



MICROCHIP

10BASE-T1S Scrambler Analysis

**Jiachi Yu, Dixon Chen, Hongming An,
John Zang, Kevin Yang**

**IEEE 802.3cg Rosemont Meeting
March 6, 2018**

Outline

- Compare different schemes including the complementary Golay sequence preamble and payload scrambler as proposed by Tazebay, Cordaro, et al. (*referred as Tazebay's proposal in following slides*)
- Synchronous scrambler vs self-synchronizing scrambler
- Propose a new scheme which scrambles both the payload as well as part of the standard preamble
- Simulation and laboratory measurement results
- Conclusion

Introduction

- Tazebay, Cordaro, et al., proposed payload scrambling and complementary Golay sequence preamble replacement ([cordaro_8023cg_short_reach_new_preamble_proposal_1220.pdf](#), [cordaro_8023cg_01_0118_v2.pdf](#), [tazebay_3cg_01b_0118.pdf](#))
 - Scramble payload – reduce the peak emissions for some payload
 - New preamble – better synchronization performance, further improvement in the PSD

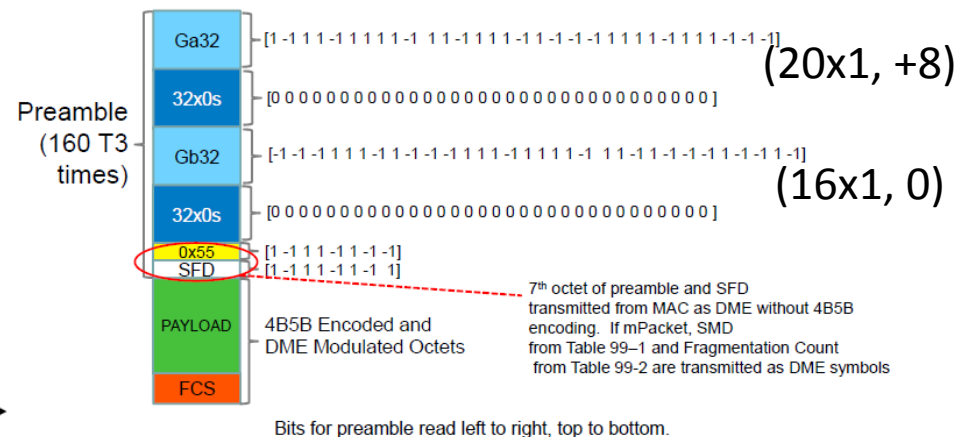
How to scramble 10BASE-T1S Frames

- 10BASE-T1S does not transmit IDLEs on the line when no data present.
 - No continuously running scrambler.
- How to synchronize the scrambler?
 1. Transmit preamble unscrambled in order to synchronize.
 - Emissions performance of raw preamble is important.
 2. Scramble the 4B5B-encoded payload at the transmitter with $x^{15}+x^4+1$ scrambler with same initial state loaded at beginning of every frame.

$scr_initial_state = ([0011111100110101])$
 3. At receiver, detect preamble and then start descrambler with same fixed initial state at beginning of packet data.



Proposed Preamble and Payload Format



Introduction

802.3cg D1.1 147.3.3 Preamble and payload format



Preamble and payload format proposed by Cordaro, Tazebay, et al.



Some observations on Cordaro and Tazebay's proposed changes:

- The scrambler does effectively reduce the worse payload peak emissions
- However, the proposed complementary Golay preamble results in:
 - Preamble is not encoded and not DC balanced – AC coupling drift, PoDL issue
 - Breaks DME encoding and its self-clocking property – the most important feature for a multi-drop system for fast data and clock recovery (a few bits); but still doubles the channel bandwidth (no advantages over DME but inherits the disadvantages)
 - Much longer synchronization (lock) time for the receiver; requires an individual preamble generator and detector increasing design complexity;
 - Introduces a 3 level signaling scheme instead of a 2 level binary of DME requiring an ADC for preamble detection; a dramatic increase in the receiver complexity;
 - PLCA may require significant modification, e.g. BEACON needs to be synchronized the same way as preamble (add preamble for sync), significantly reducing network efficiency

Possible Alternate Solution

- After analysis of different payload patterns, we propose to scramble the 6 preamble octets following the JJK and the payload of the frame:

Scramble Preamble (six octets) and Payload



- This has the advantage of:
 - Maintains DME and clocking recovery
 - Does not introduce three-level encoding and unbalanced DC
 - Does not change 10BASE-T1S frame format
 - PLCA scheme remains the same

Can the same level of PSD peak emission reduction be obtained??

Analysis

Standard Preamble, Unscrambled Payload



Standard Preamble, Scrambled Payload



Both Standard Preamble (six octets) and Payload Scrambled (new proposal)

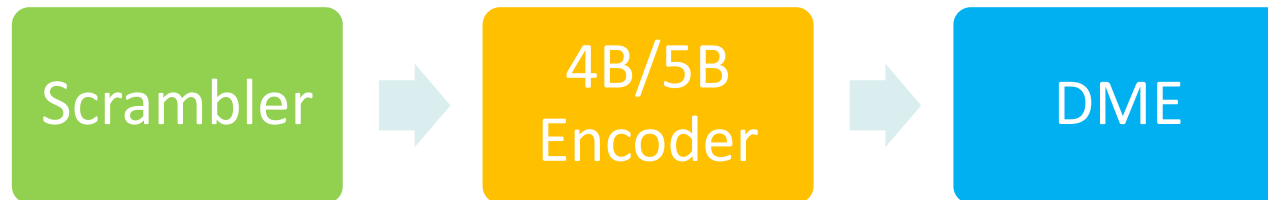


Cordaro & Tazebay's Complementary Golay Sequence Preamble, Scrambled Payload



Scrambler Positioning

Mode 1: Scrambler inserted before 4B/5B Encoder



Mode 2: Scrambler inserted after 4B/5B Encoder



- Simulations show that better performance is achieved by inserting the Scrambler after the 4B/5B Encoder (Mode 2) than when inserting the Scrambler before the 4B/5B Encoder (Mode 1).
- The following plots only show the cases for Mode 2



Simulation Conditions

Scrambler: $x^{15} + x^4 + 1$

Scrambler initial (Tazebay): [0 0 1 1 1 1 1 0 0 1 1 0 1 0 1]
(new set) [0 0 1 0 1 0 0 1 1 0 0 0 0 0 1]

Payload: 5 different payloads captured by Wireshark

5 different Payload lengths: 60, 160, 170, 342, and 1560 bytes

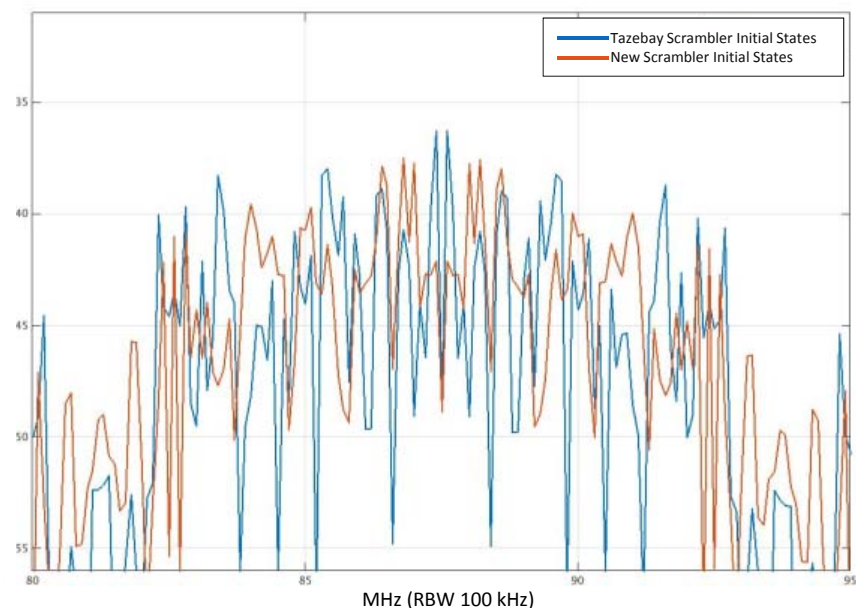
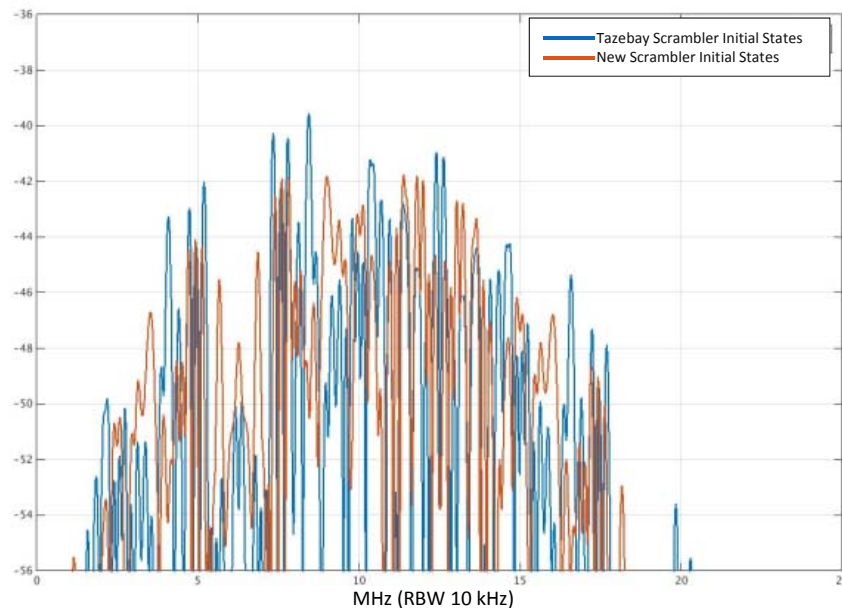
Spectrum RBW: 10 kHz, 100 kHz



Determine Scrambler Initial State

Scrambler initial state (Tazebay): [0 0 1 1 1 1 1 0 0 1 1 0 1 0 1]

New initial states were searched by PSD flatness in the 6 preamble octets 55 55 55 55 55 SFD



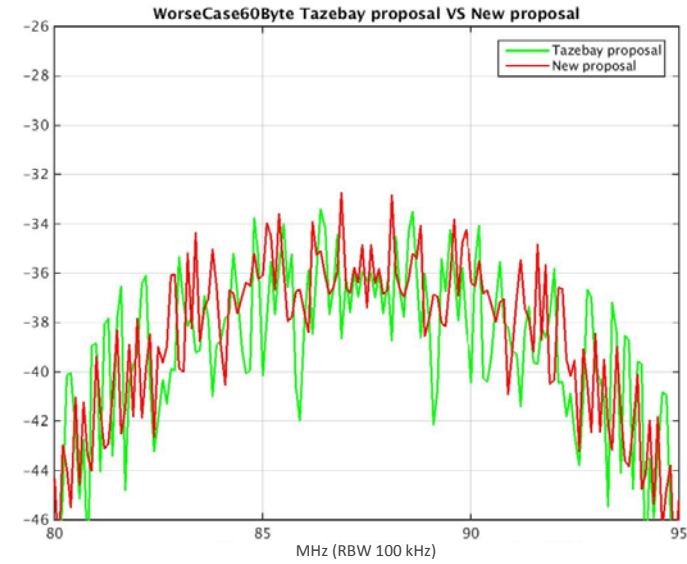
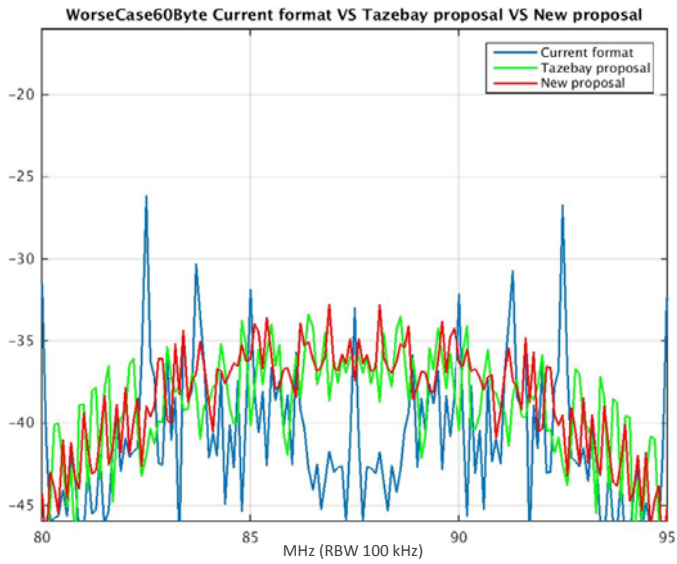
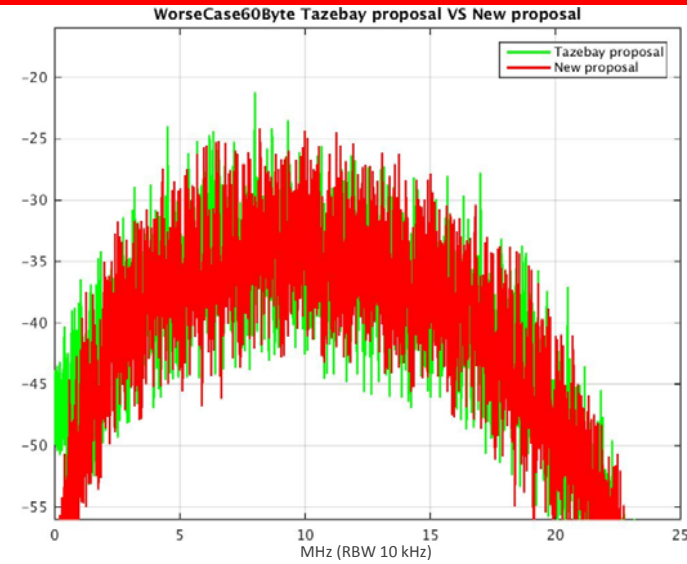
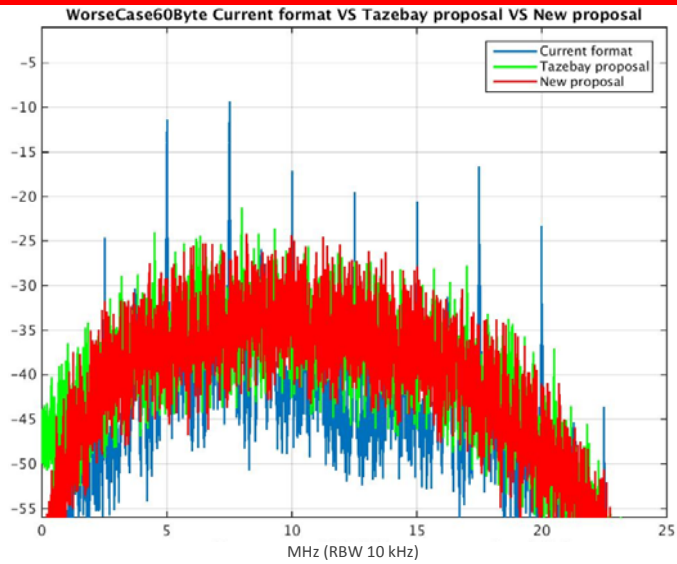
New Scrambler initial state (this work): [0 0 1 0 1 0 0 1 1 0 0 0 0 0 1]

Gives 2.2 dB and 1.2 dB better results for 10 kHz and 100 kHz RBW, respectively



60 Byte – Worse Case

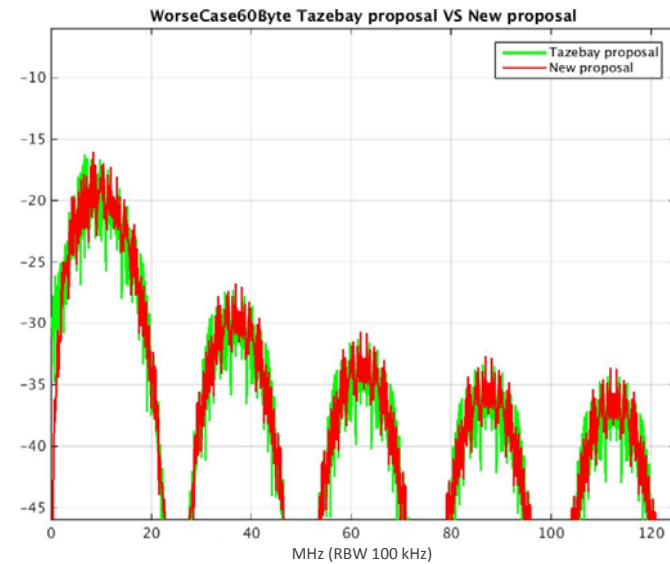
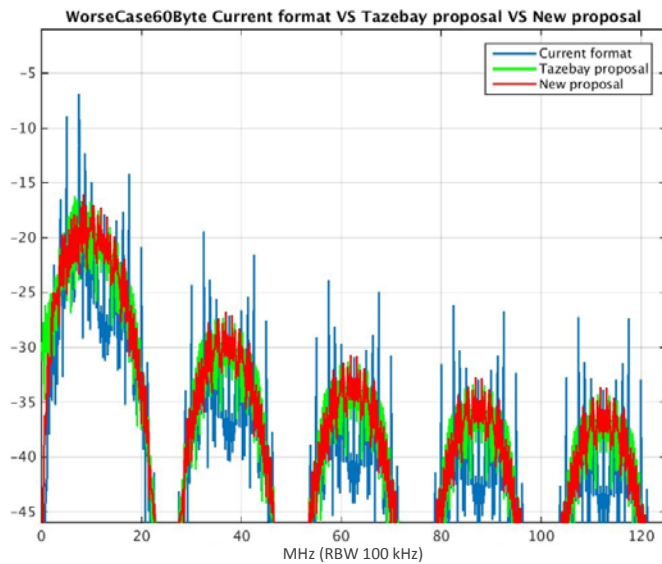
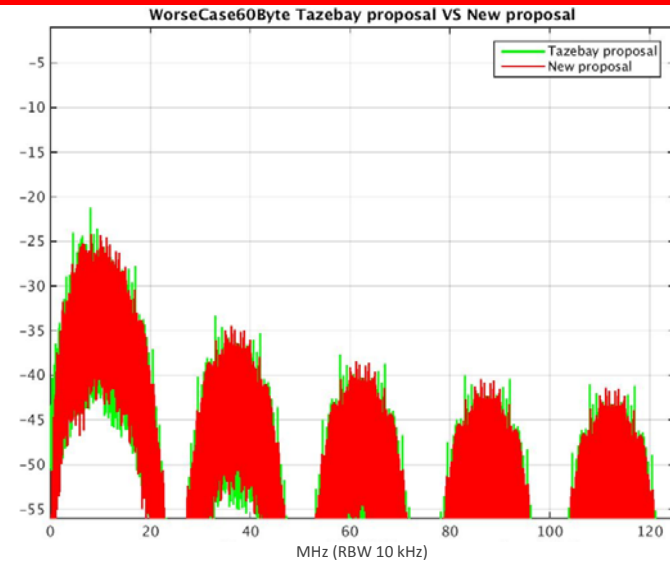
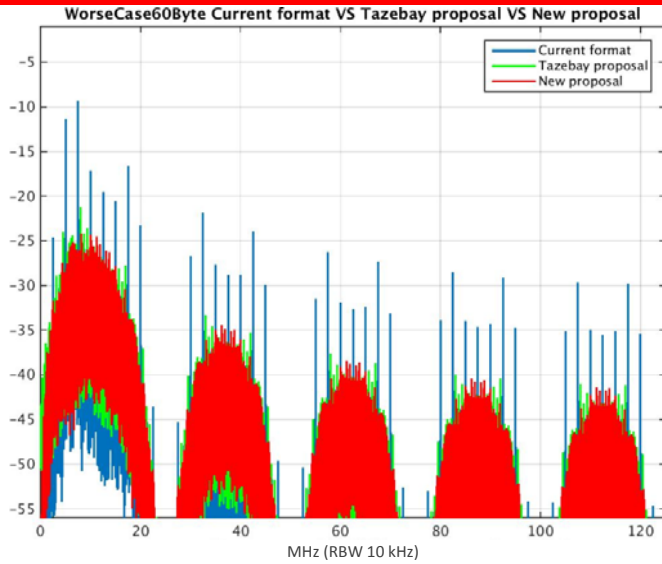
(3 dB better at low band, 0.6 dB worse at high band)
(Tezebay's vs this new proposal, same for next slides)





60 Byte – Worse Case

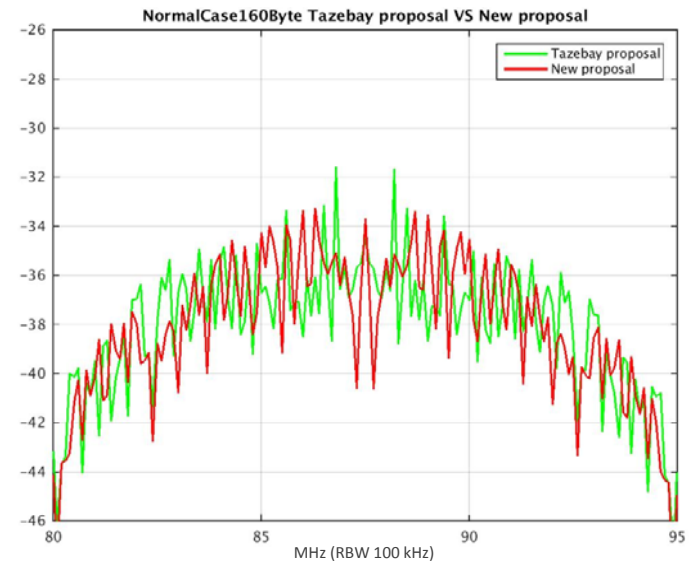
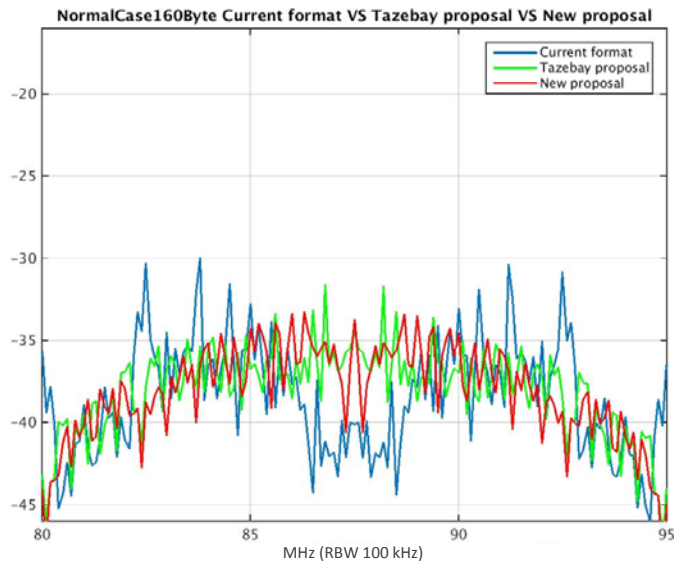
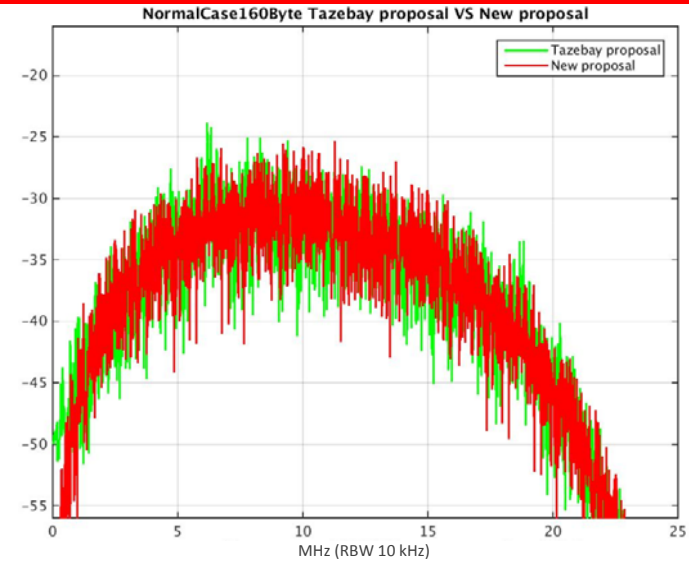
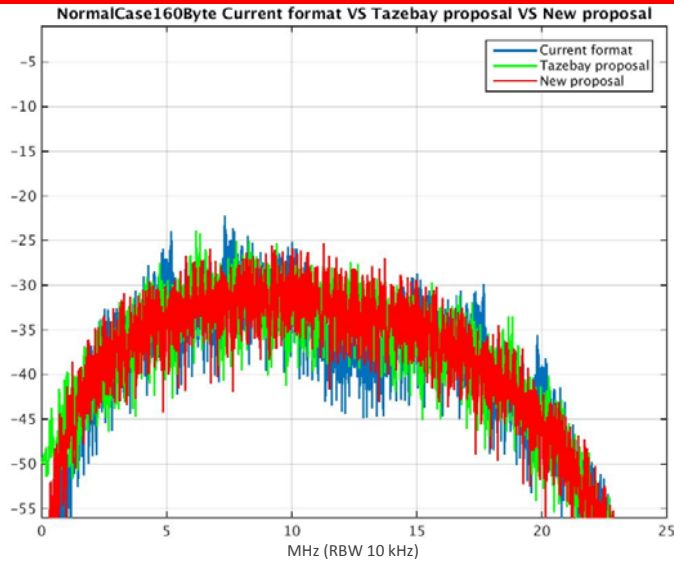
(3 dB better at low band, 0.6 dB worse at high band)





160 Byte – Normal Case

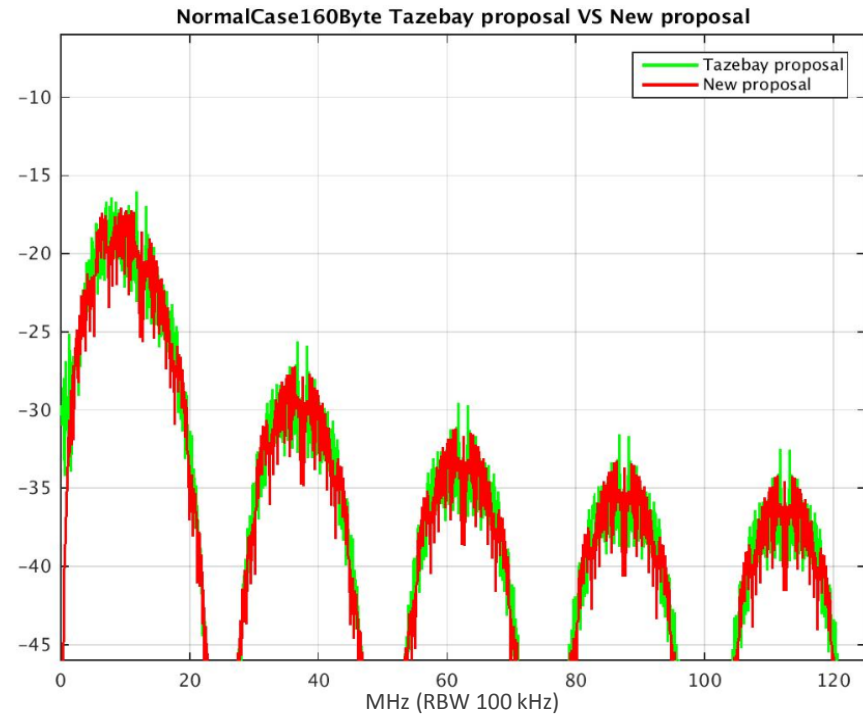
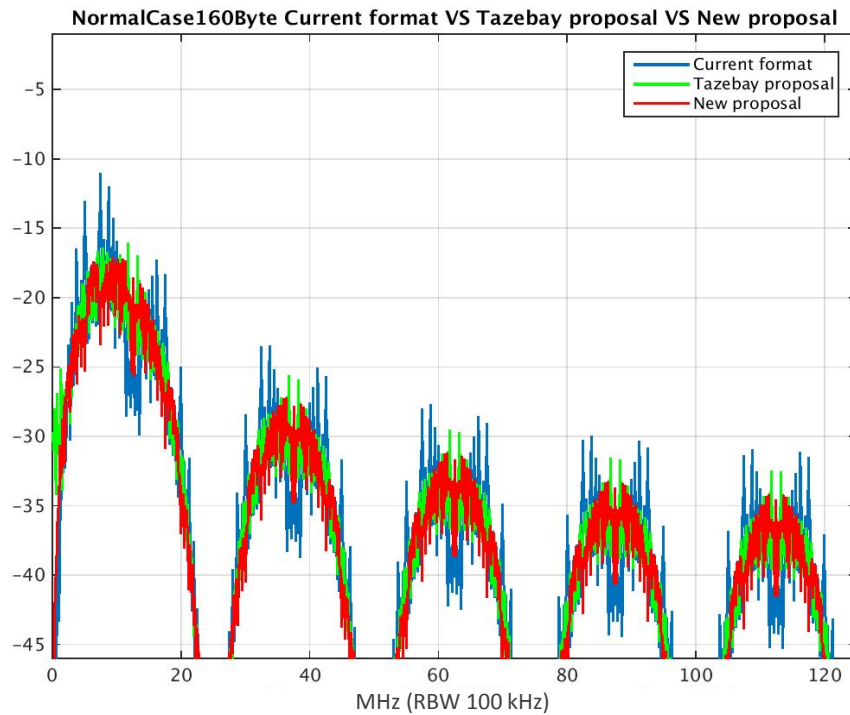
(1.5 dB better at low band, 1.7 dB better at high band)





160 Byte – Normal Case

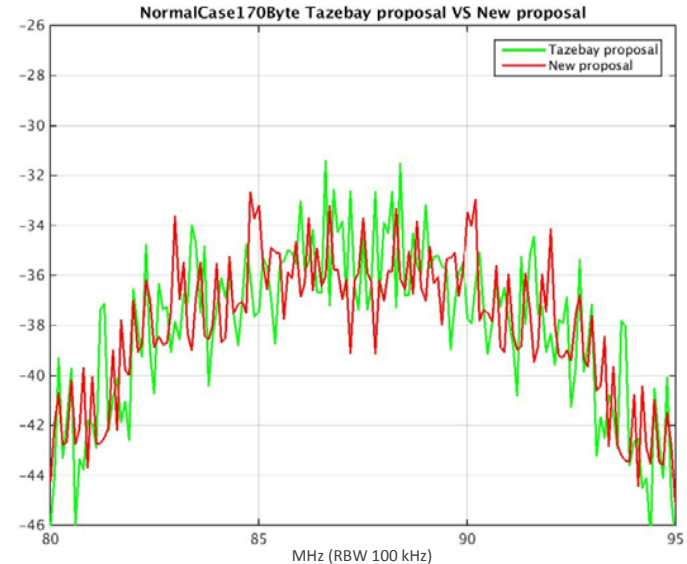
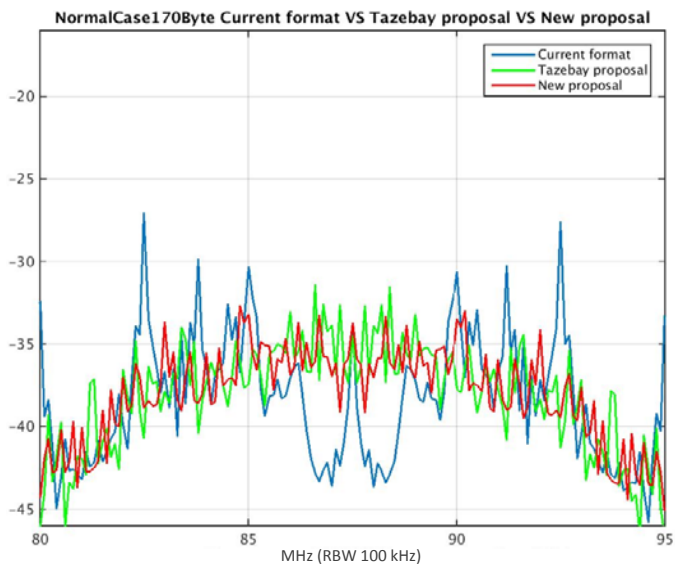
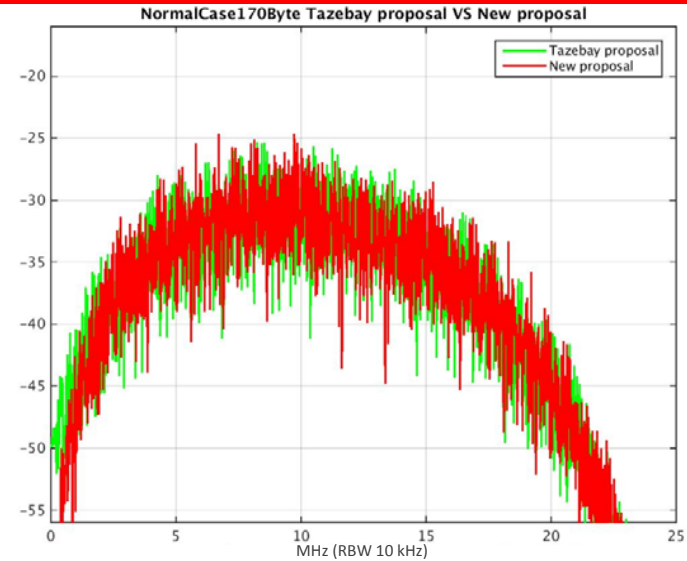
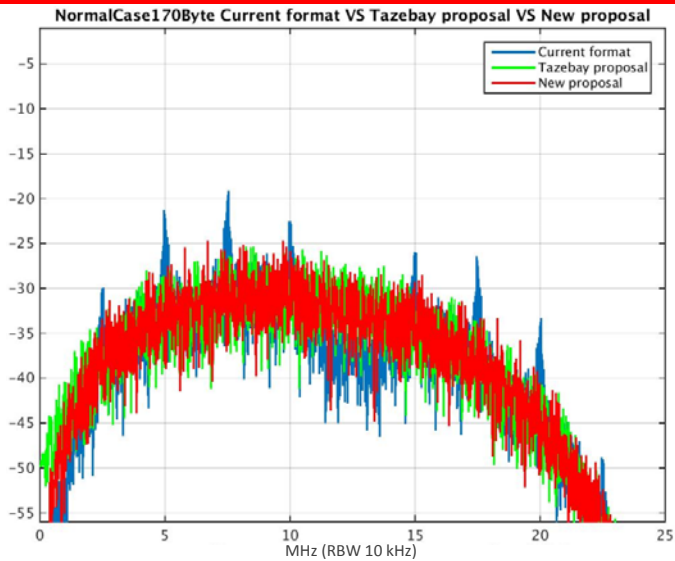
(1.5 dB better at low band, 1.7 dB better at high band)





170 Byte – Normal Case

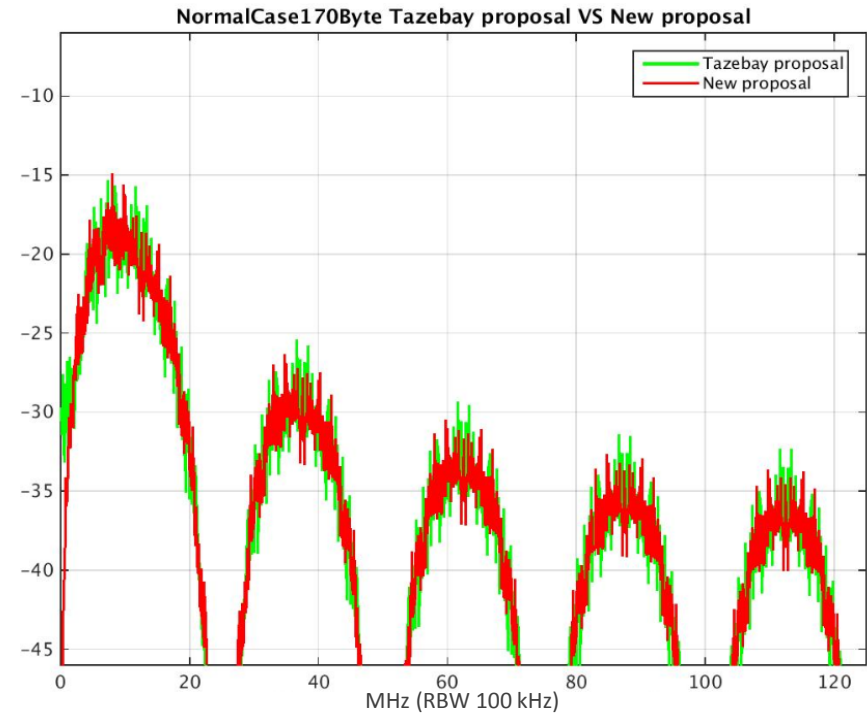
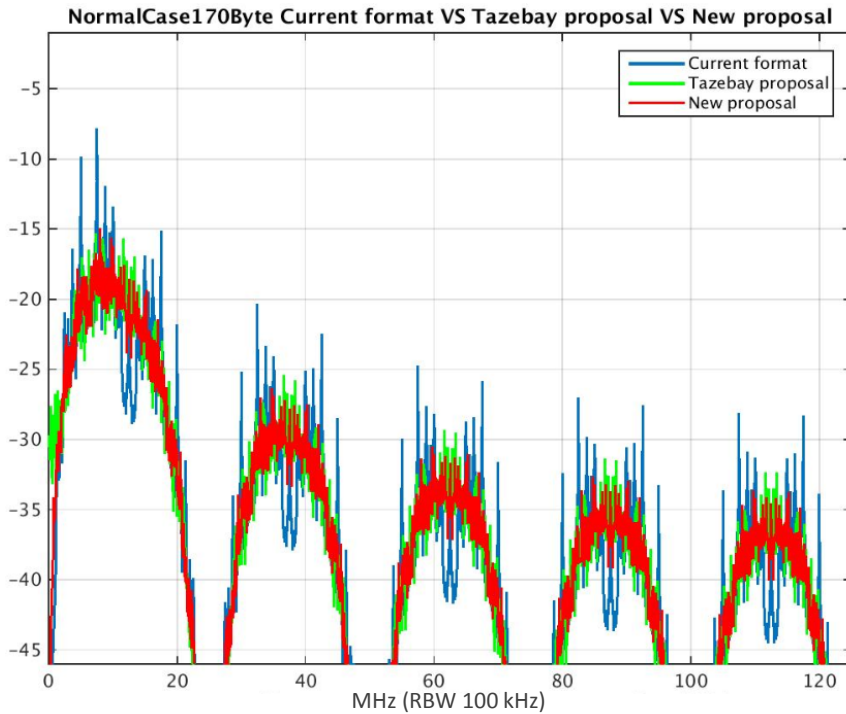
(0.7 dB worse at low band, 1.2 dB better at high band)





170 Byte – Normal Case

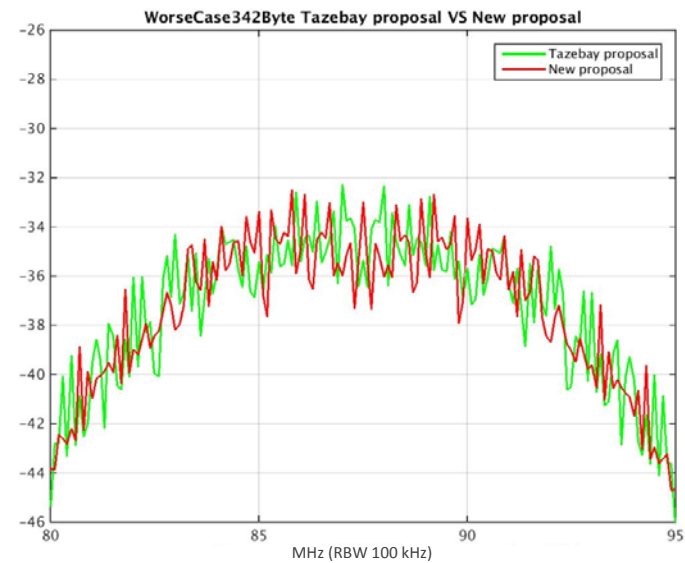
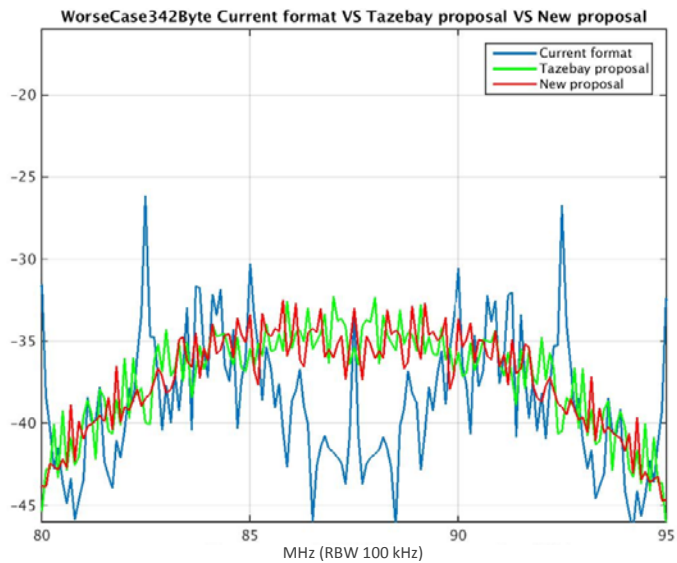
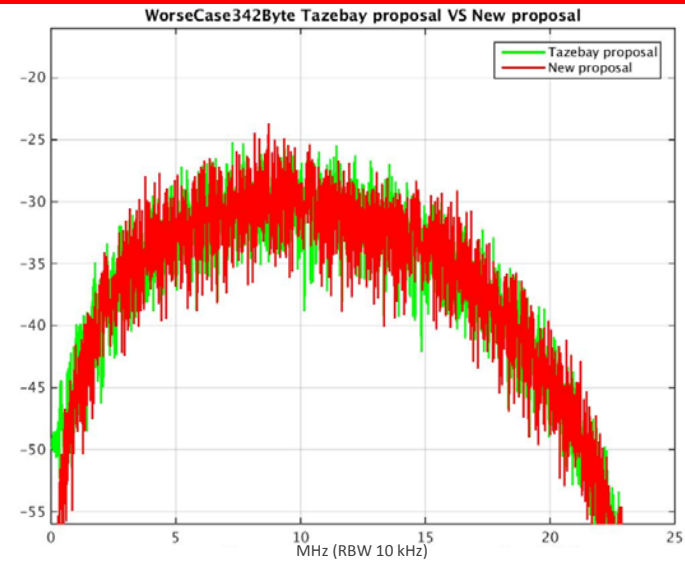
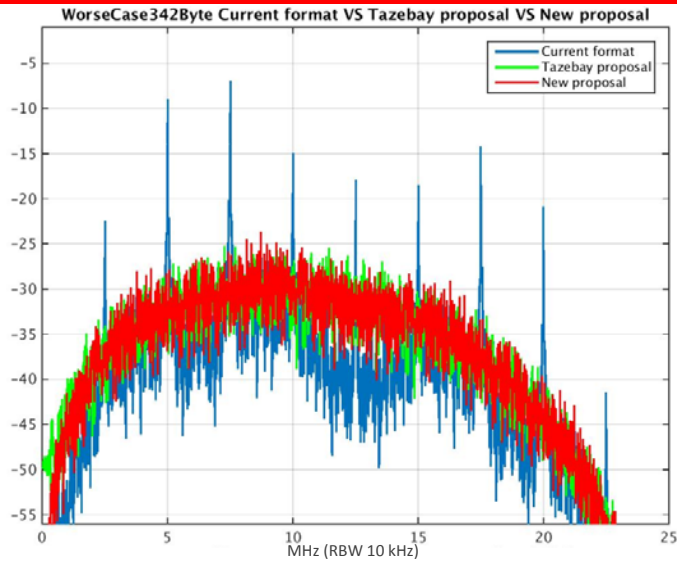
(0.7 dB worse at low band, 1.2 dB better at high band)





342 Byte – Worst Case

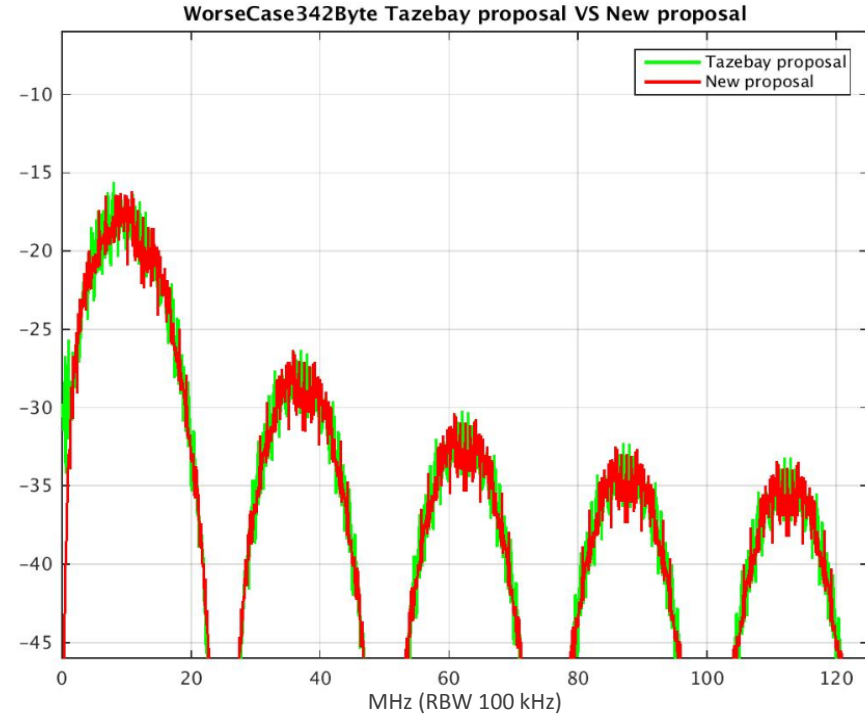
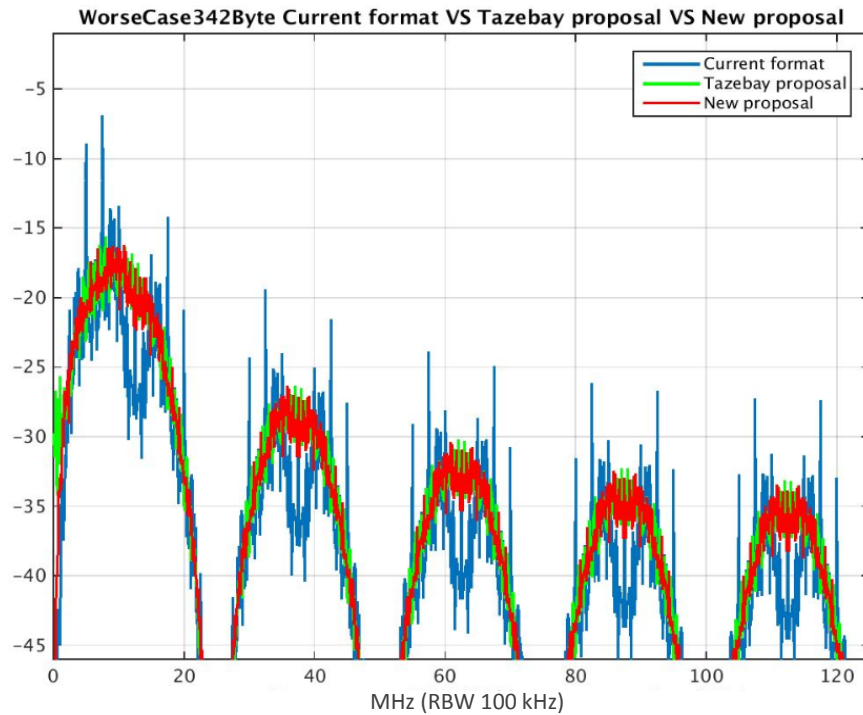
(1.5 dB worse at low band, 0.2 dB better at high band)





342 Byte – Worst Case

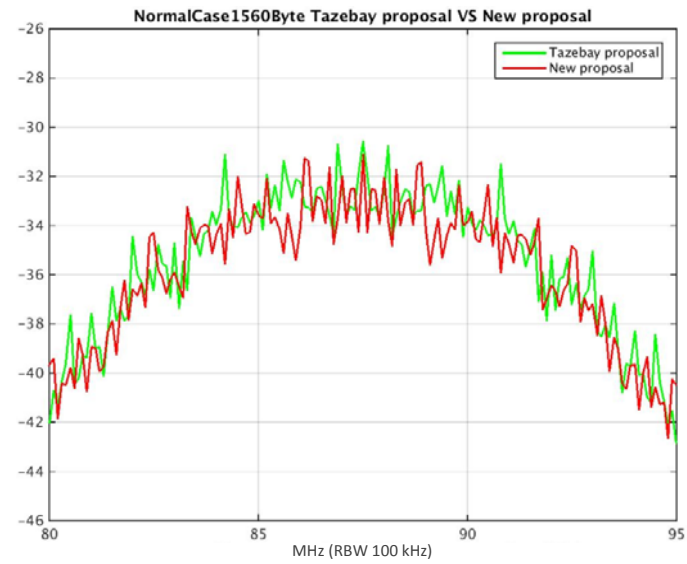
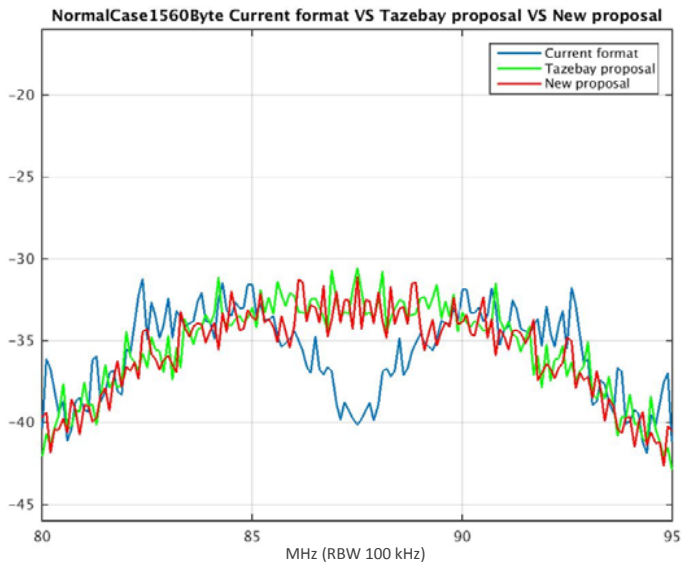
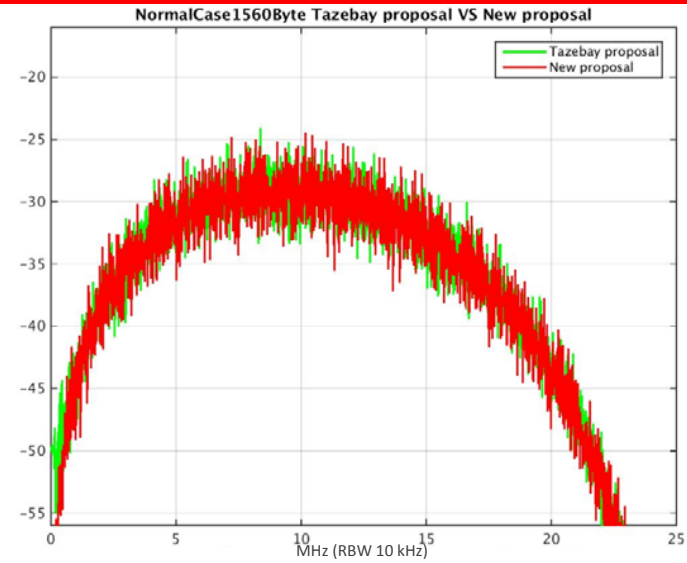
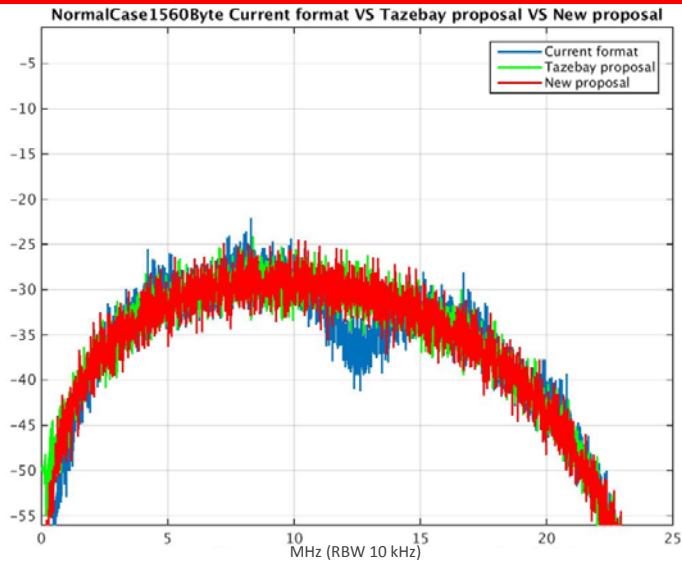
(1.5 dB worse at low band, 0.2 dB better at high band)





1560 Byte – Normal Case

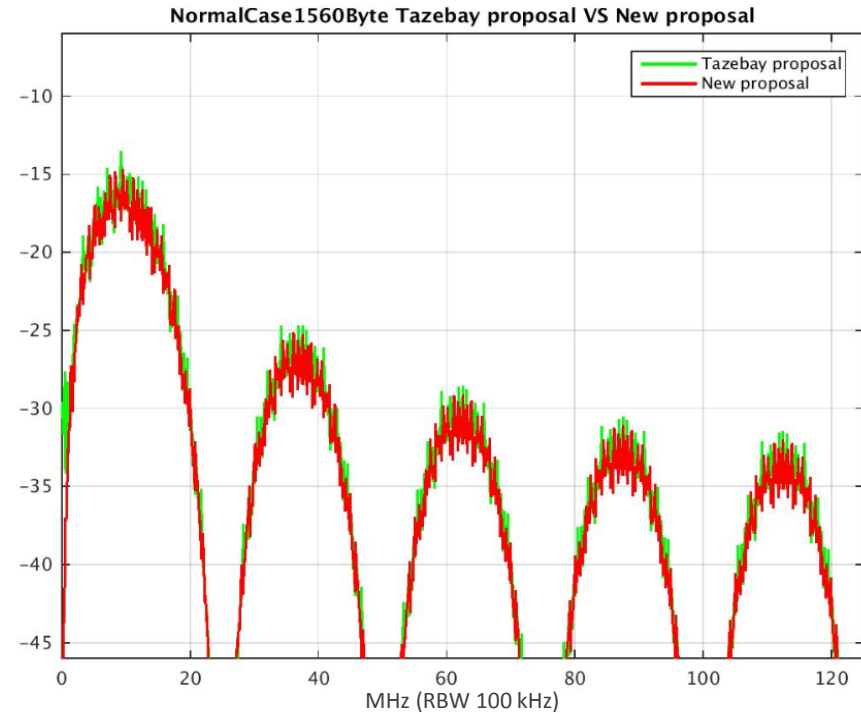
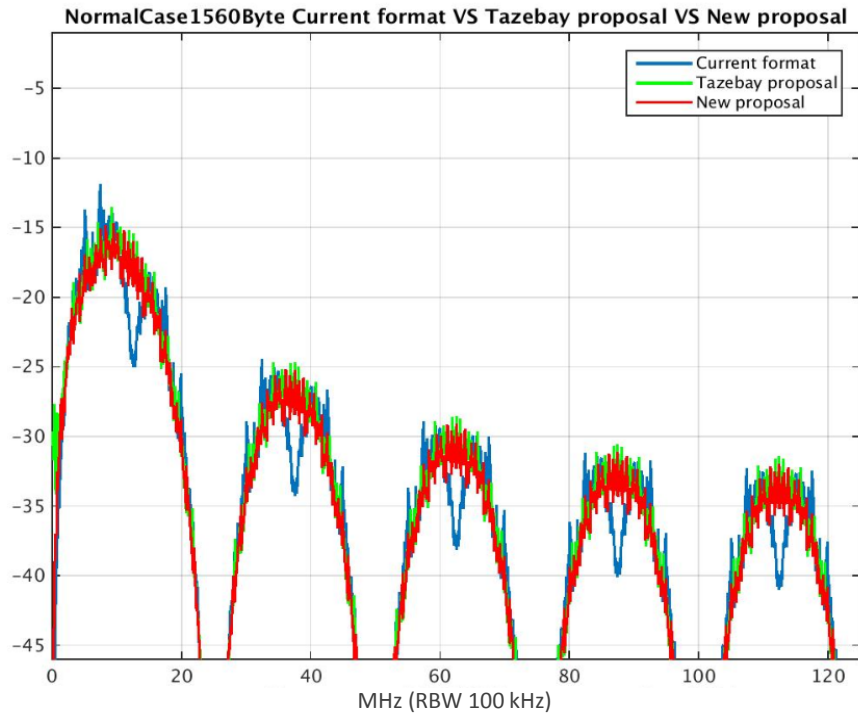
(0.4dB better at low band, 0.5 dB better at high band)





1560 Byte – Normal Case

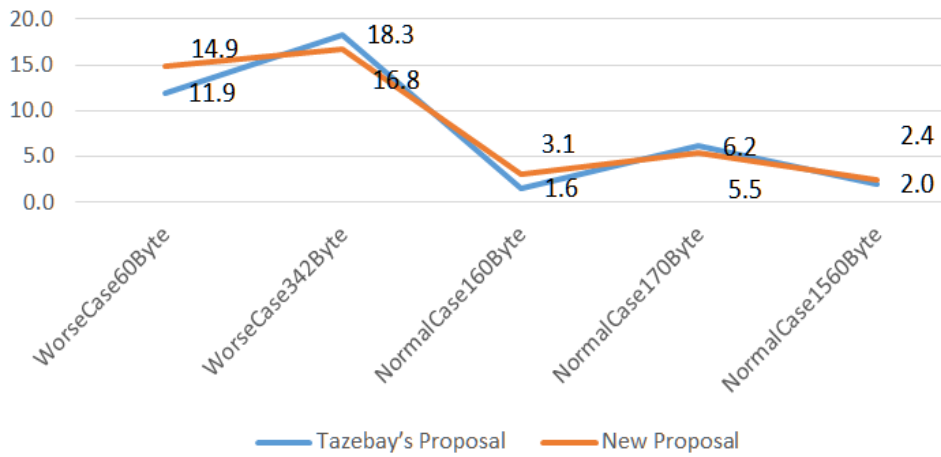
(0.4dB better at low band, 0.5 dB better at high band)



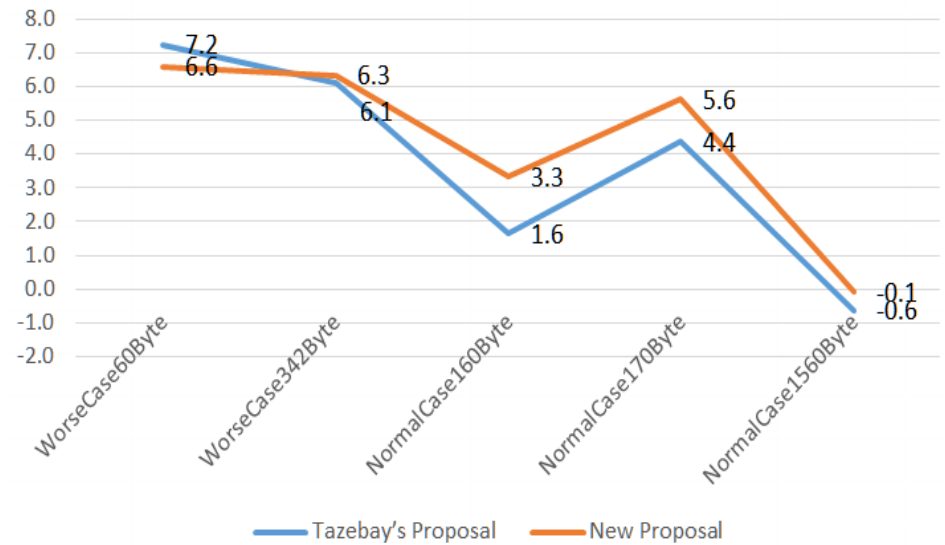


Improvement Summary

RBW10Khz DC-25Mhz improvement (dB)



RBW100Khz 80-95Mhz improvement (dB)



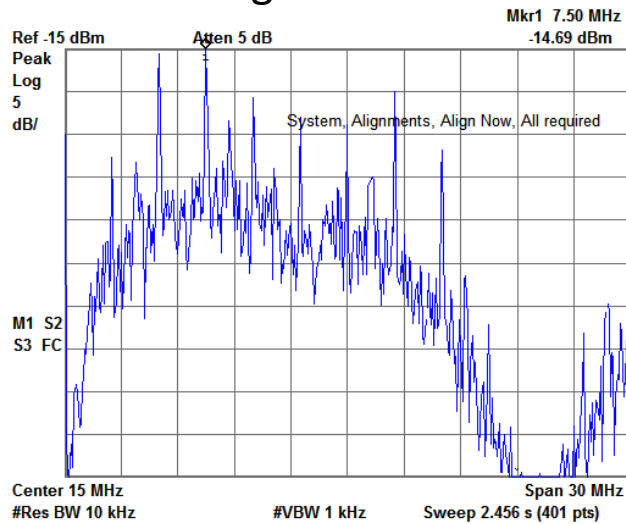
Lab Measurements

- The lab data is measured using:
 - Tektronix AWG4162 arbitrary waveform generator
 - Agilent E4404B spectrum analyzer

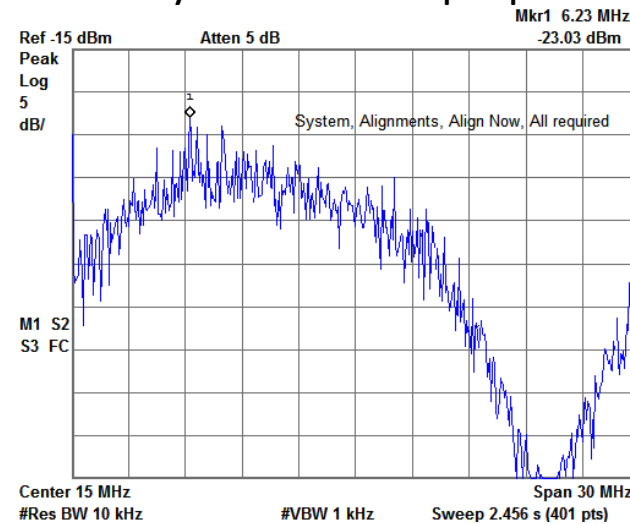


60 Byte – Worst Case Lab Results (0-30 MHz Span)

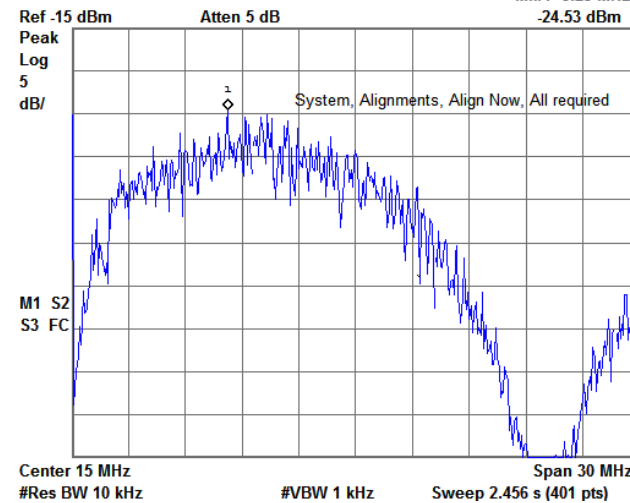
Original data



Tazebay & Cordaro's proposal



- 15 dB peak on original data
- 8.3 dB reduction with Tazebay & Cordaro's proposal
- 9.8 dB with this proposal
- 1.5 dB vs 3 dB from analysis

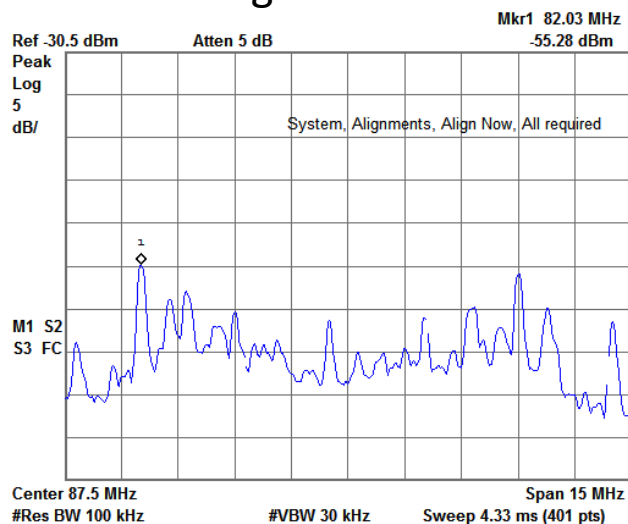


This proposal

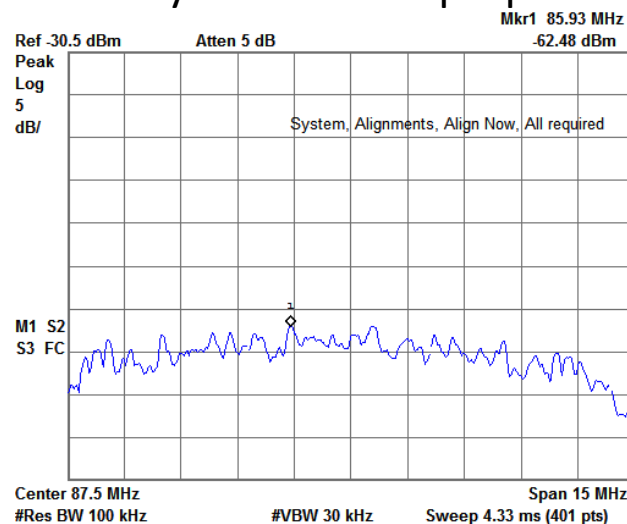


60 Byte – Worst Case Lab Results (80-95 MHz Span)

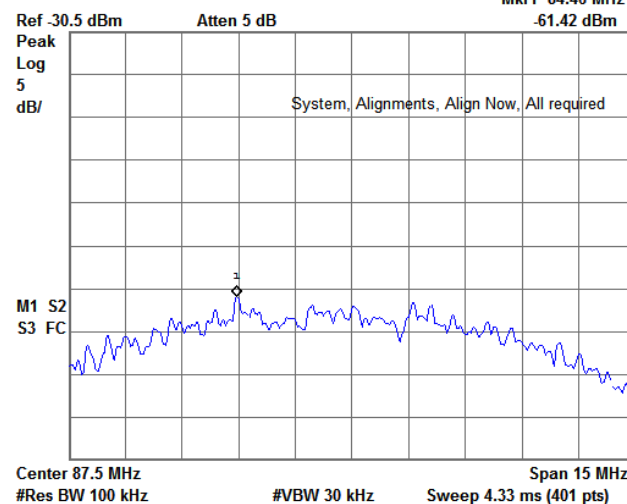
Original data



Tazebay & Cordaro's proposal



- 7.2 dB reduction with Tazebay & Cordaro's proposal
- 6.1 dB with this proposal
- 1.1 dB vs 0.6 dB from analysis

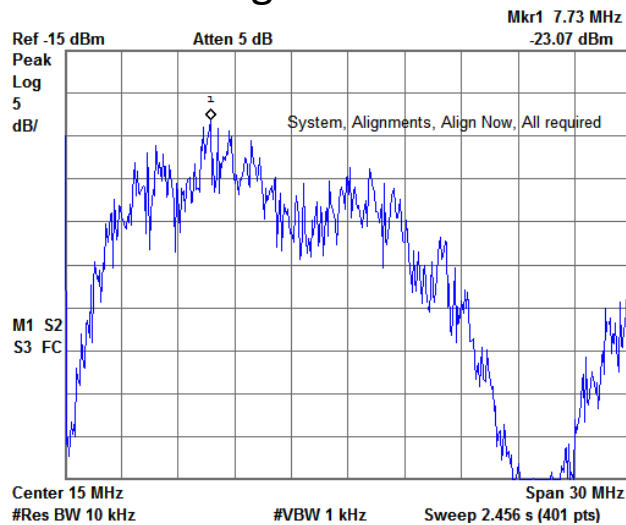


This proposal

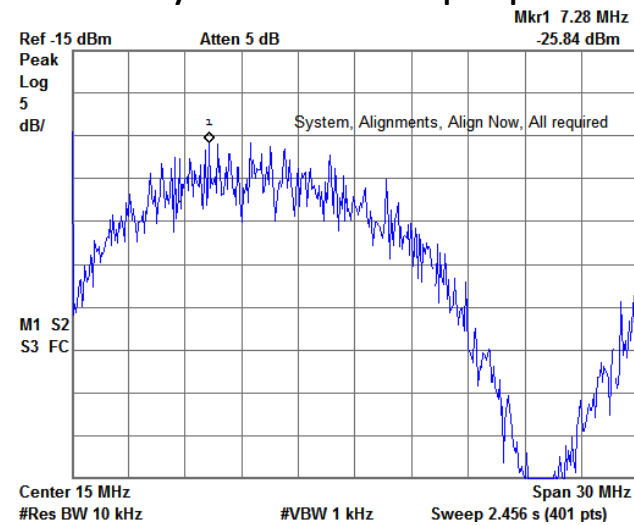


1560 Byte – Normal case Lab Results (0-30 MHz Span)

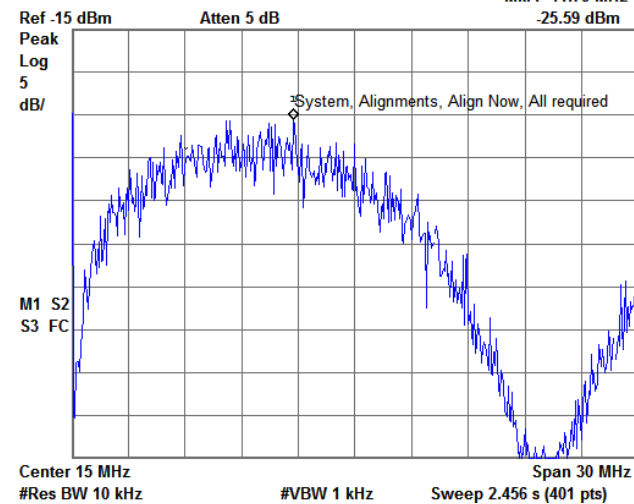
Original data



Tazebay & Cordaro's proposal



- 2.8 dB reduction with Tazebay & Cordaro's proposal
- 2.5 dB with this proposal
- 0.3 dB worse vs 0.5 dB better from analysis

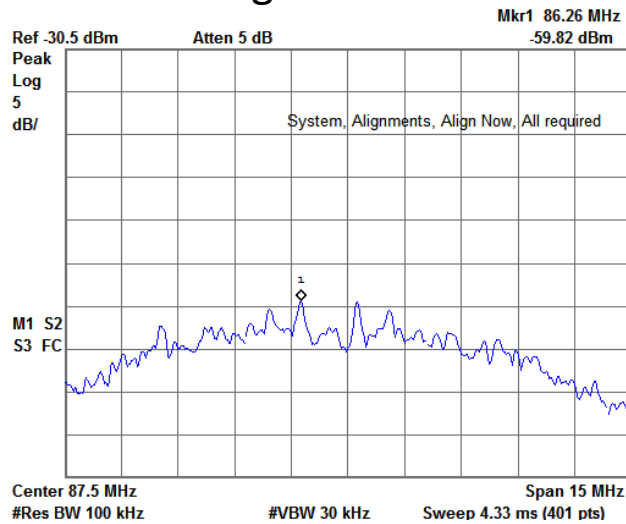


This proposal

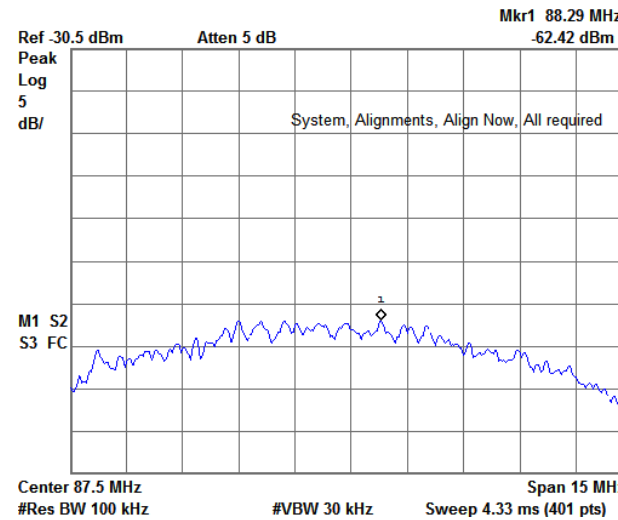


1560 Byte – Normal case Lab Results (80-95 MHz Span)

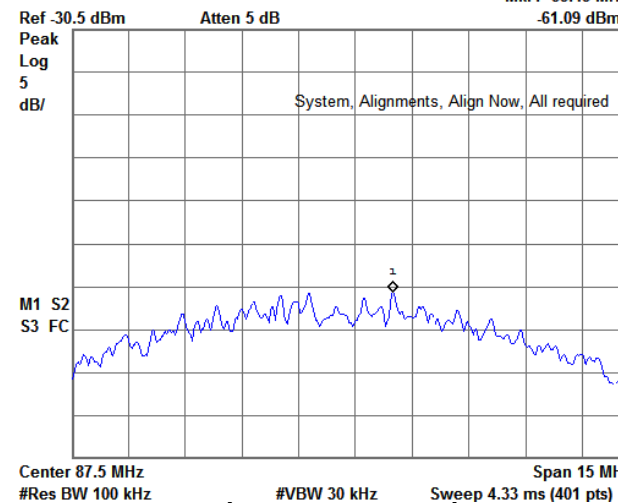
Original data



Tazebay & Cordaro's proposal



- 2.6 dB reduction with Tazebay & Cordaro's proposal
- 2.1 dB with this proposal
- 0.5 dB worse vs 0.5 dB better from analysis



This proposal

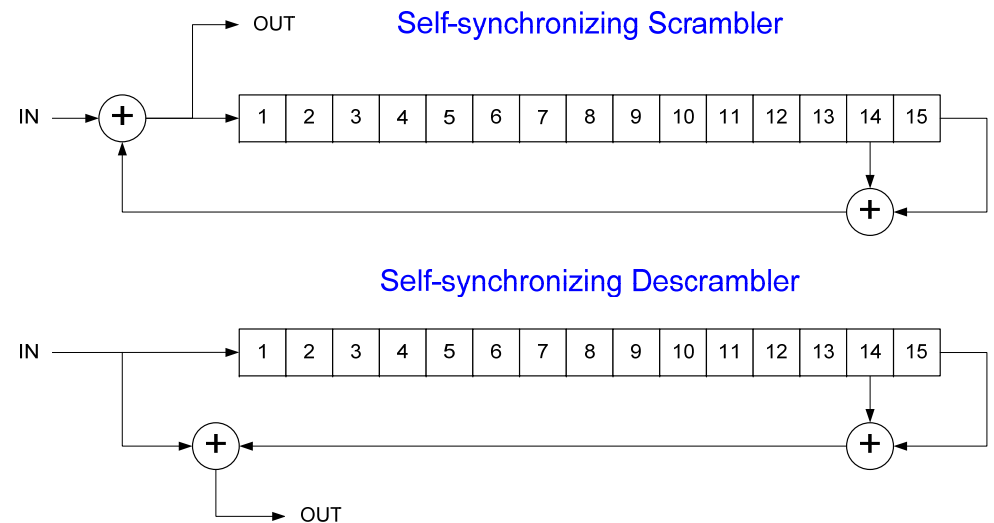
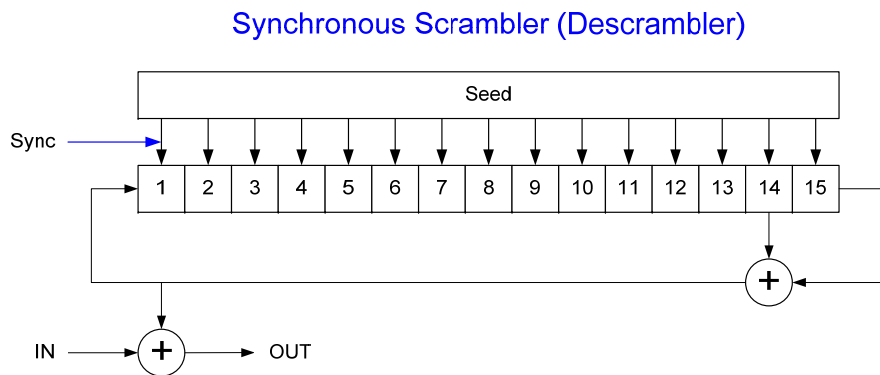
Results Summary

Simulation and lab measurements demonstrate that

- the proposed scrambling of the preamble and payload will achieve the same level of PSD peak emission reduction the preamble and scrambler proposed by Cordaro, Tazebay, et al.,
- keeps the PLCA untouched and maintaining the DME encoding self-clocking property permitting a lower cost system implementation.

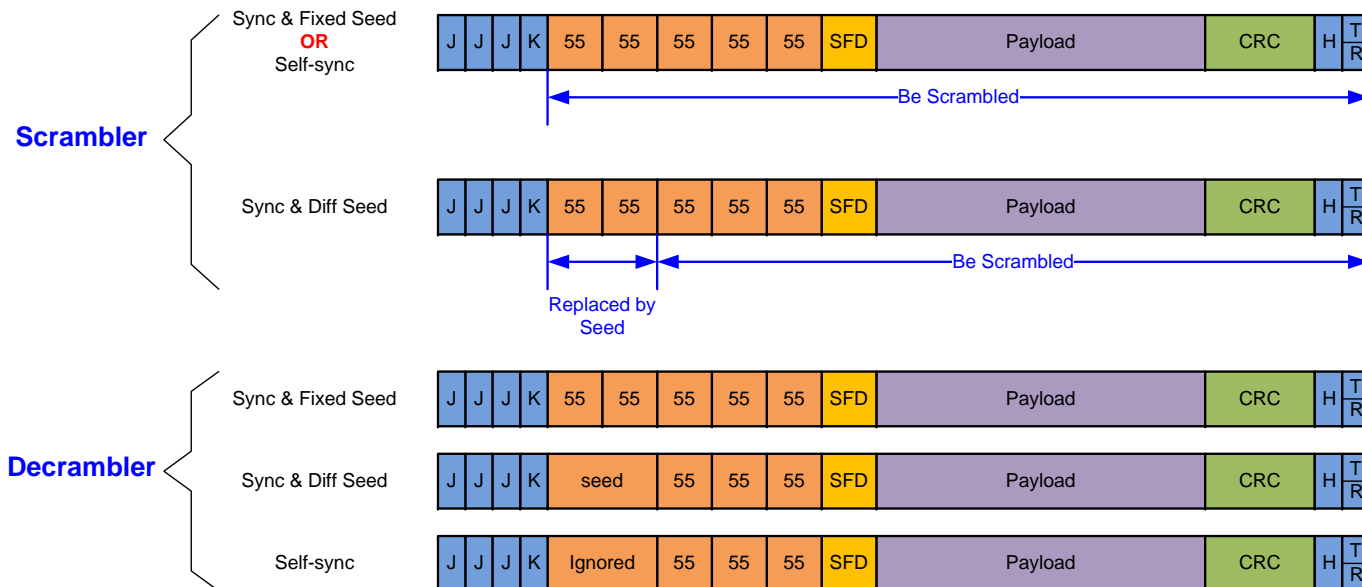
Scrambler Types

- Synchronous scrambler
 - With fixed seed for each frame
 - With different seed for each frame
- Self-synchronizing scrambler



Scrambler Types Comparison

- Synchronous scrambler with fixed seed
 - Same scrambled data if same payload data
- Synchronous scrambler with diff seed
 - Need transfer seed to receiver
- Self-synchronizing scrambler
 - The first 15 descrambled bits will be ignored





Scrambler Error Propagation

- Synchronous scrambler with fixed seed
 - A single-bit error at descrambler input will result an error on descrambled data
- Synchronous scrambler with diff seed
 - An error on transferred seed will result massive error propagation, as a complete frame cannot be correctly descrambled
 - A single-bit error at descrambler input will result an error on descrambled data
- Self-synchronizing scrambler
 - An error will result w errors (w =scrambler's taps, limited error propagation)

- **Packet**

- Payload length

- 60 byte
 - 170 byte

- payload

- Same payload (worse case)
 - Different payload (normal case)

- **Worst case**

- Shortest packets with same payload

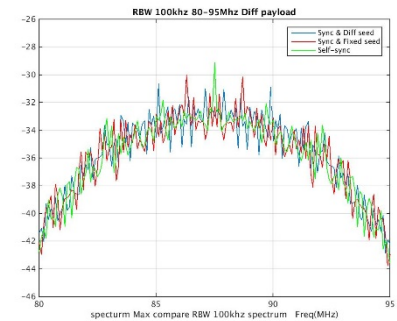
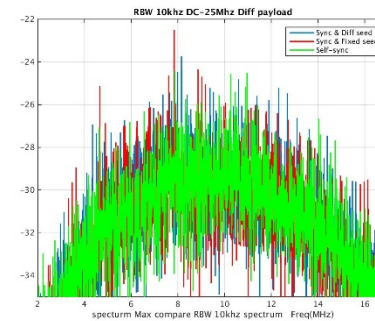
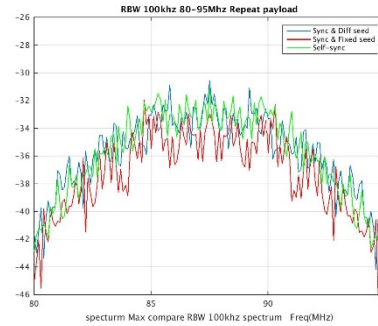
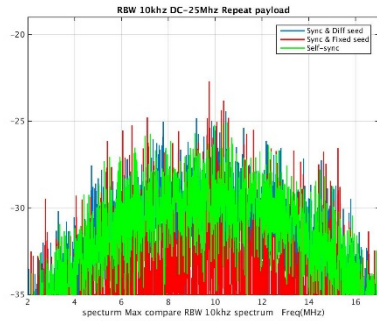
- 60 byte + same payload

60 Byte Payload

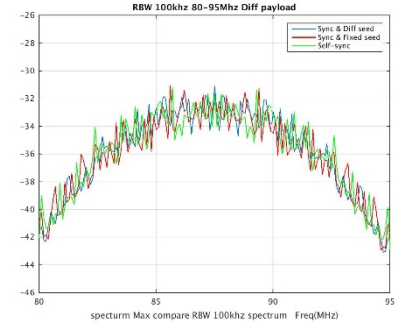
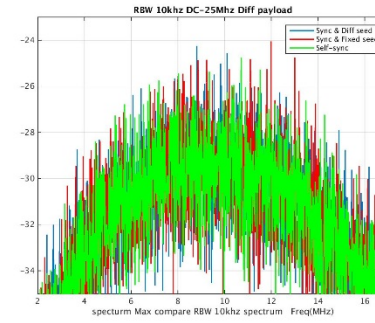
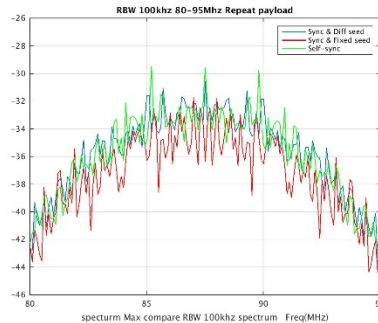
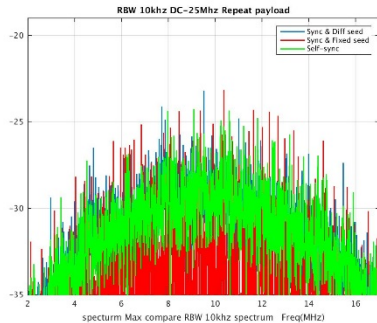
Same Payload

Different Payload

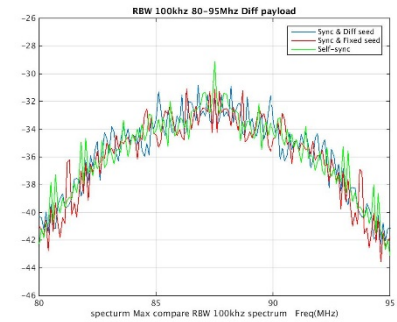
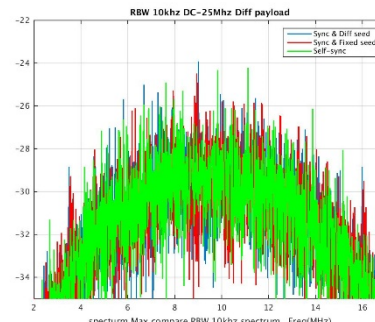
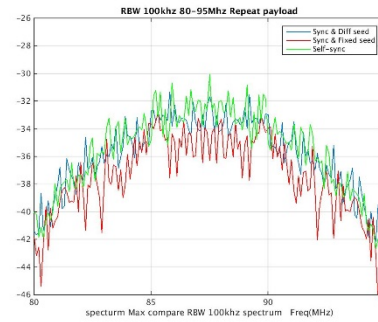
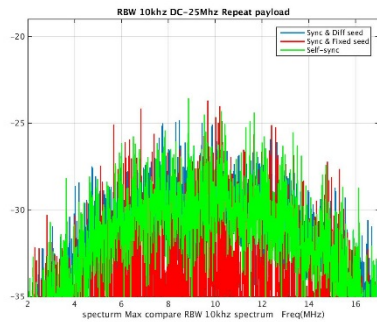
Pattern 1



Pattern 2



Pattern 3

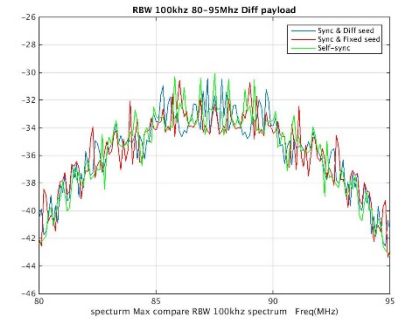
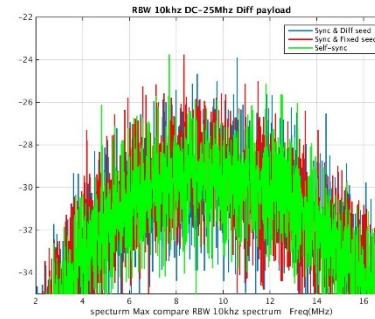
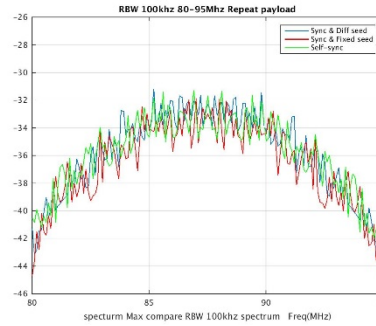
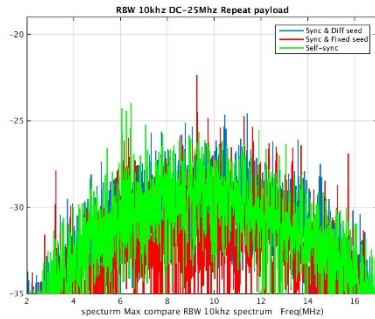


170 Byte Payload

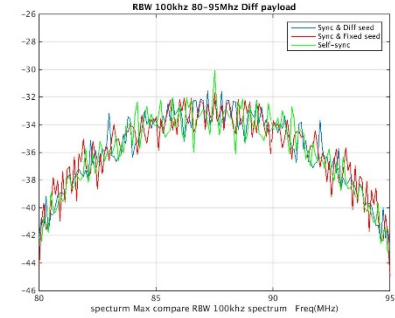
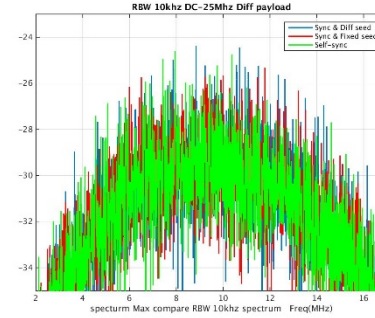
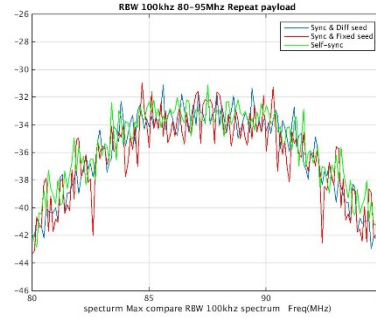
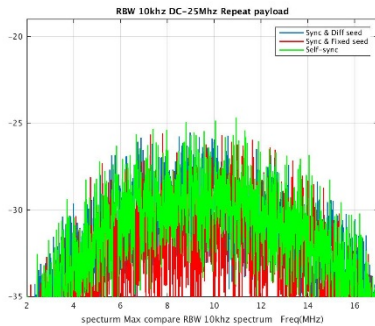
Same Payload

Different Payload

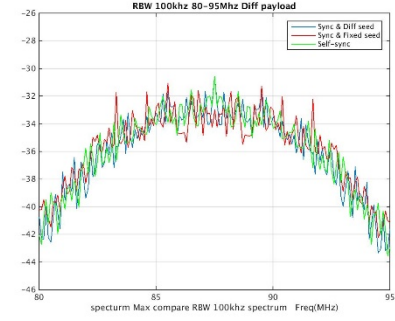
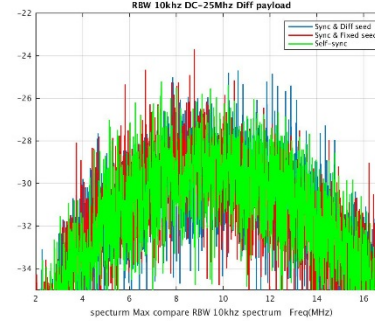
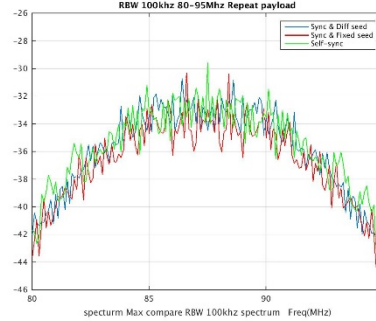
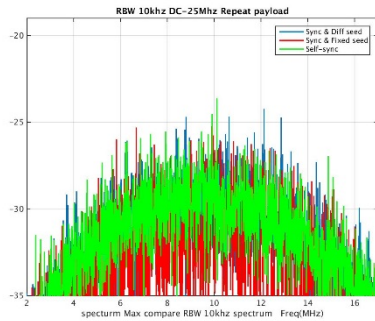
Pattern 1



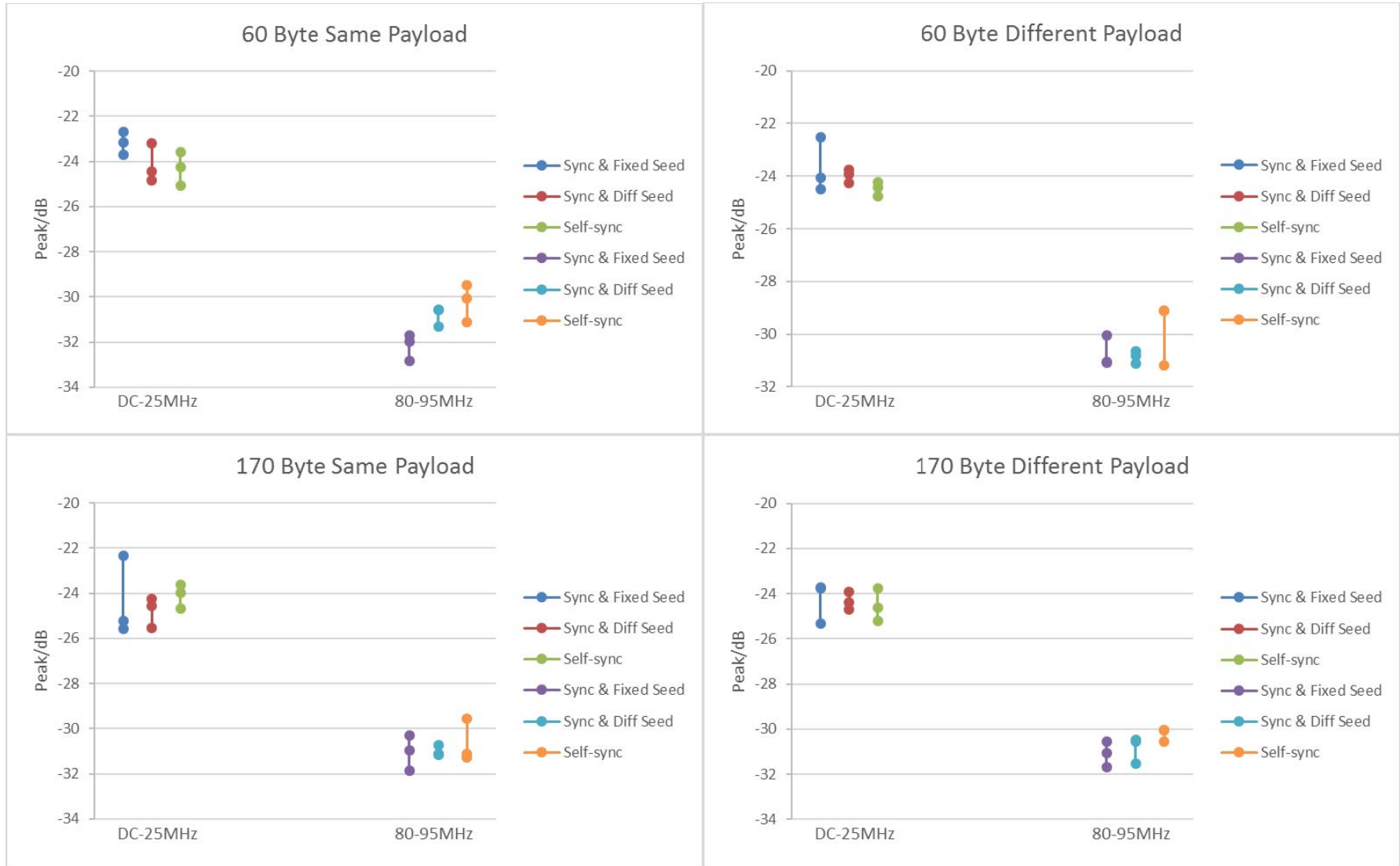
Pattern 2



Pattern 3



Peak Comparison





Scrambler Type Summary

Items			Scrambler		
			Sync & fixed seed	Sync & diff seed	Self-sync
Need transfer seed			N	Y	N
PSD Peack	Shortest packet	Same payload	1-2 dB High	low	
		Diff payload	Same		
	Not shortest packet		Same		
Error Propagation			None	Massive (error in seed)	Limited (# of Tap)

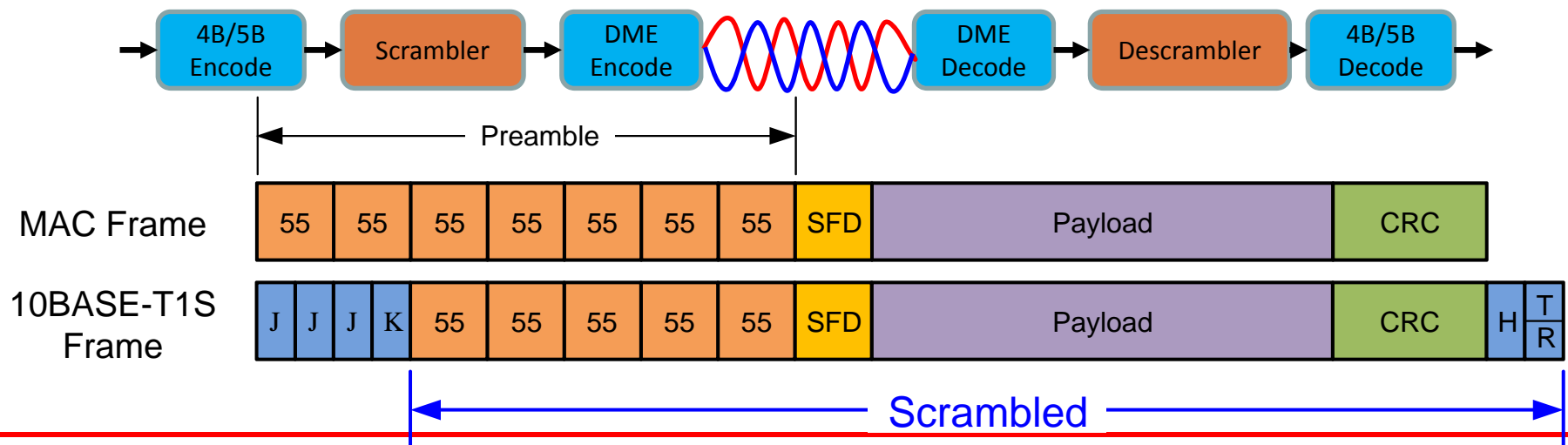
- Synchronous scrambler with fixed seed has no error propagation, but has a little higher peak emission around 10MHz with short repeat packet
- Synchronous scrambler with different seed has better peak emission with short repeat packet, but can not descramble the whole packet if has error on transferred seed
- Self-synchronizing scrambler has same peak emission as synchronous scrambler with different seed , and has limited error propagation
- The scrambler should be selected between synchronous scrambler with fixed seed and self-synchronizing scrambler



10BASE-T1S Scrambler Proposal

Based on the trade off for PSD and error propagation, we propose:

- 10BASE-T1S adopt synchronous scrambler with fixed seed
- Scrambler is inserted after the 4B/5B encoder and before the DME
- Scrambler:
 - Polynomial: $X^{15} + X^4 + 1$
 - Initial value: [0 0 1 0 1 0 0 1 1 0 0 0 0 0 1]
- Scramble all data after “JJJK”, including “ESD” and “ESDOK/ESDERR”
- Descrambler is initialized after receiving “K”



Conclusion

- A new scrambler for scrambling the preamble and payload is proposed for 10BASE-T1S
 - The proposed preamble scheme provides advantages:
 - No error propagation, with same level of PSD as other scrambler types
 - As effective at reducing peak PSD peak emissions as the method proposed by Cordao, Tazebay, et al.,
 - Maintains the DME self-clocking property for fast CDR (a few bits),
 - Retains two-level binary signaling,
 - Keeps the original frame preamble and format, and
 - No change in PLCA scheme needed and therefore no reduction in the efficiency of the network.
-