TO DETERMINE NETWORKED TELEMETRY RESYNCHRONIZATION TIME

Daniel T. Laird
AFFTC Edwards AFB CA

ABSTRACT

The Central Test and Evaluation Investment Program (CTEIP) Integrated Network Enhanced Telemetry (iNET) program is currently testing networked telemetry transceivers (IP.TM-Tx/Rx) using the Internet Protocol (IP), for use in telemetry (TM) channels. A unique characteristic of networked telemetry channel is packet drops due to radio frequency (RF) signal dynamics, i.e., terrain, weather, aircraft attitude, manmade objects, etc.. One of the key measures of the IP.TM-Tx/Rx is reliability is link availability (LA), and a key element of LA is time to resynchronize after RF link loss.

KEY WORDS

IP, TM, IP.TM-Tx/Rx, RF, OFDM, BPSK, QPSK, QAM, FEC, TDMA, Ethernet, Link-Layer, Transport-Layer, UDP, MTU, FPS, Sink, Source, Channel, LA, Differences, Delta-Counts, Delta-Time, Time-Window, Slot-Time, Epoch-Time, Resync-Time, Means Model, CI, ANOVA

INTRODUCTION

In this paper we discuss a method and results to model resynchronization time and simulate a degraded channel to estimate a measure on the time for a Teletronics Technology Corporation (TTC) IP.TM-Tx/Rx to resynchronize after RF signal loss. We test the IP.TM-Tx/Rx in a two-node degraded channel, under four *modulation schemes* of an Orthogonal Frequency Division Multiplexed (OFDM) signal, over three Time Division Multiple Access (TDMA) timing *epochs*. For our discussion, a modulation scheme entails a waveform type, which is either fixed or variable, a forward error correction code (FEC) and an adaptive power level. A timing epoch entails ratio of sink-to-source transmission time over a one second epoch.

TEST METHOD

To estimate the time to resynchronize – which we expected to be relatively short with respect to the transmission epoch – we created a model of epoch time variation based on known link availability. To simulate channel degradation we generated Ethernet traffic with a Spirent network emulator and switched the channel into binary states (ON | OFF) of high and low impedance to force RF signal loss and transceiver resynchronization. We processed time measures based on increasing and stale packet data, as described below, to statistically estimate the resynchronization time.

Degraded Channel and Tuned Payload:
We degrade the *channel* with a high dB attenuator to impede signal transception between two IP.TM-Tx/Rx carrying unidirectional universal datagram protocol (UDP) transport-layer payload from emulated wired Ethernet terminals. We use UDP to allow for dropped packets. The traffic flow is unidirectional, which we designate from *source* to *sink*. We 'tune' traffic payload (in bytes) and frames per second (FPS) between 'wired and wireless sides' of each node to a maximum transmission unit (MTU) in bytes for each modulation scheme such that there are no significant packet drops in the pristine 'ON' channel. We then capture transmission packets with an emulator application that samples time-tagged byte counts of the tuned traffic at the data-link layer and stores count-time samples in a spreadsheet for analysis.

Test Configuration-Traffic Emulator:
Using a Spirent Ethernet traffic emulator, we configure the two wired sides of the nodes. The wireless channel is linked through a switched (0|80dB) attenuator. The channel is switched over a 2s, 50% duty-cycle clock (½ -period = 1s, our *epoch-time*), which is triggered by the source GPS-1pps clock. We configure the emulator to capture time-stamped Ethernet packets at 1ms accuracy with precision 0.1μs over approximately 65s intervals, for each *modulation scheme* and *epoch timing*. The emulator is also GPS-1pps triggered for clock alignment.

Test Configuration-Transceivers:
Using the TTC transceiver command-line interface (CLI), we configure each for three timing epochs of 1:1, 10:10 and 4:16. These designators signify *sink* : *source slot-time* message ratios of the epoch-time. For each epoch we test four OFDM modulation modes: BPSK4, QPSK4, QAM164 and AUTO. The '4' represents ¾ forward error correction (¾FEC) coding. AUTO mode selects a modulation mode based on channel conditions, and is not restricted to ¾FEC. AUTO mode can and does run ½FEC. The power leveling is automated by an IP.TM-Tx/Rx algorithm.

DATA SAMPLING, MODELING AND PROCESSING

Data Sampling:
The Spirent emulator captures data packets at the link-layer and time-tags data accurate to 1ms intervals resolved to 0.1μs precision. There are ~65×10^3 samples over a ~65s sample interval, of which ~32 are in the 'OFF state' and ~32 'ON states' of 1s each. We sample all data and cull timing information to these 32 intervals of 1s plus the unknown time, δt, to resynchronize when the state transitions from OFF to ON. Thus our data model is resync-time is a positive variation of a unit: $1 + \delta t$. We use two differences and offsets to cull the variations, δt.

Data Analysis Model:
From the ~32 OFF and ON intervals, each duration >1s, we filter ~65×10^3 samples down to 32 data points of counts and time. We create a statistical *means-model* based on the equivalence of

1st *variation* of a 1 second epoch-time of the 2nd *difference* of our processing algorithm sample set:

$$\Delta^2\mathbf{t}(1) \leftrightarrow 1 + \delta\mathbf{t} \equiv 1 + \mu(\delta\mathbf{t}) + \boldsymbol{\varepsilon}. \quad (1)$$

Where $\mu(\delta\mathbf{t})$ is the mean of all time measures of a modulation scheme, $\boldsymbol{\varepsilon}$ is a vector of *sample residuals*, of an assumed zero mean ($\mu = 0$) normal-, $\mathcal{N}(0,\sigma^2)$, time series; $\mu(\delta\mathbf{t})$ serves as the variant origin. In our model, 1 is the unit of our timing, and 2nd difference $\Delta^2\mathbf{t}$ is the differenced measures modulo this timing unit: the timing unit is the epoch-time plus a sink bias-time due to epoch messaging structure and sample clocking. The epoch-time is a composite of transceiver *slot-times*, in which a node either receives (sinks) or transmits (sources) message traffic. Slot-time is a composite of 1ms *frames*, and minimum slot-time duration is 5ms, or 5 frames. From these details we create a model of delta-time as a function of resync-time for the sampled data and known *timing constants*:

$$\Delta^2 t^e_{m,k}(1_k) = 1_k + \delta t^e_{m,k};\ k \in \{1,\ldots,\mathrm{K}\}, \quad (2a)$$

$$1_k \equiv T_{1pps,k} + T^e_{snk,k} \quad \text{(measure time-unit)}, \quad (2b.1)$$

$$1_K \equiv T_{1pps} + T^e_{snk}. \quad \text{(measure time-series-unit)} \quad (2b.2)$$

Subscript '*m*' indexes the set of four modulation mnemonics: {BP4, QP4, QA4, A}; the superscript '*e*' indexes the set three epoch mnemonics: {1:1,4:16,10:10}; and K designates the samples culled via our processing the time-series per modulation mode. Assuming a simple means-model for resync-time we have in time-series *element form*:

$$\delta t^e_{m,k} = \mu(\delta\mathbf{t}^e_m) + \varepsilon^e_{m,k};\ k \in \{1,\ldots,\mathrm{K}\}. \quad (3a.1)$$

In a time-series *array form*, for all *k*, we rewrite (2) and (3a) as:

$$\delta\mathbf{t}^e_m = \mu(\delta\mathbf{t}^e_m) + \boldsymbol{\varepsilon}^e_m, \quad (3a.2)$$

$$\Delta^2\mathbf{t}^e_m(1_K) \equiv 1_K + \mu(\delta\mathbf{t}^e_m) + \boldsymbol{\varepsilon}^e_m. \quad (3b)$$

Where data array $\delta\mathbf{t}^e_{mk}$ designates the resync-times we seek, and $\boldsymbol{\varepsilon}^e_m$ is an array of scalar sampling errors, $\varepsilon^e_{m,k}$, the assumed normally distributed, zero mean of unknown variance: $\mathcal{N}(0,\sigma^2)$ – 'white-noise' – residuals. The sample clock constant, T_{1pps} is a *time window* on the epoch. Each clock ½-period is trigger-aligned on GPS-1pps edge controls switching 'ON and OFF' times. The sample clock period is $T_{sample} = 2s$, so the ON and OFF windows are $T_{1pps} = T_{sample}/2 = 1s$.

Tuning Payload:

To tune payload we match the Ethernet MTU (in bytes per frame (BPF)) to the radio data rate. All rates are based on these time constants:

$$T_{epoch} = 1000ms = 1s;\ T_{sample} = 2s;\ T_{½pps} = ½T_{sample} = 1s;\ T_{frame} = 1ms$$

$$T_{sink} = T_{source} = 5T_{frame};\ T_{grd} \equiv 1T_{frame}$$

And message structure satisfy:

$$T_{snk} \equiv \rho T_{sink};\ T_{src} \equiv \tau T_{source} \quad (4a.1)$$

$$T_{msg} = \rho T_{sink} + \tau T_{source} = (\rho + \tau)5T_{frame} = (\rho + \tau)5ms \quad (4a.2)$$

$$T_{snk}/T_{src} = \rho T_{sink}/\tau T_{source} = \rho 5T_{frame}/\tau 5T_{frame} = \rho/\tau \quad (4a.3)$$

The transmission ratio is a composite of source to message ratio,

$$r_{src/msg} = T_{src}/T_{msg} = \tau/(\tau+\rho), \quad (4b.1)$$

and source minus guard to source ratio,

$$r_{grd/src} = T_{src} - T_{grd}/T_{src} = (\tau-1)/\tau. \quad (4b.2)$$

So,

$$(r_{src/msg})(r_{grd/src}) = (\tau/\tau+\rho)(\tau-1/\tau) = (\tau-1)/(\tau+\rho). \quad (4c.1)$$

This is the ratio of 'guarded transmission' with respect to message. For obvious reasons we define the *radio transmission ratio*:

$$\tau_{radio} \equiv (\tau-1)/(\tau+\rho). \quad (4c.2)$$

The transmission ratio is dimensionless and serves as an *operator*. We can use it to find the radio *payload* transmission rate and transmission time:

$$R_{epoch} = \tau_{radio}(R_{frame}) = \tau_{radio}(10^3 \text{ FPS});\ T_{epoch} = \tau_{radio}(1000\text{ms}). \quad (4d)$$

Epoch and Ethernet loads are:

$$L_{epoch} = R_{epoch} \times b\text{BPF} = R_{epoch} \times h\text{MTU}, \quad (4e.1)$$
$$L_{eth} = R_{eth} \times b\text{BPF} = R_{epoch} \times h\text{MTU}. \quad (4e.2)$$
$$1 \leq h \leq 4,\ 751 \leq b \leq 3036$$

We tune the radio and Ethernet via equal MTUs:

$$L_{eth}/R_{eth} = h\text{MTU} = L_{epoch}/R_{epoch}, \quad (4f.1)$$
$$L_{eth} = R_{eth}(L_{epoch})/R_{epoch} = L_{epoch}(R_{eth}/R_{epoch}) = R_{eth}/R_{epoch}(L_{epoch}), \quad (4f.2)$$
$$R_{eth}/R_{epoch} = R_{eth}/\tau_{radio}(R_{frame}) = \tau^{-1}{}_{radio}(10^3 \text{ FPS})\, R_{eth}. \quad (4f.3)$$

This last is the Ethernet rate modulo the transmission ratio. We define the *Ethernet-Epoch rate ratio*:

$$\tau^{-1}{}_{eth/epoch} \equiv \tau^{-1}{}_{radio}\, R_{eth}/R_{frame}. \quad (4g)$$

So we can operate the epoch load to yield the Ethernet load. This is how we tune the payload traffic flow.

$$L_{eth} = \tau^{-1}{}_{eth/radio}(L_{epoch}). \quad (4h)$$

Data Processing:
The processing algorithm is:

1. Get time and count *arrays*: yields time-series matrix [**c**,**t**].
2. 1st delta-counts and time, Δ[**c**,**t**] with zeros: yields matrix [Δ**c**,Δ**t**], adjacent differences.
3. 2nd delta zeros removed: yields matrix [Δ^2**c**,Δ^2**t**], interval difference: '*delta-time*' and '*delta-counts*'.
4. Filter Δ^2**t** with 'time-offsets' via '*time-window*': W[Δ^2**t**| T_L,T_U]: yields '*resync-time*' (δ**t**) we seek as an error distribution.

The period drifts, and 'drift error' is realized with offset correction modulo the sum of epoch-time, T_{epoch} = 1s and bias-time, T^{e}_{bias}, which is sink slot-time duration, for each culled sample. We correct for these offsets and model them as arrays scalar errors. To offset we refer the epoch to the GPS-1pps: $(T_{1pps,k} - T_{epoch}) \in \varepsilon^{e}_{k}$; and the bias to the sink slot-time: $(T^{e}_{snk,k} - T^{e}_{bias}) \in \varepsilon^{e}_{b,k}$. We then fold these errors and the sample measure error into one $\mathcal{N}(0,\sigma_T^2)$ *residual distribution*. We have an error array:

$$\varepsilon^{e}_{m,b,k} \equiv \varepsilon^{e}_{k} + \varepsilon^{e}_{b,k} + \varepsilon^{e}_{m,k} = 1_k - 1_M + \varepsilon^{e}_{m,k};\ k \in \{1,\ldots,K\}. \tag{5a}$$

In array form for all *k*, and from (6b.3) below:

$$\boldsymbol{\varepsilon}^{e}_{m,b} \equiv \boldsymbol{\varepsilon}^{e} + \boldsymbol{\varepsilon}^{e}_{b} + \boldsymbol{\varepsilon}^{e}_{m} = 1_K - 1_M + \boldsymbol{\varepsilon}^{e}_{m}. \tag{5b}$$

Total error includes trigger clock drift, slot bias jitter and sampling errors. Twice differencing the matrix [**c**,**t**], yields (3b) we filter with theoretical offsets via a *time-window filter*:

$$W[\mathbf{t}|\ T_L,T_U] \rightarrow \mathbf{t}:\{t \in \mathbf{t}|\ T_L \le t \le T_U\}. \tag{6a.1}$$

This operation filters the resync-time plus sample-clock ½-period and sink slot-time biases, which contribute to 'total error' via this process. Using definition (3c):

$$W[\Delta^2\mathbf{t}^{e}_{m}|\ T_L,T_U] \rightarrow T_L \le 1_K + \mu(\delta\mathbf{t}^{e}_{m}) + \boldsymbol{\varepsilon}^{e}_{m} \le T_U, \tag{6a.2}$$

where:

$$T_L = 1_M + T_l \text{ (resync-time lower bound offset by unit)}, \tag{6b.1}$$
$$T_U = 1_M + T_u \text{ (resync-time upper bound offset by unit)}, \tag{6b.2}$$
$$1_M \equiv T_{epoch} + T^{e}_{bias} = 1 + T^{e}_{bias} \text{ (model time-unit)}. \tag{6b.3}$$

Offsetting the window filter by T_s yields:

$$W[\Delta^2\mathbf{t}^{e}_{m} - 1_M \mid T_L - 1_M, T_U - 1_M] \rightarrow T_l \le \mu(\delta\mathbf{t}^{e}_{m}) + 1_K - 1_M + \boldsymbol{\varepsilon}^{e}_{m} \le T_{u.} \tag{6a.3}$$

After processing, we have a matrix of 30 plus (K = 30+) samples. In (5d) we simplified (3b) using (2b.2), (5b), (6c.3) and we substitute

$$\tau^{e}_{m} \equiv \mu(\delta\mathbf{t}^{e}_{m}), \tag{7a}$$

to get:

$$\delta\mathbf{t}^{e}_{m} = \tau^{e}_{m} + 1_K - 1_M + \boldsymbol{\varepsilon}^{e}_{m}, \tag{7b.1}$$

or collecting all errors, simply:

$$\delta\mathbf{t}^{e}_{m} = \tau^{e}_{m} + \boldsymbol{\varepsilon}^{e}_{m,b}. \tag{7b.2}$$

These are the resync-times we seek as a data distribution. Also note that delta-counts are counted to filter delta-times (3b). We use counts to confirm payload, which should approximate the MTU of the Ethernet traffic 'tuned' to the modulation-epoch scheme: MTU^{e}_{m}. As it turns out, all

$$c \in \Delta^2\mathbf{c} \approx h MTU^{e}_{m}\ ;\ h \in \{1,\ldots,4\}. \tag{8}$$

All time-tagged counts are stored in spreadsheet format for easy export for later analyses. We discuss analyses next and then summarize our results in the conclusion.

DATA ANALYSES AND INTERPRETATION

We use Microsoft Excel™ and R (open source statistical program), for standard statistical analyses. We analyze via our *means model* to derive a Confidence Interval (CI) statistic on $\delta\mathbf{t}^e_m$ for *each* modulation mode *within* an epoch. We then extend the means model to an *effects model* for an Analysis of Variance (ANOVA) on $\delta\mathbf{t}^e_m$ for *all* modulation modes *across* epochs. We employ a one- and two-factor ANOVA respectively for modulation schemes within an epoch, and modulation within and across epoch.

TEST RESULTS

Confidence Interval:
Figure 1 shows CI results of a 2-node channel for epoch 1:1 at 5ms. From the table we can list the resync-times (in ms) to 95% confidence of 31 degrees of freedom, using the simplified nomenclature yields:

$$CI_p(\tau^e_m) = \tau^e_m \pm q\underline{\sigma}(\delta\mathbf{t}^e_m)/\sqrt{n}. \qquad (9a.1)$$

For our data and 95% confidence:

$$CI_{.95}(\tau^e_m) = \tau^e_m \pm (2.0395)\underline{\sigma}^e_m/\sqrt{31}. \qquad (9a.2)$$

For our data in a 2-node TDMA of 1:1 epoch ratio at 5ms frames, these resolve as:

- $CI_{.95}(\tau^{1:1}_{BP4}) = 2.0006 \pm 0.0012$
- $CI_{.95}(\tau^{1:1}_{QP4}) = 1.9894 \pm 0.1039$
- $CI_{.95}(\tau^{1:1}_{QA4}) = 1.9995 \pm 0.0009$
- $CI_{.95}(\tau^{1:1}_{A}) = 2.4691 \pm 0.1784$

The $CI_{.95}$ results of 4:16 ratio at 5ms are:

- $CI_{.95}(\tau^{4:16}_{BP4}) = 3.2438 \pm 0.1560$
- $CI_{.95}(\tau^{4:16}_{QP4}) = 2.5006 \pm 0.1992$
- $CI_{.95}(\tau^{4:16}_{QA4}) = 2.008 \pm 0.0007$
- $CI_{.95}(\tau^{4:16}_{A}) = 2.0310 \pm 0.0622$

The $CI_{.95}$ results of 10:10 ratio at 5ms are:

- $CI_{.95}(\tau^{10:10}_{BP4}) = 3.1879 \pm 0.2890$
- $CI_{.95}(\tau^{10:10}_{QP4}) = 2.0933 \pm 0.1042$
- $CI_{.95}(\tau^{10:10}_{QA4}) = 2.808 \pm 0.1608$
- $CI_{.95}(\tau^{10:10}_{A}) = 2.2187 \pm 0.1479$

The length of the confidence interval is

$$|CI(\tau)_{.95}| \approx 2q\underline{\sigma}/\sqrt{n}. \qquad (9b)$$

This interval centers on sample mean time τ. The *sample standard deviatio*n, σ, is an estimate of *population standard deviation* for resync-times; σ/√n denotes the *standard error* for 'n' degrees of freedom, and *q* is a quantile weighting factor of the residual-distribution. For our tests, we chose 95% confidence, so $q = 2.0395$ for a Normal distribution.

	BPSK4	QPSK4	QAM164	AUTO
	1.9983	2.9990	1.9999	2.0000
	1.9983	2.9992	1.9997	2.0003
	1.9983	2.0000	1.9997	1.9996
	2.0002	2.0000	1.9997	2.0000
	2.0014	1.9998	2.0018	2.0004
	2.0002	2.9901	1.9999	2.0002
	2.0000	1.9998	1.9987	2.0020
	2.0001	2.0003	2.0000	2.0011
	2.0000	2.0000	1.9999	3.0003
	2.0003	1.9983	1.9999	3.0000
	1.9989	1.9981	1.9998	2.9995
	2.0004	2.0011	2.0003	3.0003
	2.0000	2.0021	2.0136	2.9997
	2.0019	1.9999	1.9998	3.0001
	2.0021	2.0003	1.9997	2.9998
	2.0174	2.0001	1.9999	3.0002
	2.0018	2.0001	1.9996	3.0018
	2.0020	1.9986	1.9999	2.9991
	2.0019	2.0001	1.9998	2.9997
	2.0021	1.9999	1.9998	3.0000
	2.0000	2.0001	1.9999	2.9986
	2.0001	2.0022	1.9999	3.0003
	1.9980	2.0018	2.0017	2.0002
	1.9981	2.0000	2.0018	3.0003
	1.9983	1.9999	2.0000	2.0017
	2.0019	1.9999	1.9999	2.0019
	1.9998	2.0001	2.0011	2.0016
	2.0001	1.9981	1.9997	1.9999
	2.0002	1.9982	1.9997	1.9997
	2.0000	1.9980	2.0000	2.0002
	1.9985	1.9979	1.9975	2.0009
	1.9984	2.0001	1.9999	2.0002
μ(δt):	2.0006	2.0932	2.0004	2.4691
σ(δt):	0.0033	0.2950	0.0025	0.5067
95% cf:	0.0012	0.1039	0.0009	0.1784
95% CI				
lcl	1.9994	1.9894	1.9995	2.2907
ucl	2.0018	2.1971	2.0013	2.6474
|CI|	0.0024	0.2077	0.0018	0.3567

EPOCH 1:1

Anova: Single Factor						
SUMMARY						
Modulations	*Count*	*Sum*	*Average*	*Variance*		
BPSK4	32	64.01898	2.000593	1.12E-05		
QPSK4	32	66.9831	2.093222	0.087053		
QAM164	32	64.01249	2.00039	6.44E-06		
AUTO	32	79.0096	2.46905	0.256747		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Modulations	4.78031	3	1.593437	18.53814	5.34E-10	2.677699
Within Modulations	10.65836	124	0.085955			
Total	15.43867	127				

Figure 1: Data and 95% CI for Resync-Time Statistics for 1:1 Epoch at 5ms

ANOVA:

We ran ANOVA to find any statistical significance between resync-times within and across modulations, and across epochs. Figure 1 also shows the 'one-way' ANOVA for epoch 1:1 alone. The model is based on equation (3), which contains one dependent variable, resync-time, and one independent factor, modulation. Note the very small '*p-value*' – an indicator of the probability of occurrence that the variation between modulations being significantly different than the variation within any one modulation scheme. The very small probability of this random event indicates there is a statistically significant difference between resync-time due to modulation. But for practical purposes, the differences are not important.

Anova: Two-Factor With Replication

SUMMARY	BPSK4	QPSK4	QAM164	AUTO	Total
EPOCH1:1					
Count	32	32	32	32	128
Sum	64.01898	66.9831	64.01249	79.0096	274.0242
Average	2.000593	2.093222	2.00039	2.46905	2.140814
Variance	1.12E-05	0.087053	6.44E-06	0.256747	0.121564
EPOCH10:10					
Count	32	32	32	32	128
Sum	102.0125	66.9842	72.9842	71	312.9809
Average	3.187891	2.093256	2.280756	2.21875	2.445163
Variance	0.674071	0.08754	0.208591	0.17647	0.469824
EPOCH4:16					
Count	32	32	32	32	128
Sum	103.8013	80.01809	64.02531	64.9906	312.8353
Average	3.243791	2.500565	2.000791	2.030956	2.444026
Variance	0.196502	0.320102	3.68E-06	0.031187	0.388178
Total					
Count	96	96	96	96	
Sum	269.8328	213.9854	201.022	215.0002	
Average	2.810758	2.229014	2.093979	2.239585	
Variance	0.61625	0.198685	0.085696	0.184086	

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Epochs	7.874879	2	3.93744	23.18091	3.25E-10	3.019987
Modulations	29.23162	3	9.743874	57.36516	1.68E-30	2.628903
Interaction	31.98646	6	5.331076	31.38567	1.66E-30	2.122964
Within	63.18681	372	0.169857			
Total	132.2798	383				

Figure 2: ANOVA for Resync-Time across All Epochs

For the 'two-way' ANOVA we model variation due to modulation and epoch structure, so we extend the mean of equation (3) to an *effects model* by extension of the mean to include effects due to modulation and epoch:

$$\tau^{e}_{m} \equiv \tau + \alpha_{m} + \beta^{e}. \qquad (10a)$$

If we assumed interaction between effects:

$$\delta t^{e}_{m,k} = \tau + \alpha_{m} + \beta^{e} + (\alpha\beta)^{e}_{m} + \varepsilon^{e}_{m,k}\ ;\ k \in \{1,\ldots,K\}. \qquad (10b)$$

Where, τ, is the *grand-mean*, α_m is the *modulation effect* and β^e is the *epoch effect*, and $(\alpha\beta)^e_m$ denotes the *interaction effect* of modulation and epoch, $\varepsilon^e_{m,k}$ is the error. In this model, the ‘effects’ cause variation around the *grand-mean*, τ. The two-factor ANOVA compares all modulations and epochs, as shown in Figure 2. I’ve left the data table out, as it is too long to fit legibly. All data is included in the attached Excel workbook, and available on request.

CONCLUSION

We have presented a viable test method for determining a very small duration of time for a known TDMA epoch of a commercially available networked telemetry transceiver to resynchronize. We were able to isolate and analyze this resynchronization time using standard network testing tools, statistical models and analyses. In the ANOVA results, small *p-values* of the analyses indicate that there is a statistical significance with respect to modulation and epoch structure (see row *p*-value in one- and two- factor tables), i.e., the variation across modulation schemes is statistically significant with respect to modulation within an epoch structure. If we examine resync-time over epochs, the average time is also significantly different with respect to epoch. But these differences have no practical effect on channel recovery due to signal dynamics. The small CIs for each modulation scheme suggests our samples are a good estimate of the ‘unknown population mean’ that represents the transceiver resync-time and epoch reacquire-time. Whatever the form of ANOVA model employed, we are confident that resync-time never exceeds the duration of minimum slot-time of 5ms.

ACKNOWLEDGEMENTS

The author would like to thank the OSD/CTEIP program office for the support and opportunity to develop these prototypes and evaluate performance under the iNET Program. Thanks in particular go to iNET Program Manger, Bruce Lipe. The author also thanks Mark Radke and Glen Wolf for their technical assistance and discussions. As always, thanks to Kip Temple for his confidence in our work.