Time Iteration Protocol for TOD Clock Synchronization

Eric E. Johnson

January 23, 1992

Introduction

This report presents a protocol for bringing HF stations into closer synchronization than is normally achievable using the Time Service protocols described in the Linking Protection Implementation Guide (January 1991). The technique described here employs exchanges of time and time difference reports among stations to enable the participating stations to estimate both the differences among their local times and the transmission and processing delays among the stations. (The technique was initially developed by Gene Harrison and the author on 31 March 1990 in a New Mexican cantina!)

To review, the Time Service protocols employ Time Command, Coarse Time, and Authentication words in the formats shown in Figure 1 to request and to provide time.

3	7	3	6	5
C M D	Time Service	Time Quality	Seconds	40 ms ticks

3	1	4	5	11
D A T A	0	Month	Day	Minute

3	21
R E P	Authenticator

Figure 1: Time Command, Coarse Time, and Authentication Word Formats

The typical Time Service message section includes one of each of these words, as shown in Figure 2, and conveys to the recipient the local time at the sending station with 40 ms *precision*, an estimate of the *accuracy* of the report in the Time Quality field, and authentication of the time, if the transmission in question is not the first of the exchange (as discussed below). Linking protection for Time Service protocol transmissions uses the sender's current coarse TOD (i.e., the seconds field is set to –1), which may differ from that used by the other station.

TO <requester> **CMD** *Time Is* <Time> **DATA** <Coarse Time> **REP** <authenticator> **THIS WAS** <time server>

Figure 2: Typical Time Service Transmission

The authenticator in the authentication word is the 24-bit result of the 3-way exclusive-or of the command word and the coarse time word from the transmission to be sent, with the authentication word (including the preamble) from the previous transmission. In the case of the first transmission of a handshake, no previous authenticator is available to use as a "challenge" from the other station, so no protection is provided against playback attacks.

The Time Quality field in the Command word is encoded as shown in Table 1 to indicate the uncertainty at the sending station of its local time (e.g., an uncertainty of ±300 ms results in a window of 600 ms, and a time quality of 4). Stations maintain an estimate of their local uncertainty window that includes the uncertainty in the last setting of the time plus a bound on the drift that may have occurred since then. This window is converted to a time quality code only for transmission. If a station that receives a time value decides to use that value to set its local time, the initial uncertainty of that time is set to the full window indicated in Table 1 for the received time quality, plus an estimate of additional uncertainty due to variations in propagation and local processing delays. Thus, the receiver of quality Q time will at best have quality Q+1 time.

Table 1: Time Quality

Time Quality Code	Time Uncertainty Window
0	none
1	20 ms
2	100 ms
3	500 ms
4	2 s
5	10 s
6	60 s
7	unbounded

<u>Time Iteration Protocol</u>

The intent of the Time Iteration protocol is to allow stations to estimate the differences in their local times, and the propagation and processing delays between pairs of stations, so that network members may be more closely synchronized than via the Time Service protocol, whose principal function is to provide time to stations without accurate time. Because of this difference in intent, the Time Service protocol assumes only that the participating stations are in coarse sync (i.e., correct month, day, and minute only), and uses only coarse TOD in applying linking protection to its transmissions, while the Time Iteration protocol assumes that stations are in fine sync (correct seconds as well), and uses full linking protection for its transmissions.

In place of the Coarse Time word used in the Time Service protocol, the Time Iteration protocol uses a Delta Time word as shown in Figure 3. This word carries, in sign + magnitude format, the difference calculated between the time contained in a received time report and the local time (at the sender of this Delta Time word) of the time of receipt of that time report. Note that the Delta Time word is distinguished from the Coarse Time word by the presence of a 1 bit immediately after the preamble. In some cases, more than one Delta Time word may be sent between the Time Command word and the authentication word, as determined by the Status field.

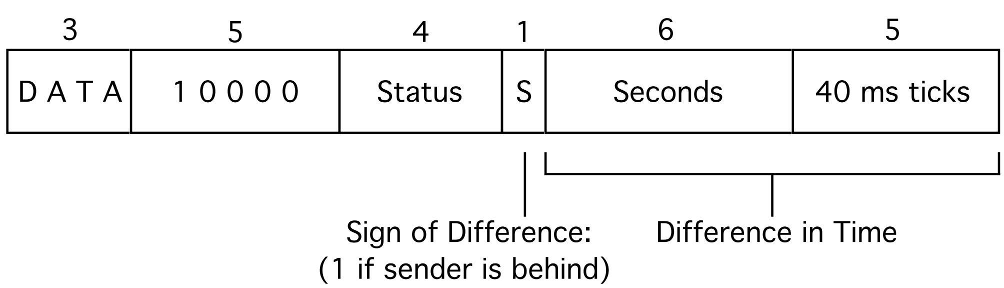

Figure 3: Delta Time Word Format

The Status field in the Delta Time word is encoded as follows:

<u>Status</u> <u>Meaning</u>

0 TIME ITERATION REQUEST; sign is set to 0; seconds and ticks fields contain current time uncertainty window at sender; always final Delta Time word.

1 TIME ITERATION CONTINUE; the seconds and ticks fields report the absolute value of the difference between the time last reported by the addressee and the time the sender received that report; sign = 1 if that reported time was later than the sender's local time. Requests continuation of iteration; final Delta Time word.

2 CONCLUDE TIME ITERATION; delta time reported as above; suggests termination of time iteration; final Delta Time word. Other station may continue (1) or terminate (6).

3 DELTA TIME REPORT; reports the difference in local times between the sender of the report and another station whose identity is implicit in the protocol (e.g., in slotted-response order). Used to disseminate computed differences, rather than the raw differences reported under codes 1 and 2. Always followed by another Delta Time word.

4 TIME UNCERTAINTY REQUEST; requests a report of recipient's current local time uncertainty; sign is set to 0; seconds and ticks fields contain current time uncertainty window at sender. Always followed by another Delta Time word.

5 TIME UNCERTAINTY REPORT; sign is set to 0; seconds and ticks fields contain current time uncertainty window at sender. Always followed by another Delta Time word.

6 TERMINATE TIME ITERATION; delta time reported as in 1; mandatory termination of time iteration; final Delta Time word. (Requests to continue will be ignored.)

15 NAK; cannot perform requested operation; sign, seconds, and ticks fields are set to all 1s; always final Delta Time word.

All other encodings are reserved until standardized.

Two-Station Time Iteration

The Time Iteration protocol works as shown in Figure 4. The absolute time at which the exchange shown commences is denoted T. The local time at station A is assumed to differ from absolute time by **a**; similarly, local time at B differs from absolute time by **b**.

<u>Absolute Time</u>	<u>Event</u>
T	Station A requests time iteration and reports its local time, which is $T + a$, and its time uncertainty window.
$T + d_{ab}$	After propagation and processing delays, this transmission arrives at B at local time $T + b + d_{ab}$. B computes the difference between the reported time and the time of its arrival: $\Delta_{B1} = b + d_{ab} - a$.
$T + m$	B sends its local time $T + b + m$, its uncertainty window in a Time Uncertainty Report (<u>TURpt</u>), and Δ_{B1} in a Time Iteration Continue (<u>Cont</u>) word to A.
$T + m + d_{ba}$	This transmission arrives at A (local time at A is $T + a + m + d_{ba}$). A computes the difference $\Delta_{A1} = a + d_{ba} - b$. The round-trip propagation delay $d_{ab}+ d_{ba}$ can now be found as $\Delta_{A1} + \Delta_{B1}$; if $d_{ab} = d_{ba} = d$, then $d = (\Delta_{A1} + \Delta_{B1})/2$, and the

	difference in local times at A and B (a – b) can be estimated: $\Delta_1 = (\Delta_{A1} - \Delta_{B1})/2$.
T + n	A sends its local time T + a + n and Δ_{A1} to B.
$T + n + d_{ab}$	This report arrives at B (local time at B is $T + b + n + d_{ab}$). B computes the difference $\Delta_{B2} = b + d_{ab} - a$, and can now compute two samples of the difference in local times: $\Delta_1 = (\Delta_{A1} - \Delta_{B1})/2$ and $\Delta_2 = (\Delta_{A1} - \Delta_{B2})/2$.

The stations will continue the iteration until enough samples of the difference in local times (a – b) are obtained to estimate this difference to within some confidence interval (based upon their time qualities), after which the station with the larger uncertainty window can update its time using the estimated difference, and reset its uncertainty window to that of the other station plus the size of the confidence interval of the estimate.

In this way, it is possible for a station with time quality Q to bring other stations up to the same time quality (certainly achievable if Q ≥ 2, perhaps for Q = 1), although the updated stations will have slightly larger uncertainty windows.

Absolute Time	Local Time A	EVENT	Local Time B
T	T+a	>>> Time Is T+a | Time Iteration Request >>>	
$T+d_{ab}$		>>> Time Is T+a | Time Iteration Request >>> *B computes* $\Delta_B = (T+b+d_{ab}) - (T+a)$ $= b+d_{ab}-a$	$T+b+d_{ab}$
T+m		<<< Time Is T+b+m, TURpt, Cont ($\Delta_B = b+d_{ab}-a$) <<<	T+b+m
$T+m+d_{ba}$	$T+a+m+d_{ba}$	<<< Time Is T+b+m, TURpt, Cont ($\Delta_B = b+d_{ab}-a$) <<< *A computes* $\Delta_A = (T+a+m+d_{ba}) - (T+b+m)$ $= a+d_{ba}-b$	
		• • •	

Figure 4: Two-Station Time Iteration

Note that after N + 1 transmissions, one station will have obtained N samples, the other N – 1.

The authentication procedure for the Time Iteration protocol is identical to that of the Time Service protocol: the last word of the message is an authentication word, with a 21-bit authenticator computed as the exclusive-or of the authenticator from the preceding transmission (if any) with all words preceding the authentication word in the current message. Because the number of words in the message may vary, the preamble of the authentication word may be either DATA or REP.

Network Time Iteration

In network operations, the Time Iteration protocol can be used to synchronize all network members to a station of singular time quality (a time standard), or the protocol can be used to estimate differences in the local times of all network members, pair by pair, and thereby statistically achieve a consensus network time in the absence of a time standard.

Both variations use a net call variant of the protocol, in which one station polls the remaining stations with a Time Iteration Request net call; net members respond in their assigned slots, the caller acks with a list of Delta Time Reports or Time Iteration Continue words, net members again make slotted responses, and so on. Clearly, this net call technique could consume significant amounts of air time, especially in the multi-point mode (no time standard).

Analysis

The following analysis addresses only the two-station case of time iteration. However, the generalization to the net/group and multipoint cases is straightforward.

Mathematical Background

The task of estimating the difference in local times between two stations is complicated by variations in propagation times and processing times of transmissions between those stations. Let the actual difference in local times between the two stations A and B be denoted Δ (which should not change appreciably over the period of the sequences of transmissions considered here), and consider the probability density $f_D(d)$ of the propagation plus processing delays experienced by time iteration transmissions. Each measurement of the difference between the time of arrival of a transmission and the timestamp of that transmission will be a measurement of $\pm\Delta + d$. The difference between successive measurements is a random variable (a function) depending upon Δ and two samples from $f_D(d)$:

$$\Delta_i = (\Delta_{Ai} - \Delta_{Bi})/2 = \Delta + (d_{ba} - d_{ab})/2$$

If we assume that the samples from $f_D(d)$ are independent, then the distribution of the Δ_i's has mean Δ and the same variance as $f_D(d)$. Given a sequence of such measurements, we can estimate Δ, and we can compute a confidence interval for that estimate from the number of samples (of Δ_i) taken and the variance of those samples.

The delay term d is composed of many factors, including the difference between the timestamp in the transmission and the actual time of release of the word following the Time Is word, propagation delays through the HF channel, and processing delays at the receiver which are not accounted for in the receive processing of the protocol. The components of d which contain a variable factor are as follows:

1. Propagation time may vary by 1 - 2 ms, due to multipath and other path effects.
2. The timestamp in the Time Is word may vary from the actual time of release of the following word due to word phasing requirements (392 ms) or a free-running baud clock (8 ms).
3. The 40 ms resolution of time reports will introduce round-off errors if transmissions are not released precisely at 40 ms ticks.
4. Varying processing times may introduce uncertainty into the times of transmission and of receipt of time iteration words. Manufacturers estimate 1 ms of variation here.

Some of these variations in d are uniformly distributed (e.g., word or baud phase delays), while others may be normally distributed. The resulting density $f_D(d)$, as a sum of these terms, will probably approximate a normal distribution, as will the distribution of the Δ_i's.

The procedure for estimating Δ is simply to compute $E[\Delta_i]$, the mean of k samples of $\Delta_i = (\Delta_{Aj+0/-1} - \Delta_{Bj})/2$. Computing a confidence interval for this estimate follows the classic procedure for estimating a confidence interval for the mean of a normal distribution when the variance is unknown: $(E[\Delta_i] - \Delta)/(s/\sqrt{k})$ has a t distribution with (k – 1) degrees of freedom, where ***s*** is the sample standard deviation of the Δ_i's, computed as follows:

$$s = \sqrt{\frac{\left(\sum_{i=1}^{k} \Delta_i^2\right) - k\,E[\Delta_i]^2}{k-1}}\ .$$

We can therefore compute a 99% confidence interval for Δ as $E[\Delta_i] \pm t_{0.995}(k-1)[s/\sqrt{k}]$. The width of this confidence interval is added to the uncertainty window of the time source to give the new time uncertainty window at the station updating its time.

For example, assume that station B has a 1900 ms uncertainty window while A's window is 500 ms, and that they have exchanged three delta time measurements, with a mean of these three measurements of +100 ms, and a sample standard deviation of 15 ms. Since $t_{0.995}(2)/\sqrt{3} = 5.73$, the confidence interval is 2•(5.73)•15 = 172 ms. B could terminate the iteration at this point, advance its clock by 100 ms, and set its uncertainty window to 500 + 172 = 672 ms.

Terminating the Iteration

The number of samples required in an iteration is determined by the size of the confidence interval which must be achieved and the standard deviation of the samples. In operation, a small number of samples (say 3) must be taken to estimate the standard deviation ***s***; then the total number of samples required can be computed by dividing the confidence interval desired by ***s***, and looking up the number of samples needed in a table such as that in Appendix A.

The following discussion considers stations with ±10 ppm time bases which synchronize with another station once a day to maintain time quality 4 (an uncertainty window of no more than 2 seconds). Because these time bases can drift up to ±864 ms in 24 hours (1728 ms total), the initial uncertainty window achieved by time iteration must be no more than 272 ms. Four cases are considered, as described in the table below:

Table 2: Four Cases of Timing Uncertainties

(all times in ms)

Source of Variation	I	II	III	IV
Propagation time	2	2	2	2
Word phasing	-	-	-	392
Baud phasing	-	8	8	-
Time report resolution	-	-	40	40
Processing times	1	1	1	1
Sample standard deviation ***s***	0.87	3.18	14.7	126

All cases include 2 ms of variation in propagation time and 1 ms of uncompensated variation in processing time. Case II adds 8 ms of uniformly-distributed delay for baud phasing in the modem. Case III adds to that 40 ms of uniformly-distributed rounding error for a controller which doesn't control the time of release of transmissions with respect to the 40 ms intervals reported in the time words. Finally, case IV includes word phase tracking implemented in the FEC processor or the modem, without a synchronization path to the ALE/LP processor; rather than delaying words from 0 to 8 ms while awaiting the next baud boundary, this requires a uniformly distributed delay of up to one T_{rw}. The last row in the table shows the expected values of the sample standard deviations.

If every station has direct connectivity to a time standard (whose uncertainty window is identically 0) it would suffice to achieve a confidence interval of 272 ms in the time iteration for every station to be able to stay in sync for 24 hours. However, this is clearly impractical; some stations will need to be able to update their local times from stations which are not time standards, and which are perhaps themselves several generations of time iteration removed from a time standard. Table 3 below shows the number of samples needed *at every level* in a hierarchical time distribution tree for the time uncertainty window at the lowest level of the hierarchy to be 272 ms or less, under each of the four cases.

Table 3: Samples Required at Each Level for Hierarchical Time Distribution

Number of Levels	I	II	III	IV
1	2	3	3	10
2	2	3	4	27
3	2	3	4	>50
4	3	3	5	>100
5	3	3	6	>150
6	3	3	7	>200
7	3	3	8	>300

This table contains several items of interest: in case II, three samples are sufficient to achieve a confidence interval of 36 ms, which adds so little to the uncertainty window at each level that three samples are adequate even with a 7-level tree. Also, the 8 ms of uncertainty added by a free-running baud clock doesn't significantly affect the overhead imposed by this synchronization protocol. Clearly, 392 ms of uncertainty in the time of release of transmissions is a serious impediment to the operation described here, while 40 ms appears to be tolerable.

Conclusion

The time iteration protocol described here provides a fairly robust technique for synchronizing stations for linking protection and other purposes. Analysis suggests that this protocol can be used to quickly bring station time bases into synchronization with each other without the need for one or more time standards in every network. This technique is most efficient when radios and protocols are designed to keep time uncertainties well below 100 ms.

Appendix A: Confidence Interval Table

Table A: Selected Values of $2 \bullet t_{1-\frac{\delta}{2}}(k-1)/\sqrt{k}$

k	$\delta = 0.1\%$ (99.9 % C.I.)	$\delta = 1\%$ (99 % C.I.)	$\delta = 5\%$ (95 % C.I.)	$\delta = 10\%$ (90 % C.I.)
2	900	90	17.97	8.93
3	36.49	11.46	4.97	3.37
4	12.92	5.84	3.18	2.35
5	7.70	4.12	2.48	1.91
6	5.61	3.29	2.10	1.65
7	4.50	2.80	1.85	1.47
8	3.82	2.47	1.67	1.34
9	3.36	2.24	1.54	1.24
10	3.02	2.06	1.43	1.16
12	2.56	1.79	1.27	1.04
14	2.26	1.61	1.15	0.95
16	2.04	1.47	1.07	0.88
20	1.74	1.28	0.94	0.77
25	1.50	1.12	0.83	0.68
31	1.31	0.99	0.73	0.61
41	1.11	0.84	0.63	0.53
61	0.89	0.68	0.51	0.43
121	0.61	0.48	0.36	0.30
10^3	0.21	0.16	0.12	0.10
10^4	0.07	0.05	0.04	0.03
10^5	0.021	0.016	0.012	0.010

Appendix B: Variations

We here consider variations on the scenario discussed in the body of the report: in addition to a ±10 ppm time base used to hold sync for 24 hours, we also consider a ±1 ppm time base used to hold sync for a week. With uncertainty due to drift of 1.21 sec after a week, the initial uncertainty must be less than 790 ms to stay within a 2 second PI for one week.

The number of samples required for various heights of time distribution trees is calculated for the following four sizes of confidence intervals: 99.9%, 99%, 95%, and 90%. The table entries are formatted as follows: (# samples for 10 ppm, 24 hours) / (# samples for 1 ppm, 1 week). The four cases considered here are the same as those used in the body of the report.

Table B-1: Samples Required for 99.9% Confidence Interval

Number of Levels	I	II	III	IV
1	3 / 2	3 / 3	4 / 3	15 / 6
2	3 / 3	3 / 3	5 / 4	>40 / 11
3	3 / 3	4 / 3	6 / 4	>100 / 16
4	3 / 3	4 / 3	7 / 4	>150 / 24
5	3 / 3	4 / 3	9 / 5	>200 / ~35
6	3 / 3	4 / 3	10 / 5	>300 / ~45
7	3 / 3	5 / 4	12 / 6	>350 / ~62

Table B-2: Samples Required for 99% Confidence Interval

Number of Levels	I	II	III	IV
1	2 / 2	3 / 2	3 / 3	10 / 4
2	2 / 2	3 / 2	4 / 3	27 / 7
3	2 / 2	3 / 3	4 / 3	>50 / 10
4	3 / 2	3 / 3	5 / 3	>100 / 15
5	3 / 2	3 / 3	6 / 4	>150 / 21
6	3 / 2	3 / 3	7 / 4	>200 / 29

7	3 / 2	3 / 3	8 / 4	>300 / 38

Table B-3: Samples Required for 95% Confidence Interval

Number of Levels	I	II	III	IV
1	2 / 2	2 / 2	2 / 2	6 / 3
2	2 / 2	2 / 2	3 / 2	16 / 5
3	2 / 2	2 / 2	3 / 3	~33 / 7
4	2 / 2	2 / 2	4 / 3	>50 / 9
5	2 / 2	3 / 2	4 / 3	>75 / 13
6	2 / 2	3 / 2	5 / 3	>120 / 17
7	2 / 2	3 / 2	5 / 3	>150 / 22

Table B-4: Samples Required for 90% Confidence Interval

Number of Levels	I	II	III	IV
1	2 / 2	2 / 2	2 / 2	5 / 3
2	2 / 2	2 / 2	2 / 2	12 / 4
3	2 / 2	2 / 2	3 / 2	22 / 5
4	2 / 2	2 / 2	3 / 2	41 / 7
5	2 / 2	2 / 2	3 / 2	>60 / 9
6	2 / 2	2 / 2	4 / 2	>80 / 12
7	2 / 2	2 / 2	4 / 3	>120 / 16