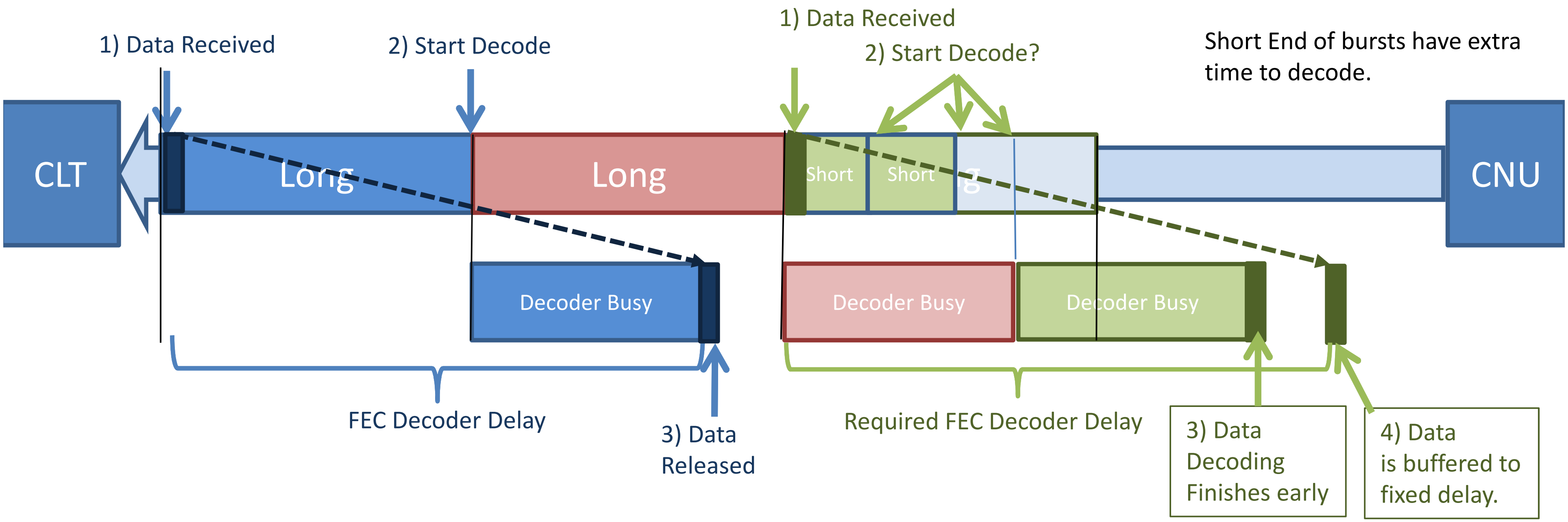
- A single CRC-40 does not decrease the CRC-40 error detectability.

- SNR is about the same for all code word sizes
  - Short (28.8db), Medium (29.1db), Long (29.7db)
    - From Broadcom Victoria Presentation.

- CRC-40 Coverage
  - CRC-40 covers 14K bits on long code words.
  - CRC-40 covers 5K bits for LS or 6.6K LMS for the longest multiple FEC end of burst.

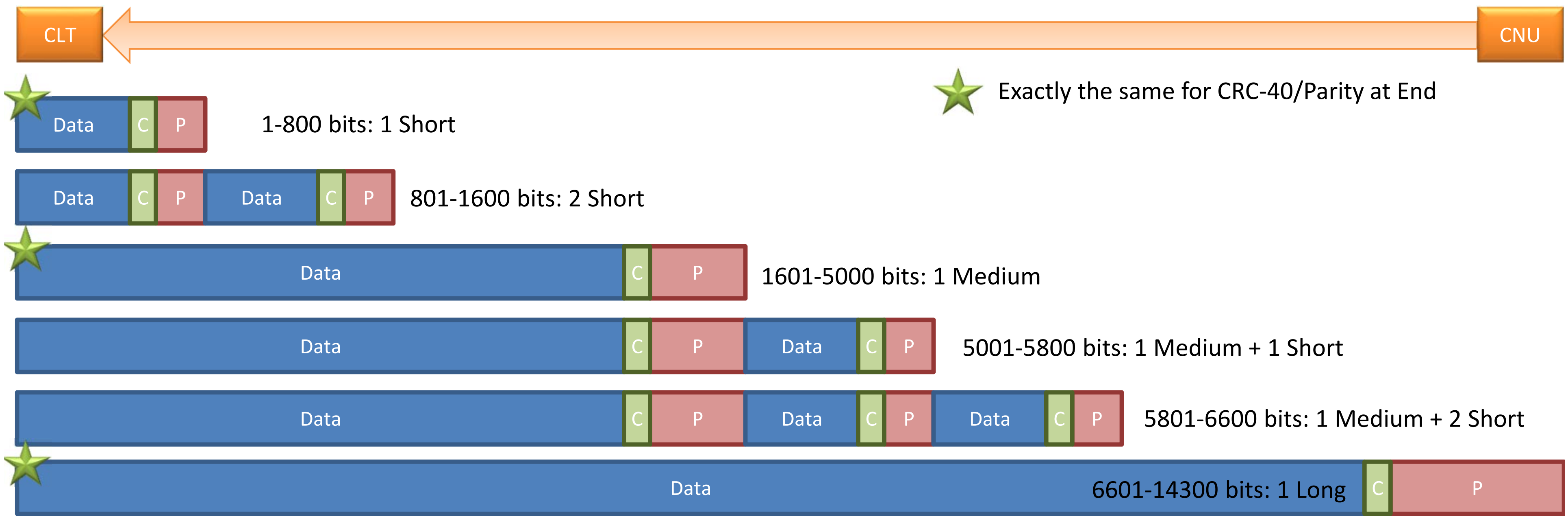
CRC-40 still covers the age of the universe

FEC Receiver Delay



- Data Delay through FEC Decoding (PCS) must be a constant.
  - FEC Decoding Delay must be the same for all data.
  - Longest Delay is Full Size Block Long Block. (Shown above).
- There is no advantage to decoding the end of burst and certain end of burst sizes faster. (data must be buffered to match PHY delay anyway).
- All 3 possible positions for decoding start (Green #2) will have data available early.

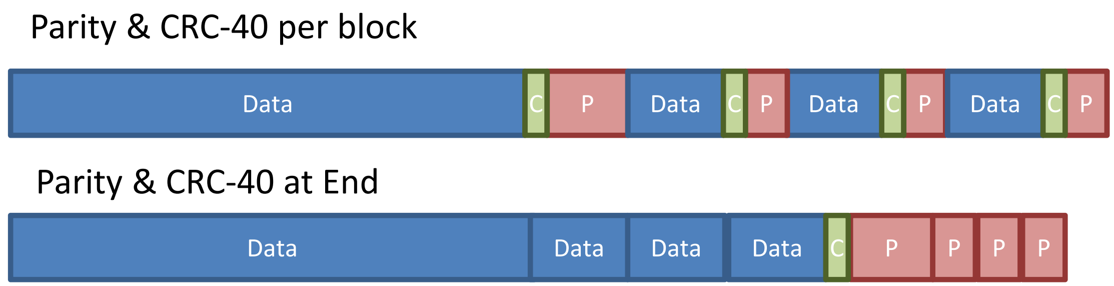
FEC Receiver Delay (2)



- 1 Short, 1 Medium, and 1 Long will start decoding at the end of the burst with or without CRC-40/Parity at the end of burst.
- Starting a few of the middle burst endings early doesn't provide a fixed delay improvement.
- Heavy Cost:
  - Decoder isn't ready until end of burst anyway.
  - 3 times decoder speed up is required to start decoding early.
- No Value:
  - A variable delay on the end of burst doesn't change the fixed PHY delay.

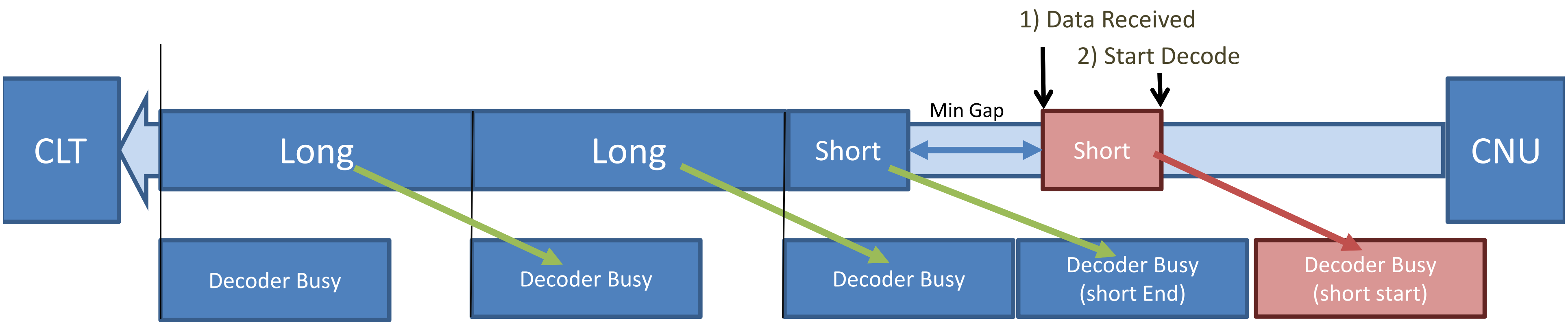
Early Decoding is more complexity for no value

FEC Receiver Throughput



- The placement of the parity does not change the FEC decoding throughput.
- The amount of time to decode the multiple short and/or medium blocks is the same for both cases.
  - Starting earlier doesn't increase the throughput.

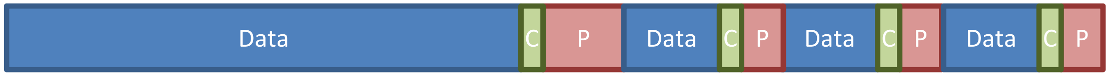
Back to Back Short Burts



- Min Gap between bursts is determined by FEC decoding rate. (it is not a function of the last blocks delay).
- Base Equations
  - LongBlockTime >= LongDecodeTime
  - 2xShortBurstTime + MinGap >= 2xShortDecodeTime
  - LongBlockTime + ShortEndTime + ShortBurstTime >= LongDecodeTime + ShortEndDecodeTime + MinGap + ShortDecodeTime

Summary

Parity & CRC-40 per block



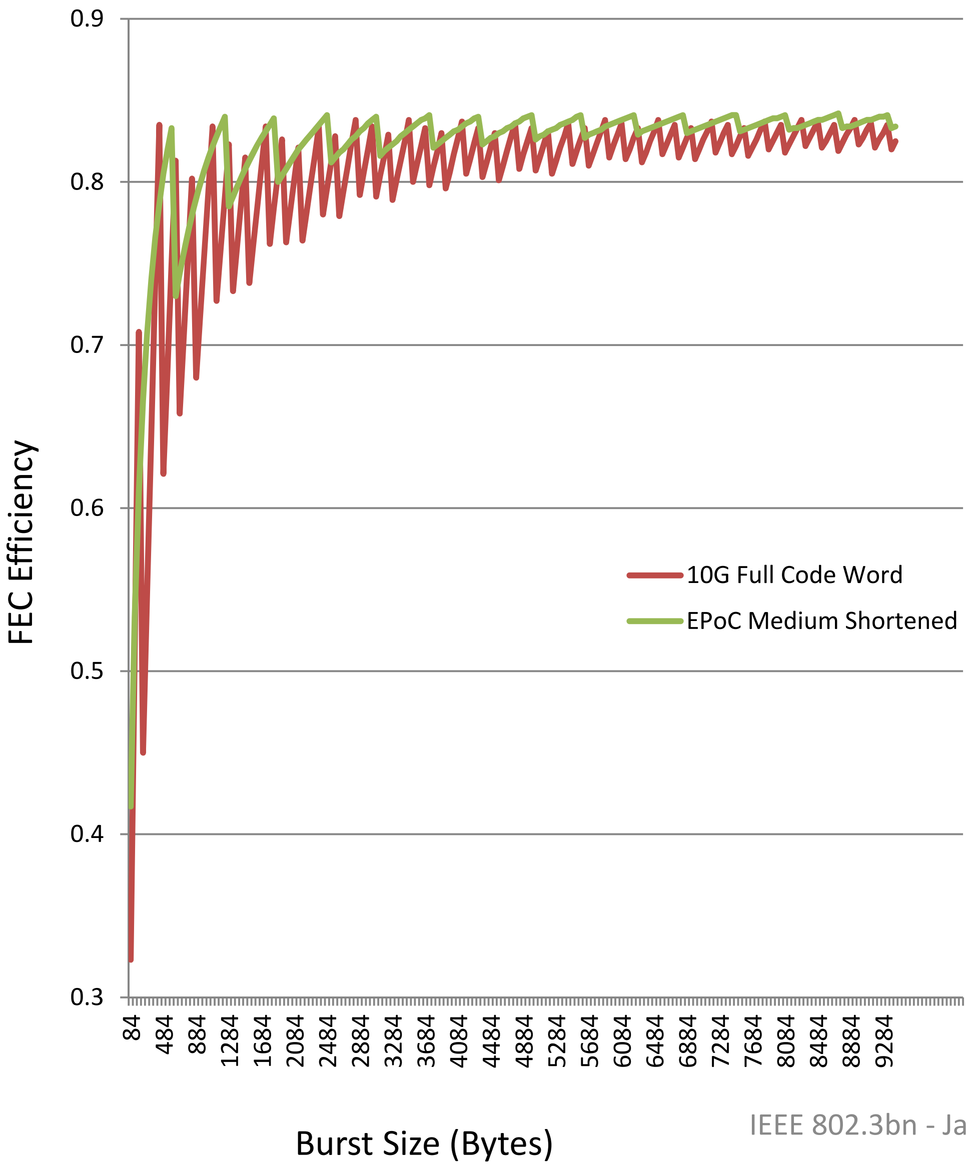
Parity & CRC-40 at End

- Parity & CRC40 at End has no decoder disadvantages
  - Decoding starts at the end of the burst for both methods.
  - Too expensive to start early on short blocks.
  - Even if started early, there is no change to PHY delay or FEC throughput.
- Parity & CRC40 at End has advantages and no disadvantages
  - It is more efficient.
  - Lower Transmitter Delay
  - Significantly reduces transmitter complexity
    - Single CRC40 generator.
    - No transmitter delay buffer to find end of bursts.
    - All PCS blocks [Idle Deletion, Data Detector, FEC Encoding, Gearbox] are working on the same burst in time.

FEC Summary

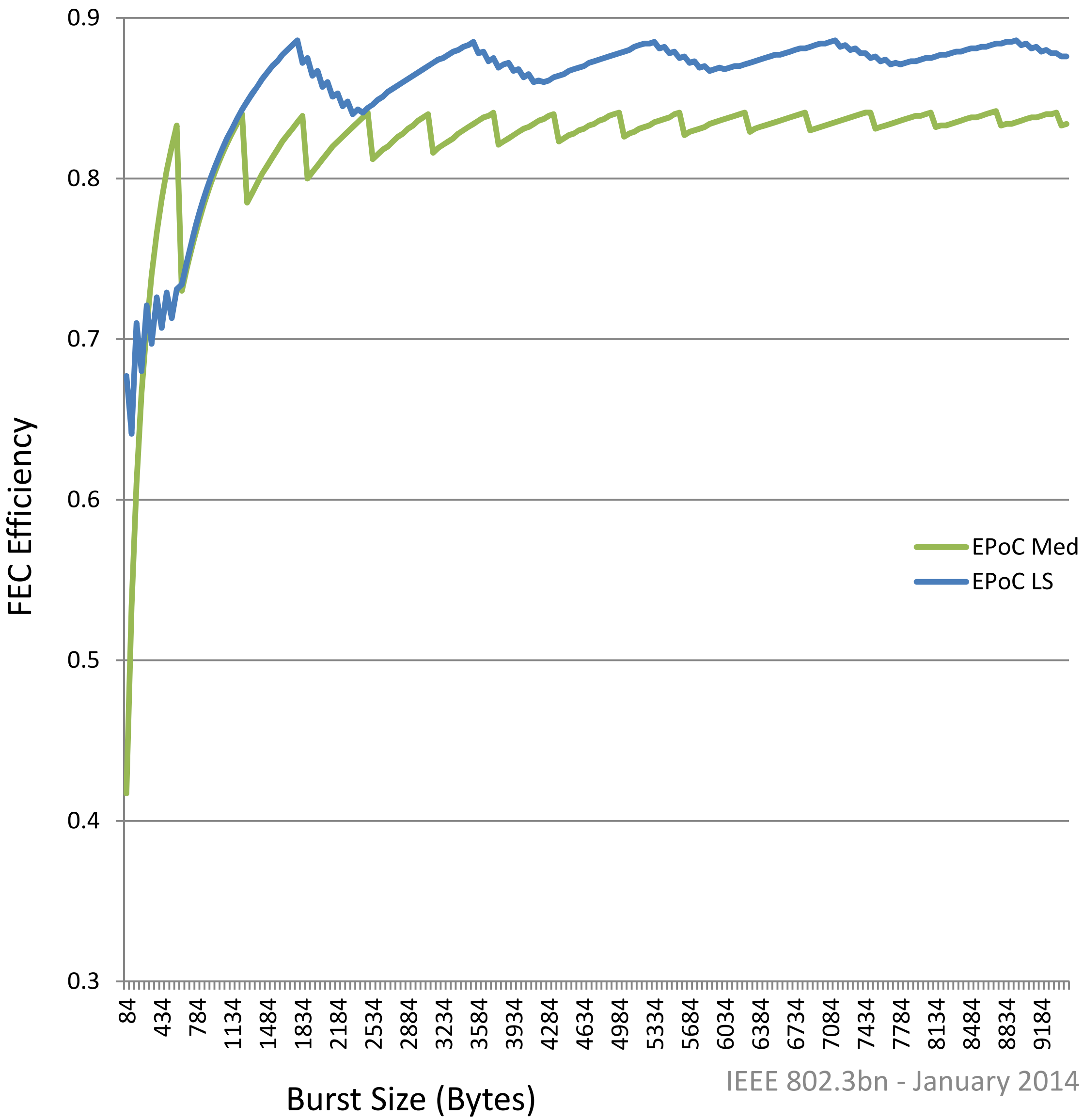
BURST PERFORMANCE

EPON 10G vs EPoC Medium



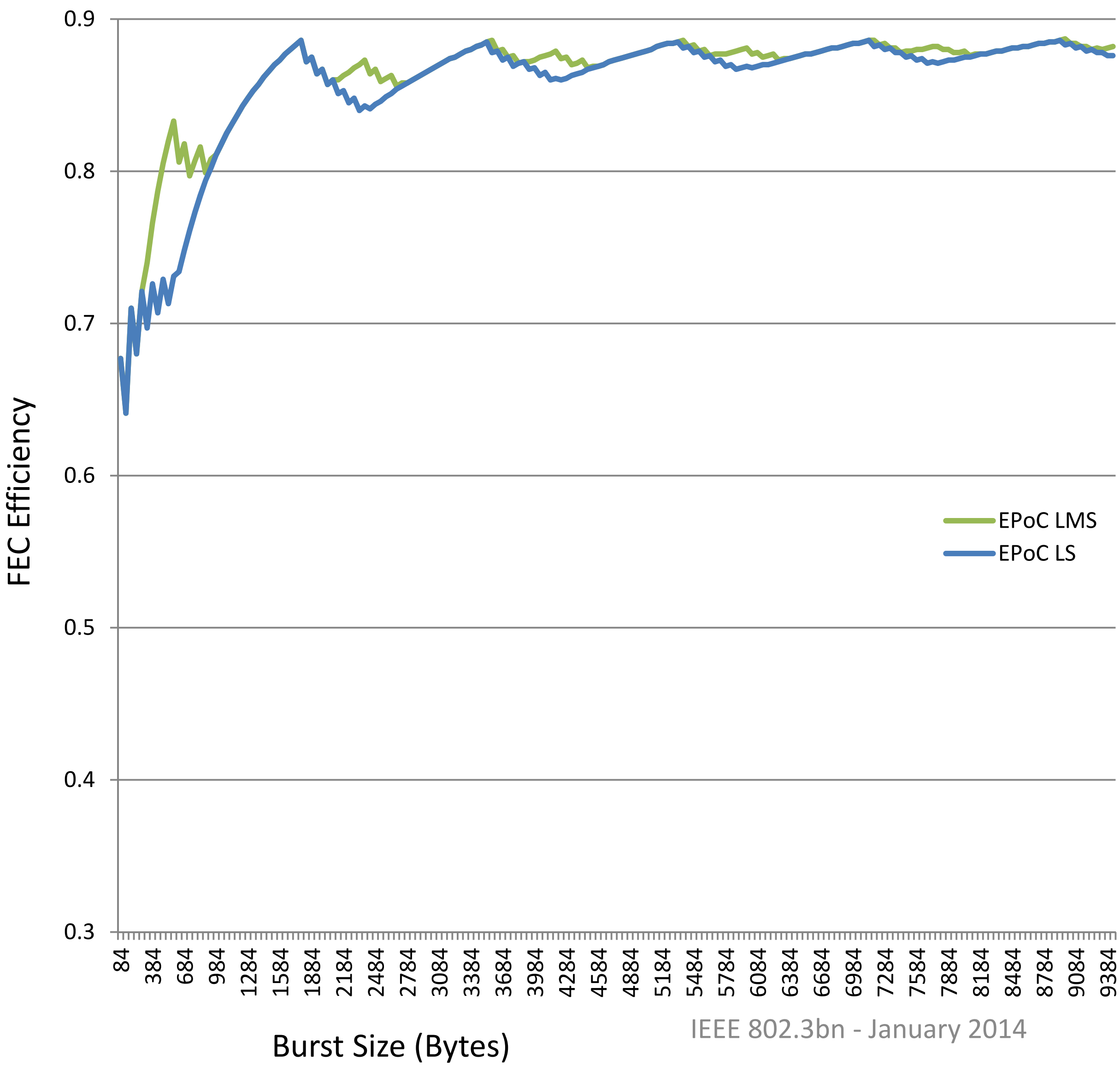
- CRC-40 and FEC Parity added for both cases.
- EPoC Medium Code Word only is generally more efficient than 10G EPON FEC
- Does EPoC needs to improve efficiency over 10G-EPON?
- How can we compare these graphs and get the overall system efficiency?

EPoC Medium vs EPoC LS



- EPoC with a mixture of Long and Short code words improves performance on short and long bursts.
- Efficiency = Vector Payload/(Vector Payload + CRC40 + Parity)
- Is it enough to justify complexity?

EPoC LS vs EPoC LMS



- EPoC with Long, Medium, and Short Code words increases the efficiency of burst sizes in the range of 400 Bytes to780 Bytes.
- Bursts will normally be smaller than 400 Bytes for ACKs, polling, etcs.
- Data Bursts on a loaded system will be larger than 780 Bytes.
- Overall, Little or no performance improvement for LMS over LS.

FEC Summary

SYSTEM EFFICIENCY

Burst vs System Efficiency

- Burst Efficiency does not give a realistic worst-case system efficiency.
  - It is impossible to only have small bursts.
    - If all CNUs are transmitting small bursts and aren't getting enough bandwidth, they will start sending large bursts.
  - It is impossible to only have large bursts.
    - Some CNUs will only have ACKs or polling to send.
- Worst Case System Efficiency
  - Upstream rate and number of CNUs are inputs.
  - Assume that all CNUs except 1 are transmitting the smallest least efficient burst.
  - One CNU is transmitting a large burst to fill in the rest of the data in a 2ms cycle time.

System Efficiency

	64 @ 250Mbps	128 @ 250Mbps	64 @ 500Mbps	128 @ 500Mbps	64 @ 1Gbps	128 @ 1Gbps
Medium	76.6%	68.4%	80.5%	76.5%	82.3%	80.4%
LS	86.1%	83.7%	87.4%	86.2%	88%	87.4%
LMS	86.1%	83.7%	87.4%	86.2%	88%	87.4%

- Medium is around 9% less efficient than Long & Short.
  - ~9%-15% @ 250Mbps, ~9%@500Mbps, ~5-7%@1Gbps
  - Is it worth the additional complexity?
- LMS has no advantage over LS
  - Small and Long bursts set efficiency.
  - 400-800 Byte burst advantage for LMS doesn't show up.
  - No need for LMS.

FEC Summary

CONCLUSIONS

Conclusions

- Medium Code Word only is the simplest solution
  - Efficiency is close to 10G EPON FEC
- Long & Short improves efficiency
  - 5% to 15% system efficiency improvement
  - Parity should be at end to avoid transmit delay/jitter.
  - Single CRC40 should be at the end of burst block
  - Is it worth the complexity added?
- Long & Medium & Short
  - Performance improvement is not worth complexity added over LS.
- K/2
  - Adds delay and complexity with no clear benefit for EPoC
- Alignment to 65 bit vectors
  - Start/Middle of burst blocks should have payload aligned to 65 bit vectors.
  - End of burst blocks should not be aligned.
  - FEC Parities should not be padded to align to 65 bit vectors.

Straw Poll #1

- Do you think that we should include K/2 for the last code word block?
  - Yes:
  - No:

Straw Poll #2

- Which FEC method do you prefer?
  - Medium Only:
  - Long & Short:
  - Long, Medium, & Short:
  - Other:
  - Undecided:

Straw Poll #3

- Assuming L-S or L-M-S, how should we handle the CRC-40's and Parity bits in the end of burst code word blocks?
  - CRC-40 in every block and parity in each block:
  - CRC-40 in every block and all parity at end:
  - Single CRC-40 and all parity at end:
  - Other:
  - Undecided:

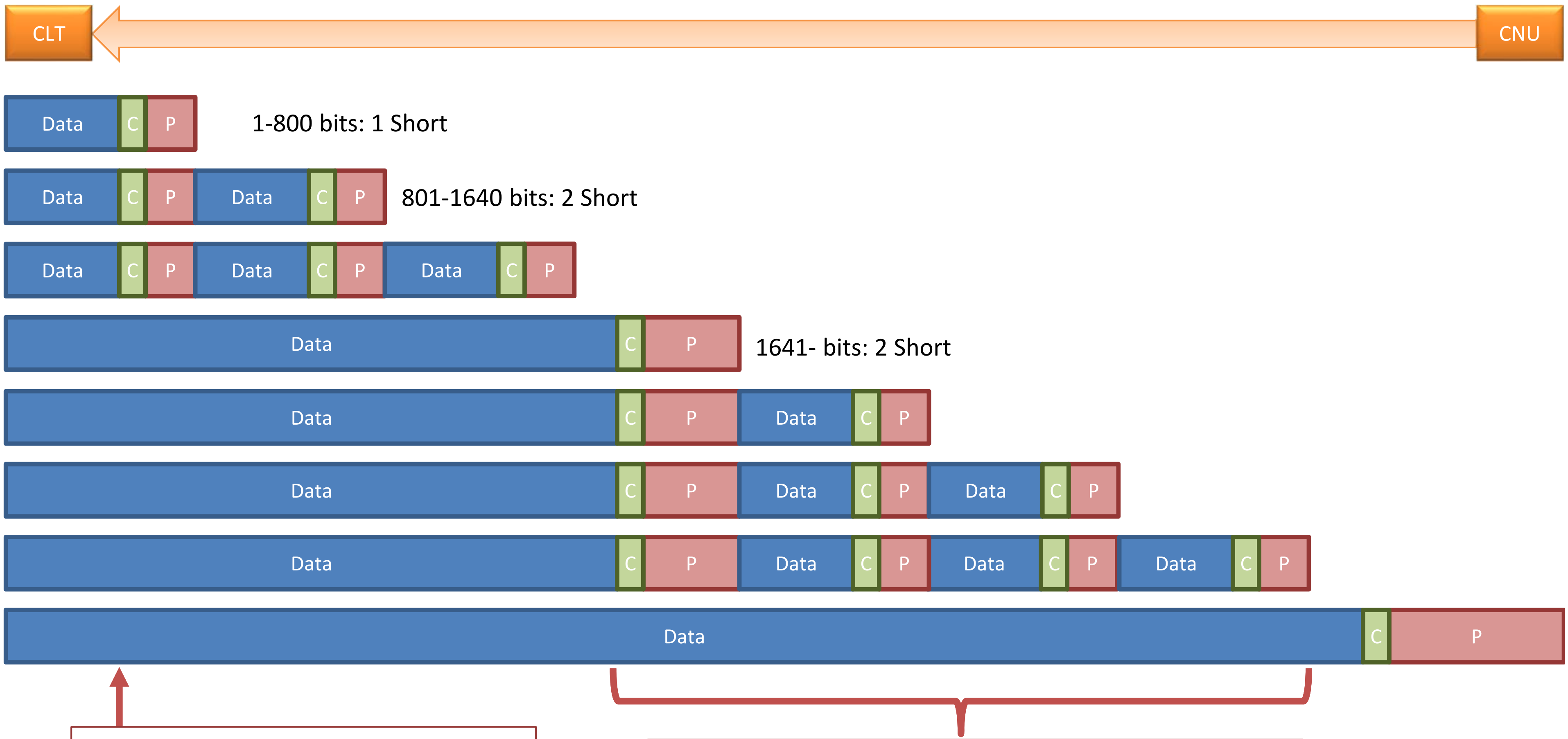
Straw Poll #4

- What should be kept as multiples of 65 bit vectors?
  - All FEC payloads and FEC parities:
  - All FEC payloads:
  - Full Length Long FEC payloads:
  - None:
  - Undecided:

FEC Summary

BACK UP

Multiple Code words (L-M-S)



Must stop transmitting and buffer
Since I may or may not need to insert CRC
and FEC parity. Stop at 800 bits of 14K bit block.

After end of burst, parity and CRC-40 must be
Calculated and inserted earlier in data stream.
Buffering and insertion delay is variable.
To avoid delay of recalculating, multiple CRC-40 calculators and
encoders could be used.