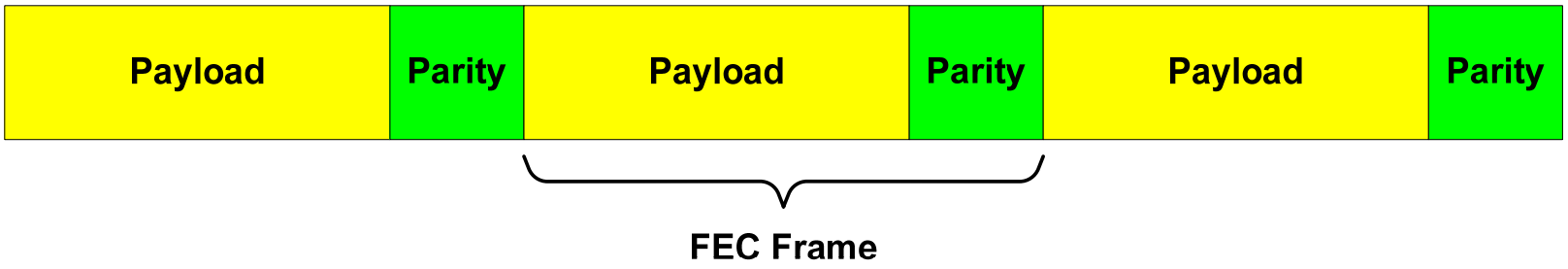
Shortened FEC Frames for 10GEPON: Is there any advantage?

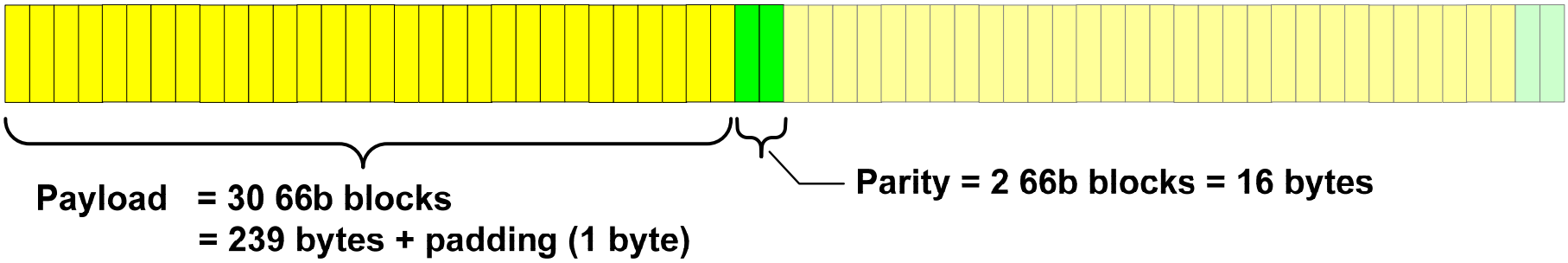
Glen Kramer
glen.kramer@teknovus.com

FEC Frame Format

- FEC frame consists of payload and parity

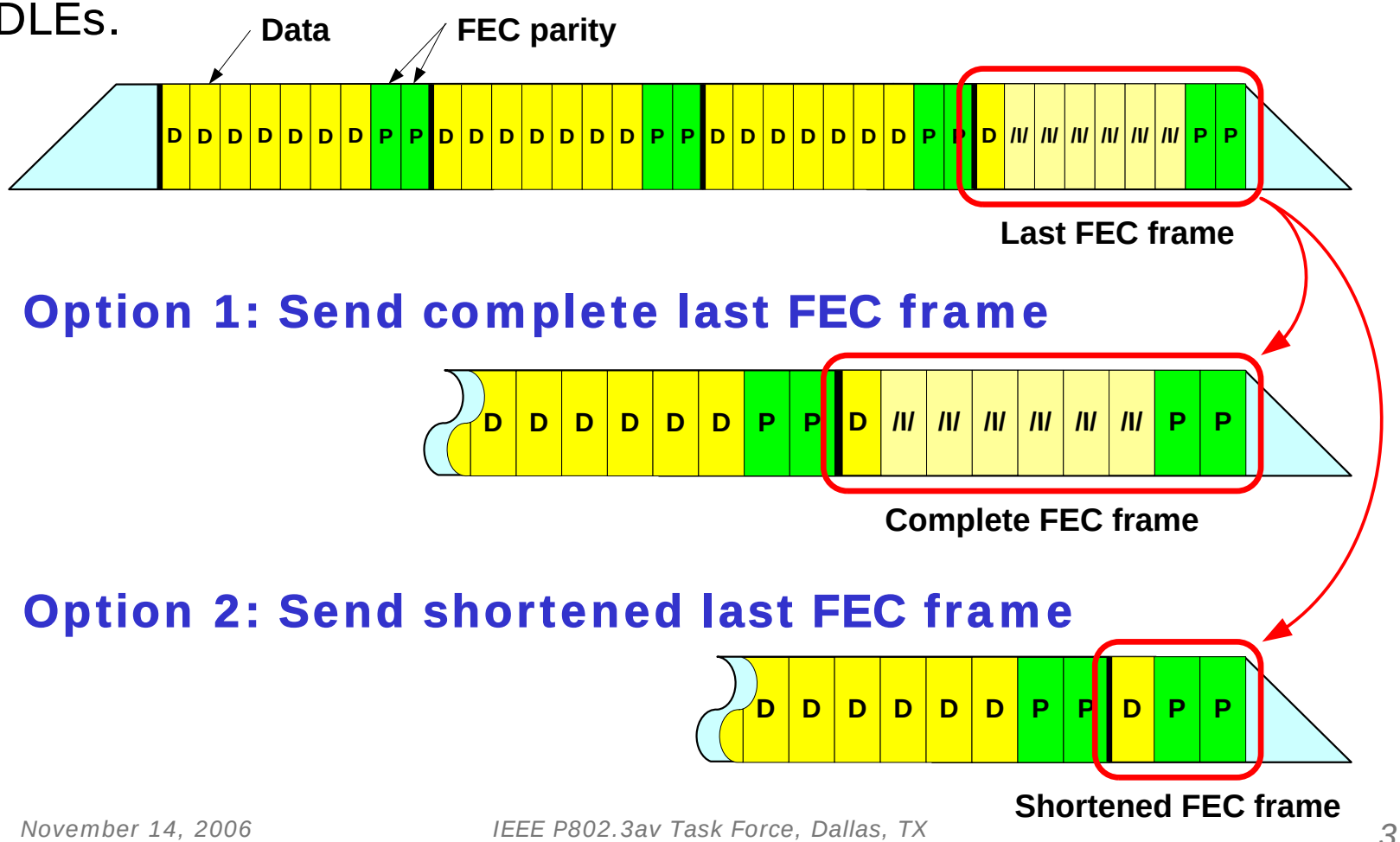


- For RS(239,255), FEC frames may look like:



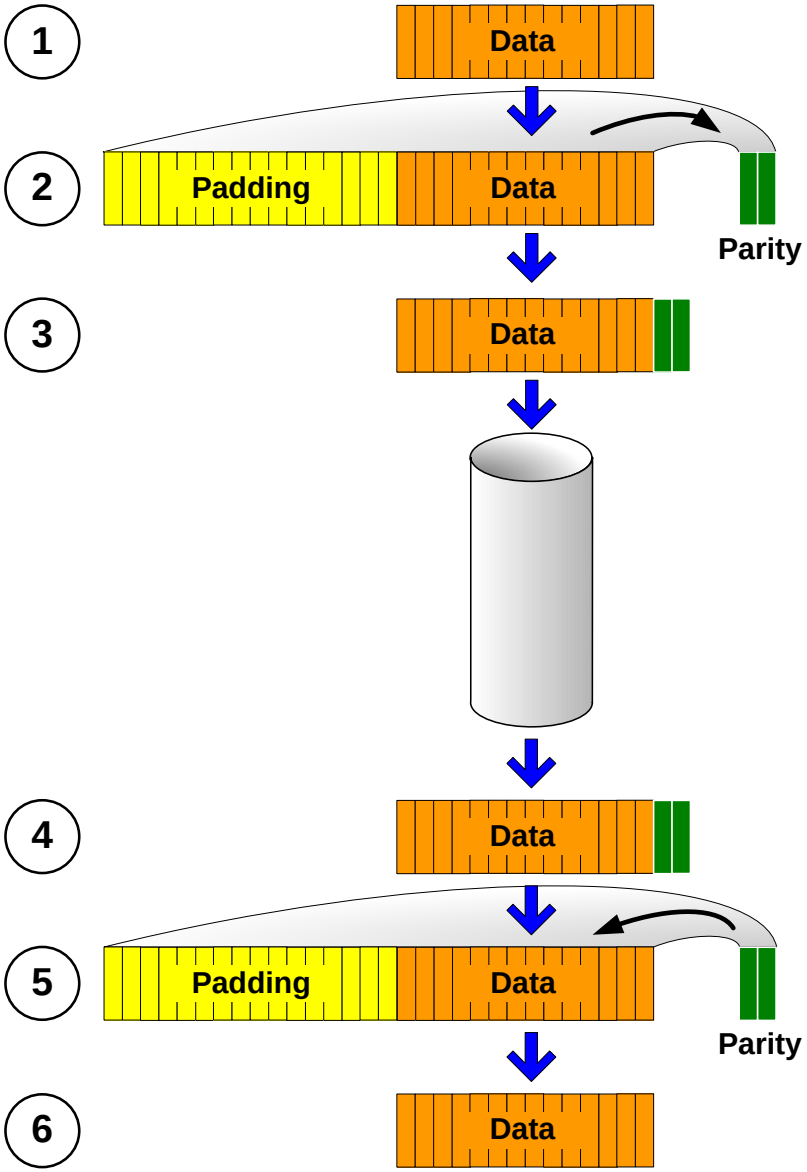
Complete vs. Shortened Last FEC Frame

- Shortened FEC frames are used only in upstream direction as the last frame in a burst.
- Last FEC frame may contain few bytes of data followed by many IDLEs.



- **Option 1: Send complete last FEC frame**
- **Option 2: Send shortened last FEC frame**

What is Shortened FEC Frame?



1. Data shorter than full payload is given to FEC encoder.
2. FEC encoder adds necessary and known padding to get full payload size. Parity is calculated over the full payload.
3. Only original data and parity is transmitted. Padding is not transmitted.
4. When the FEC decoder sees a shortened FEC block…
5. … it adds known padding, and then applies FEC algorithm to the full payload.
6. Original data is recovered.

Difficulties with Shortened Last Frame

- How receiver would know where payload ends and parity starts?
  - **Method 1: Use special FEC delimiters**

  Or
  - **Method 2: Try every position**

  Or
  - **Method 3: Add *Length* field to the previous FEC frame**
  - What else?

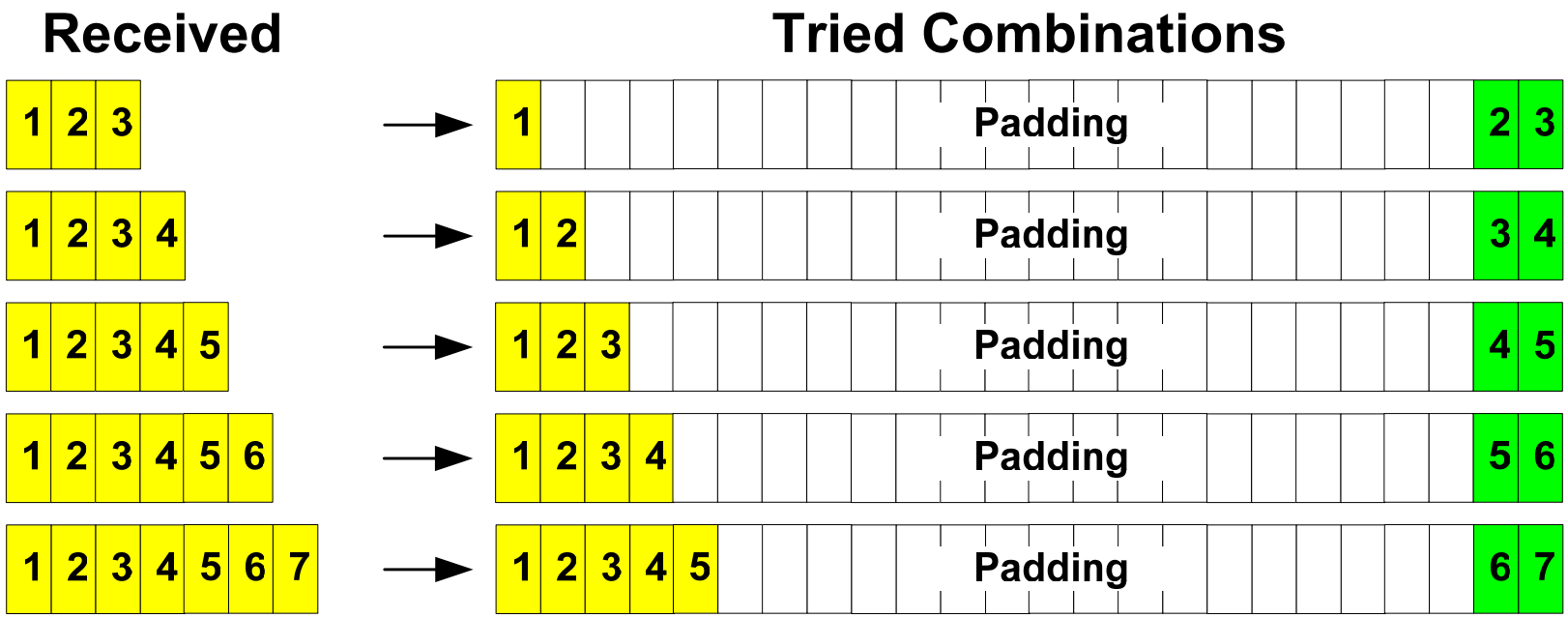
Method 1: FEC Delimiters

- FEC receiver should delineate payload from parity before it can correct the data.
- But any possible 66-bit control block value has only a Hamming distance of 2 from a valid data value.
- In presence of bit errors, FEC receiver may assume parity starts at the wrong location and will lose an entire 66-bit block, or several blocks.

Method 2: Trying Parity at Every Position

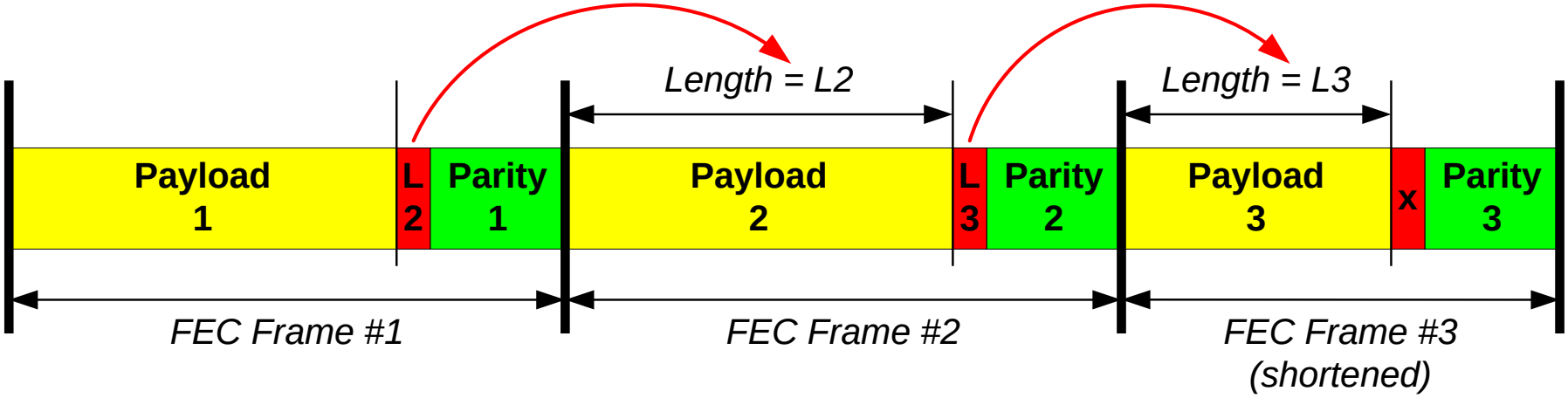
- In this scenario, as data is received, FEC receiver would try all possible parity positions in parallel

- There should be a new attempt for every 66-bit block received (every 6.4 ns)
  - **Number of parallel decoders = decoding delay / 6.4 ns**



Method 3: Length Field in FEC Frame

- A FEC frame includes *Length* field of the next FEC payload
  - Length can be a number of bytes or a number of 66-bit blocks of next payload
  - 1 byte is enough



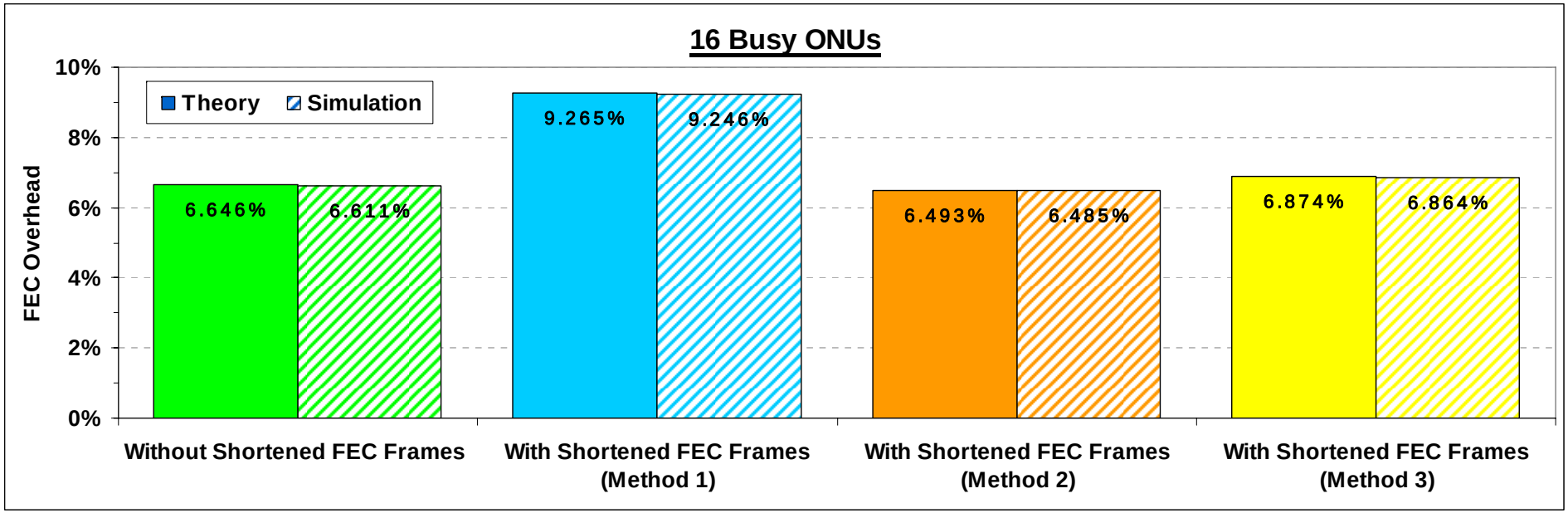
Method 3 (Continued)

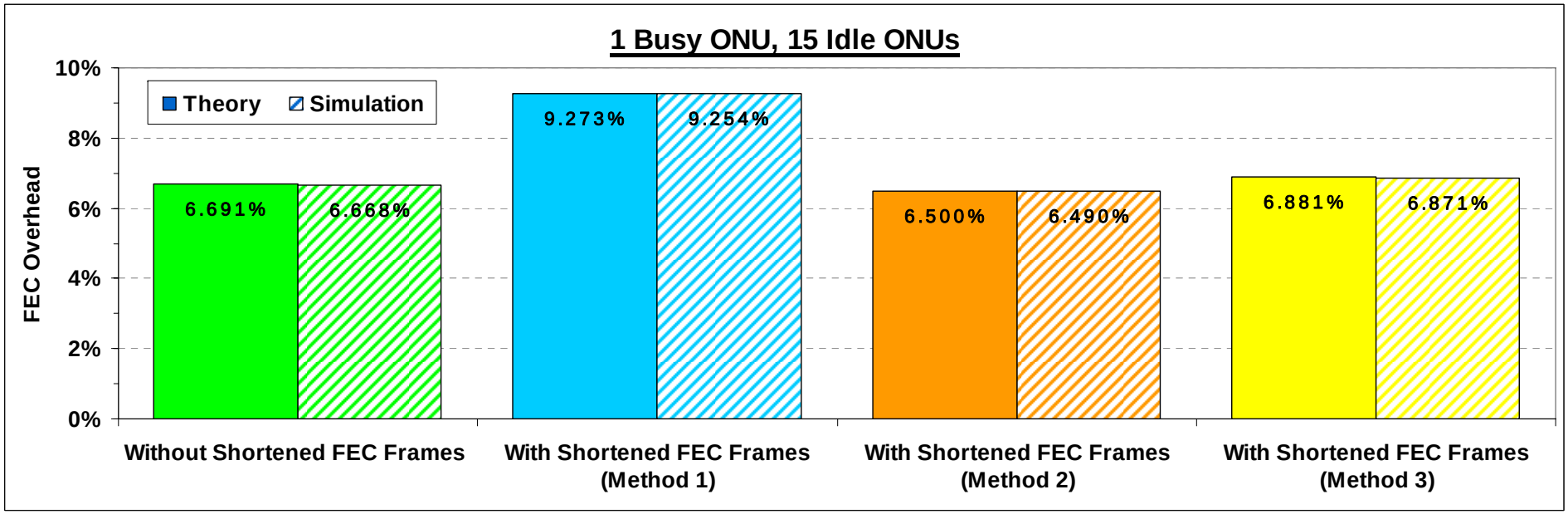
Adding a Payload Length field to a FEC frame…

1. Creates a chain effect, where loosing one 66-bit block may result in all subsequent blocks being lost
2. Increases pipeline delay at the FEC encoder
3. Contrary to initial goal, increases overhead:

	FEC OH
• Using shortened FEC frames; • 238 bytes of data + 1 byte for Length field per payload;	**6.86%**
• Not using shortened FEC frames; • 239 bytes of data per payload;	**6.61%**

FEC Overhead





Is There Any Efficiency Gain?

Even if technical problems can be solved, is there any significant throughput improvement with shortened FEC frames?

FEC Frame Format Options	Uniform Load (Every ONU is Busy)		Asymmetric Load (1 busy ONU, 15 idle ONUs)	
	FEC Overhead (%)	Throughput Delta (Mbps)	FEC Overhead (%)	Throughput Delta (Mbps)
Without Shortened FEC Frames	**6.6459%**	**baseline**	**6.6914%**	**baseline**
Method 1: • Using Shortened FEC Frames • 8-byte FEC delimiter per FEC frame	**9.2654%**	**-261.944**	**9.2726%**	**-258.120**
Method 2: • Using Shortened FEC Frames • No delimiter per FEC frame	**6.4930%**	**+15.296**	**6.5002%**	**+19.120**
Method 3: • Using Shortened FEC Frames • 1-byte Length field within the payload of FEC frame	**6.8739%**	**-22.800**	**6.8806%**	**-18.928**

Summary

	Complexity	Efficiency
Without shortened FEC frames	Easy to delineate payload from parity simply by their position within FEC frame: N payload blocks are always followed by M parity blocks.	An under-filled FEC frame on average will waste 96 ns (6 TQ) per burst. Overall, FEC overhead is within 6.6 – 6.7%.
Method 1: • Using shortened FEC frames • 8-byte FEC delimiter per FEC frame	It is not clear how to delineate variable payload from parity, since any 66-bit value may only have a Hamming distance of 2 from a valid data block.	Contrary to the initial goal, the usable bandwidth actually decreases by ~260 Mbps.
Method 2: • Using shortened FEC frames • No delimiter per FEC frame	Requires very large number of parallel decoders. Not clear how to distinguish false positives.	Using shortened FEC frames provides only a negligible gain of 20 Mbps
Method 3: • Using shortened FEC frames • 1-byte Length field within the payload of FEC frame	Increases pipeline delay at the encoder (cannot send a previous FEC frame until encodes the next frame). Creates error chain effect.	Contrary to the initial goal, the usable bandwidth actually decreases by ~20 Mbps.

Conclusion

- Using shortened FEC frames significantly increases complexity
- Using shortened FEC frames does not significantly improve bandwidth efficiency (or even make it worse).
- No reason to spend effort on shortened FEC frames