Micro-Reflection Limit

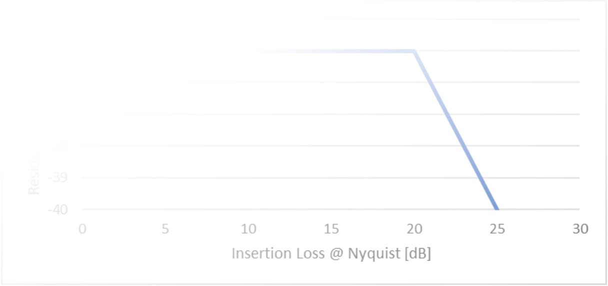
Contribution to IEEE 802.3cy

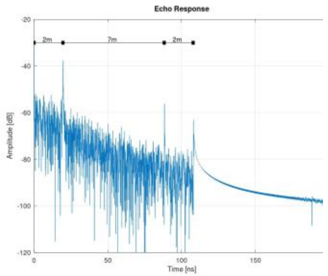
Ragnar Jonsson
Marvell

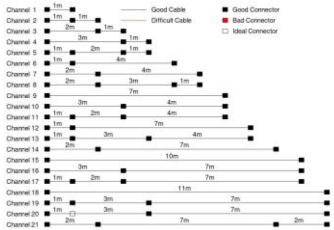
March 16, 2021

Introduction

- There have been several presentations on limiting micro-reflections:
  - jonsson_3cy_01a_0720
  - sedarat_3cy_01_0920
  - jonsson_3cy_01a_10_14_20
  - sedarat_3cy_02_10_14_20
  - sedarat_3cy_02_1120
  - jonsson_3cy_01_12_08_20
- This contribution presents specific text describing the limits for micro-reflections
- This contribution does NOT propose specific limits to use in the text







Micro-Reflection Limit Text

#1 of 4

Noise from echo outside of major discontinuities in a link segment beyond the capability required of the PHY to cancel echo is referred to as residual echo. To ensure the total residual echo energy is limited a figure of merit denoted as the Residual Echo Metric (REM) is specified. The REM is the remaining energy of a reflected impulse response after the largest time domain peaks of the reflected signal are removed.

The REM is determined using the following four-step procedure with the parameters in Table 1:

Table 1

Parameter	**Parameter Value**	**Parameter Description**
Δf_{max}	TBD	The maximum allowed frequency spacing for the frequency domain transfer function measurements
T	TBD	Time domain sampling interval
$f_{Nyquist}$	TBD	Nyquist frequency of the transmit signal (half the baud rate)
N	TBD	Number of sampling points to use for the time domain representation of the echo impulse response
N_{seg}	TBD	Number of samples in each segment
$N_{discard}$	TBD	Number of largest segments to discard
f_c	TBD	Reference frequency to use in calculation of the REM limit
REMmax	TBD	Upper bound on the REM limit
REMoffset	TBD	Offset of REM limit relative to IL at frequency f_c

Micro-Reflection Limit Text

#2 of 4

Step 1. The frequency domain transfer function for the differential mode channel echo, S_{11}, is measured at the link segment side of the MDI, e.g., the plug if the cable is terminated in a plug, and the card-edge a jack, with the far end terminated in 100Ω resistance. This measurement is performed for both ends of the link segment and provides the magnitude and phase of the transfer function, measured with frequency spacing, Δf. This frequency spacing needs to be no more than Δf_{max}.

Step 2. The frequency domain transfer function is converted to time domain impulse response with sampling interval, T, according to the following method:

Step 2a. The frequency domain transfer function is resampled to new frequency points uniformly distributed between 0Hz and $f_{Nyquist}$, with frequency spacing $1/T$. This resampling is done using Catmull–Rom spline as in Equation (xxx-1):

(Equation xxx-1)

$$E_k = S_{11}(m\Delta f + u) = \frac{1}{2}(-p_{m-1} + 3p_m - 3p_{m+1} + p_{m+2})u^3 + \frac{1}{2}(2p_{m-1} - 5p_m + 4p_{m+1} - p_{m+2})u^2 + \frac{1}{2}(-p_{m-1} + p_{m+1})u + p_m$$

where

$$p_m = S_{11}(m\Delta f)$$

$$m = \mathrm{floor}\left(\frac{k}{T\Delta f}\right)$$

$$u = \frac{k}{T\Delta f} - m$$

Micro-Reflection Limit Text

#3 of 4

Step 2b. The phase of H_k, the resampled signal, is adjusted to make the values at DC and Nyquist frequencies real. The adjustment is done by dropping any imaginary component at DC and applying linear phase adjustment, corresponding to fractional delay of the time domain signal, and is given by:

(Equation xxx-2)

$$H_k = E_k e^{-jk\theta}$$

$$H_0 = \mathrm{real}(E_0)$$

where

$$\theta = \frac{\mathrm{angle}\left(E_{K_N}\right)}{K_N}$$

$$K_N = \frac{N}{2}$$

Step 2c. The impulse response of the signal is computed by applying Hermitian symmetric extension of the signal above the Nyquist frequency as in Equation xxx-3:

(Equation xxx-3)

$$H_k = \mathrm{conj}\left(H_{K_N-k}\right), \qquad \text{for } n \in \{K_N+1, \ldots, 2K_N-1\}$$

and then computing the inverse Fourier transform according to:

(Equation xxx-4)

$$h_n = \frac{1}{K_N} \sum_{k=0}^{2K_N-1} H_k e^{j\frac{2\pi}{2K_N}kn}$$

Micro-Reflection Limit Text

#4 of 4

Step 3. The first *N/2* samples of the echo impulse response, h_n, is split into segments with N_{seg} samples in each segment. The square sum of each segment is computed

(Equation xxx-5)

$$P_r = \sum_{k=rN_{seg}}^{(r+1)N_{seg}-1} h_k^2$$

Step 4. The $N_{discard}$ largest P_r values are excluded and the square sum for the remaining $N/2\text{-}N_{discard}$ segments are added together to form the REM value:

(Equation xxx-6)

$$REM = 10\log_{10}\Big(\sum_{r\notin\{discard\}} P_r\Big) \quad (\mathrm{dB})$$

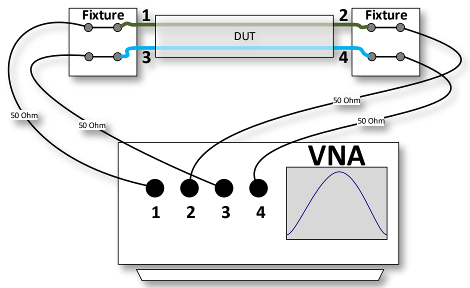
The REM value of each end of the link segment, defined by the calculation described in steps 1 through 4 above, using the parameters in Table 1, shall comply with Equation xxx-7:

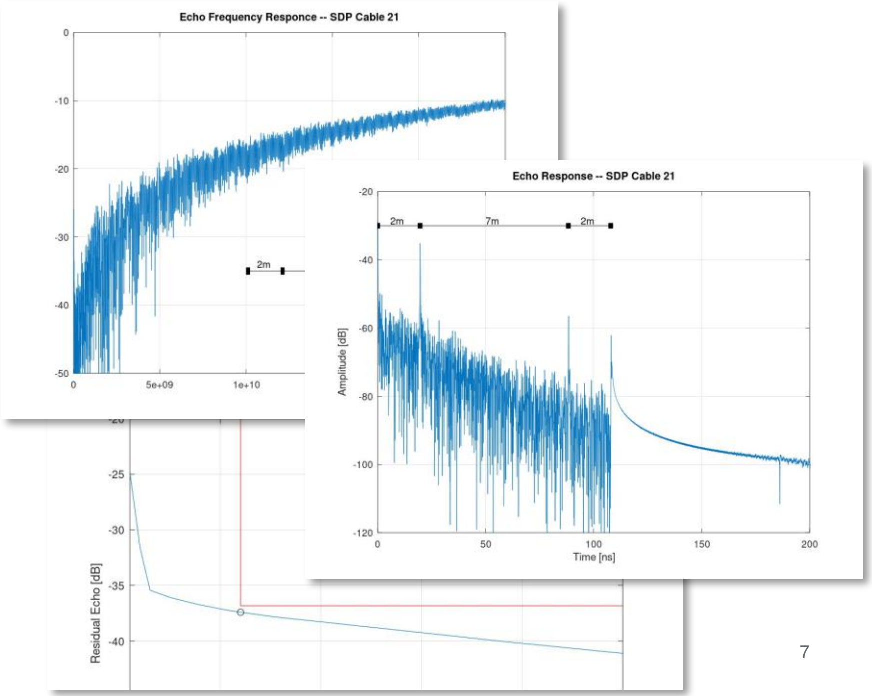
(Equation xxx-7)

$$REM \le \min(REMmax, -IL(f_c) - REMoffset) \quad (dB)$$

Practical Implementation of the Micro-Reflections Limit Method

- **Step 1** could be implemented with normal Vector Network Analyzer to measure S-parameters for the channel under test (DUT)
- **Step 2** could be implemented using the MATLAB/Octave code on following slides
- **Steps 3 & 4** could be implemented using the MATLAB/Octave code on following slides





Example Code for Steps 2

```
function h = cy_f2t(H,f,T,N)

%RJf2t - Impulse (time) response for a given frequency response.
%Usage:
%  h = RJf2t(H,f,T,N)
% where <H> is the frequency response given at frequencies <f>,
% <T> is the sampling interval, and <N> is the number of output
% samples (must be even).


%%% find size %%%
NN = prod(size(H));

%%% test arguments %%%
if( nargin < 2 )
  f = [0:NN-1]./(NN-1)*pi;
end;
if( nargin < 3 )
  T = 1;
end;
if( nargin < 4 )
  N = 256;
end;
N2 = ceil(N/2);
```

This is experimental code that is provided “as is”.

```
%%% find problem spots %%%
ix = find(H == H);
H = H(ix);
f = f(ix);

%%% re-shape arguments %%%
H = H(:);

%%% interpolate frequency response %%%
Hs1 = cr_spline(f*T,H,[0:N2]/N2/2);
ang_N = angle(Hs1(N2+1));
x0 = ang_N/(pi);
Hs1 = Hs1.*exp(-j*2*pi*x0*[0:N2]/N2/2);
Hs = [real(Hs1(1)) Hs1(2:N2) real(Hs1(N2+1)) conj(Hs1(N2:-1:2))];

%%% find impulse response from IDFT %%%
h = real(ifft(Hs));
```

Example Code for Step 2a

```
function s = cr_spline(x,y,xq)
% Catmull-Rom spline
%   s = cr_ spline(f*T,H,[0:N2]/N2/2);
% See more at https://en.wikipedia.org/wiki/Cubic_Hermite_spline

%%% initialize %%%
N_x = length(x);
[M_xq,N_xq] = size(xq);

%%% reshape arguments %%%
x = x(:);
y = y(:);
xq = xq(:);

%%% Catmull-Rom spline coefficients %%%
cr0 = [0 0 1 0];
cr1 = [0 1 0 -1]/2;
cr2 = [-1 4 -5 2]/2;
cr3 = [1 -3 3 -1]/2;

%%% find closest points for x-values %%%
xm = (x(1:end-1) + x(2:end) - 1e-10)/2; %% midpoint in interval
[d,m]=min(abs(reshape(xm,N_x-1,1) - reshape(xq,1,N_xq*M_xq)));
u = (xq - x(m))./(x(m+1) - x(m));

%%% Use Farrow structure to implement spline %%%
y0 = conv(y, cr0);
y1 = conv(y, cr1);
y2 = conv(y, cr2);
y3 = conv(y, cr3);

s = y3(m+2).*(u.^3)+y2(m+2).*(u.^2) + y1(m+2).*u + y0(m+2);

s = reshape(s,M_xq,N_xq);
```

This is experimental code that is provided “as is”.

Example Code for Steps 3 & 4

```
function [REM,p_residual,h_echo] = ureflections_test(f,s11)
%Evaluate micro-reflections
%Usage:
%  ureflections_test(f,s11)
%where <f> is frequency and <s11> is S11 parameters
%The function returns the metric value and plots the response.
%This function also uses the function RHJf2t().
%Version 1.0 -- March 9, 2021

%%% sampling interval and bin size %%%
f_s = 14062500000;
T = 1/f_s;     %% sampling interval
N_bins = 512;
L_bin = 4;     %% four samples per bin
N_discard = 12;
t_bin = L_bin*T;
t_max = N_bins * L_bin * T;

%%% number of samples and time vector %%%
N_samples = round(N_bins * L_bin)*2;  %even number of samples
t = [0:N_samples-1]*T;

%%% calculate echo impulse response and power %%%
h_echo = cy_f2t(s11,f,T,N_samples);
```

This is experimental code that is provided "as is".

```
%%% find power in each time bin %%%
h2 = h_echo.^2;
p_bin = zeros(1,N_bins);
m1 = 0;
for n=1:N_bins,
  m0 = m1 + 1;
  m1 = round(n*L_bin);
  p_bin(n) = mean(h2(m0:m1))*(L_bin);
end

%%% calculate effect of increasing number of bins %%%
[p_sort,sort_ix] = sort(p_bin);
p_sum = cumsum(p_sort);
p_residual = p_sum(end:-1:1);

%%% convert to dB %%%
REM = 10*log10(p_residual(N_discard));
endfunction
```

Conclusion

Specific text was presented describing limits on micro-reflections

Practical methods for implementing the text were shown for reference

We propose to adopt the text in slides 3 to 6 as baseline text in 802.3cy

MARVELL™

Essential technology, done right™