TDA Progress Report 42-127 November 15, 1996

The Problem and Solution of a Twice-a-Day Frame Loss in Galileo's Telemetry

M. Belongie and J. Taylor
Communications Systems and Research Section

D. Rogstad and R. Lee
Tracking Systems and Applications Section

On May 23, 1996, two brand new capabilities were used together operationally for the first time. The in-flight Galileo spacecraft operated for the first time its newly loaded science virtual machine software to produce Phase 2 packet telemetry, and the Deep Space Communications Complex (DSCC) Galileo telemetry (DGT) subsystem received and processed this downlink format from the spacecraft [1]. Previously, the spacecraft packet data stream had been simulated in the project testbed at JPL, and the DGT had received the Phase 1 time division multiplex telemetry from the spacecraft. The transmission of a complex new spacecraft data format to a wholly new ground receiving system proceeded smoothly until the unexpected loss (undecodability) of a Reed–Solomon transfer frame that occurred on May 24. On subsequent days, additional frames were lost, in a soon predictable twice-a-day pattern.

This article documents how a team, consisting of individuals from the DGT development organization, the Deep Space Network (DSN) operations and analysis organization at JPL, the Canberra DSCC, the Galileo Flight Project Office and Telecommunications Analysis Unit, and other members from the JPL Telecommunications Science and Engineering Division worked together to discover and correct the cause of the problem by June 6, 1996.

I. Discovery of the Problem

On May 23, 1996, the Galileo spacecraft started transmitting Phase 2 packet telemetry at 32 bits per second. From the start, both the Deep Space Communications Complex (DSCC) Galileo telemetry (DGT) subsystem and the Block V Receiver reported symbol signal-to-noise ratio (SNR) values close to what was predicted. In the first 3 hours after the spacecraft switched to packet data, the Canberra station had processed all the incoming data frames (after the first two, which were expected to be incomplete). However, at 20:40:48 UTC, Earth received time (ERT), the station was unable to decode spacecraft frame number 32. The first analysis suggested this frame had four extra channel symbols somewhere in it, with no apparent cause.

Over the next few days, John McKinney, the Galileo Ground Data System Manager, began updating by e-mail a list of undecodable (lost) data frames. The list went to a wide variety of Project, Deep Space

Network (DSN), and DGT development team members. By May 27, 4 days after the start of the packet downlink, the lost frame list looked as is shown in Table 1.

By inspection, it was apparent that all but one of the 10 outages listed were clustered around either 13:30 or 20:30 at DSS 43. The 13:30 time was about 1-1/2 hours into the DSS-43 pass, at an elevation of about 40 deg and an azimuth of about 90 deg. The 20:30 time was about 2-1/2 hours before the end of the pass, at an elevation of about 45 deg and an azimuth just slightly more than 270 deg. Jim Taylor suggested we start asking what was happening more or less due east or due west of DSS 43 and about midway up the sky (either local or external).

Clearly, undecoded spacecraft frame 117 didn't fit the pattern and, thus, was considered separately as an isolated failure. Soon everyone was referring to the frame losses around 13:30 as the morning losses, the ones around 20:30 as the afternoon losses (based on JPL time), and to the entire phenomenon as the DSS-43 twice-a-day phenomenon.

Table 1. The lost frame list as of May 27, 1996.

Spacecraft frame no.	Station and DGT frame no.	Date and time, ERT UTC	Length offset[a]	Polarity inversion	Concurrent activity
32	DSS 43, no. 34	May 23, 20:40:48	+4	None	Commanding
117	DSS 14, no. 12	May 24, 08:46:05	0	None	Commanding
152	DSS 43, no. 15	May 24, 13:44:44	−4	Even→odd	—
200	DSS 43, no. 63	May 24, 20:34:18	?	Odd→even	Commanding
320	DSS 43, no. 21	May 25, 13:38:14	?	None	—
368	DSS 43, no. 69	May 25, 20:27:48	?	None	—
536	DSS 43, no. 45	May 26, 20:21:18	?	Odd→even	—
656	DSS 43, no. 18	May 27, 13:25:14	?	Odd→	—
657	DSS 43, no. 19	May 27, 13:33:46	?	→Even	

[a] Length offset is the number of extra (+) or missing (−) symbols relative to the expected 65,536 channel symbols in a data frame.

II. Coding Theory Analysis

A little more analysis showed that the morning lost frame was always four symbols short, and the afternoon lost frame four symbols long. To a coding theory person, this suggested an obvious possible explanation: Since the final coding stage is a rate 1/4 convolutional code, if a single bit were inserted or deleted just before that stage, exactly four symbols would be inserted or deleted. Although the spacecraft people said they could not imagine how that could happen, it was considered a possibility by some until May 29, when the morning lost frame (at DSS 43) occurred while DSS 14 was simultaneously tracking, and DSS 14 recovered the frame perfectly. This event clearly eliminated any possible explanations that involved the spacecraft (or at least that involved *only* the spacecraft).

Because Mignon Belongie had written software to try to recover undecoded frames in situations where symbols were inserted or deleted in a frame, and/or an inversion occurred [2], she was asked to see what she could do with the lost frames. Frame 32 was easily corrected and decoded, and in fact, if the right four-symbol block of symbols was deleted, the frame could be decoded with zero Reed–Solomon (R–S) error corrections, indicating that the error was very localized. Unfortunately, this turned out to be an exceptional case, since in subsequent frames the affected region was much longer. To explain how this was determined, a brief description of Galileo's data frames is needed [3]. A frame has 16,384 bits before convolutional encoding and 65,536 channel symbols after. This consists of a 64-bit frame marker and 8 interleaved Reed–Solomon codewords of 255 eight-bit Reed–Solomon symbols. Detection of the frame markers is used to parse the received symbols into frames, which might end up having more or less than the 65,536 transmitted symbols, as they did in the frames under discussion. The eight Reed–Solomon codewords have different error correcting abilities, so that if a region of the frame is wiped out, and the length of that region is in a certain range, it might happen that the stronger codewords will be decodable while the weaker ones are not. (Note that this is not the normal scenario, where errors are expected to randomly occur throughout the frame, not all in one chunk.) However, even the strongest Reed–Solomon codeword cannot be decoded if the frame has the wrong number of channel symbols and/or an inversion occurred during the frame. The software mentioned above attempts to deal with this by estimating where the problem occurred and compensating for it. If even one (i.e., the strongest) Reed–Solomon codeword can be made to decode, then additional refinement can be used to more precisely determine the location and extent of a bad region, which in turn might lead to decoding more of the codewords, if the damage is not too bad. As previously mentioned, the damage was, in fact, usually too bad to decode the weakest codewords, although at least one codeword was recovered in every case, yielding an estimate of the location and duration of the problem.

III. Learning to Predict Frame Losses

As mentioned above, a pattern in the occurrence times of the lost frames was apparent early on. The two remaining steps in formulating a model for when future frame losses would occur were (a) to realize the successive losses were not at constant clock times, but were moving earlier by several minutes each day, and (b) to connect the time offset with the geometry of the DSS-43 station pass.

The tabulated predictions of symbol SNR and other link quantities appear on hourly centers, with associated station elevation and azimuth angles and spacecraft antenna bore site offpoint from Earth. Familiarity with these tabulations immediately led Jim Taylor to suspect the twice-a-day phenomenon was moving about 4 minutes earlier each day, the same rate as the rise and set times at the station. Rise and set are geometrically defined as when the elevation above horizontal is 6 deg.

The afternoon of May 28, Jim Taylor sent e-mail to the investigation team and to Byron Yetter, the Galileo Network Operations Project Engineer (NOPE). He predicted that with no change to the DGT configuration in place through the weekend, there would be frame losses at DSS 43 at 13:20 and 20:10 on May 29, and about 4 minutes earlier than that on May 30. This prediction was possible because the frame loss times tracked the rise and set times very well, with the morning ones almost exactly 3 hours after rise, and the afternoon ones about 3 hours and 10 minutes before set. He added the caveat that the predictions might be off by a minute or so due to the reported frame loss times being on ∼8-1/2-min centers.

In his role as Telecommunications Analyst and Planner on the Galileo Flight Team, Jim Taylor distributed these predictions to the Mission Control Team and to the Sequence Team. The Mission Controller logged the times of frame outages. The Sequence Team used the predicted outage times, expressed in spacecraft event time, to consider replacing valuable science data with less valuable fill frames to span the outages. (Fortunately, the cause of the frame losses was determined and corrected before the project had to command the spacecraft to implement the fill frame workaround.)

Jim used the frame loss times logged by the Mission Controller to refine his predictions of subsequent outages, with the intent of verifying the speculation that the losses were related to geometric events. Figure 1 illustrates his analysis and predictions.

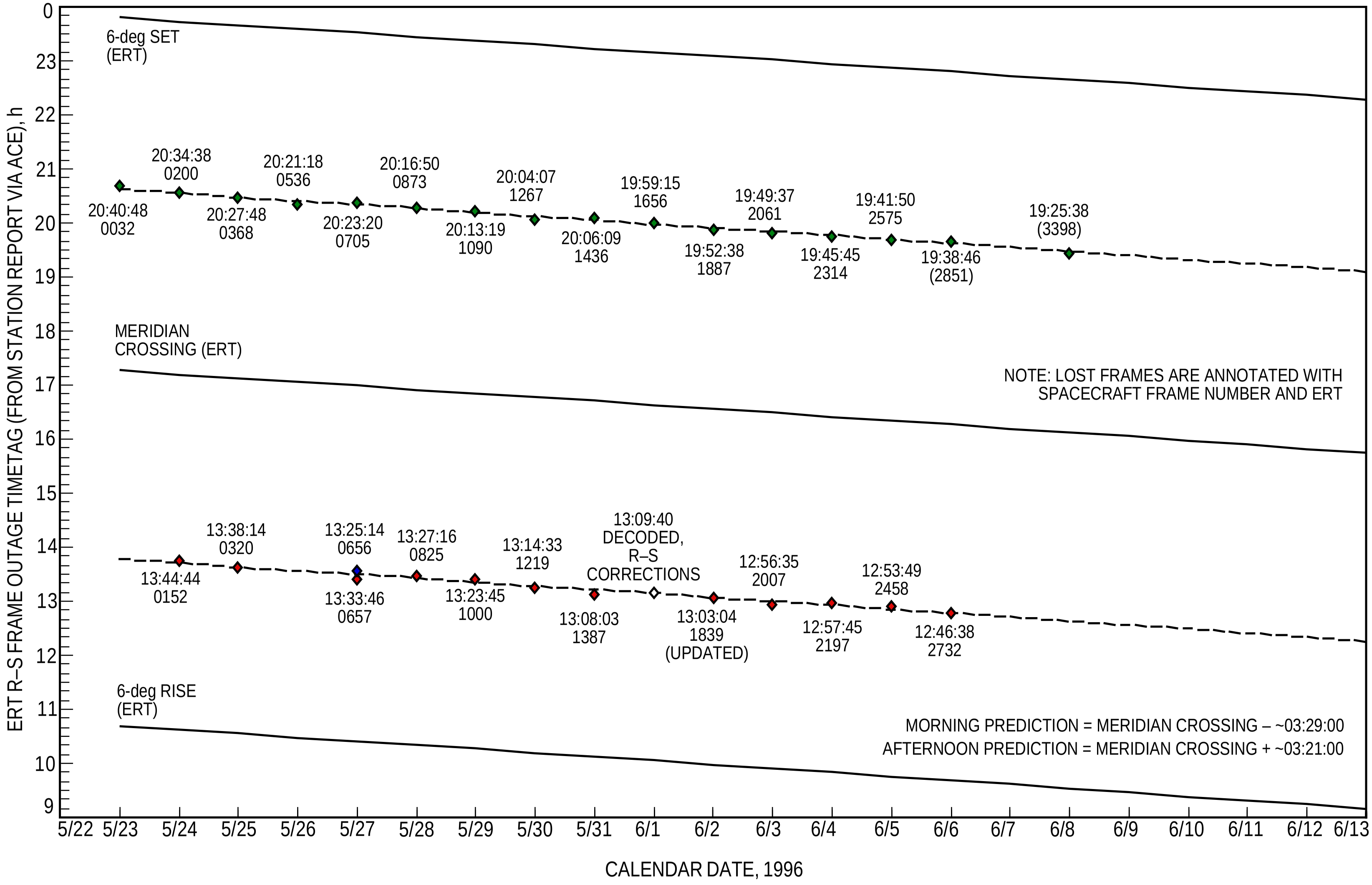

Fig. 1. Times (ERT) of the DSS-43 twice-a-day Reed–Solomon transfer frame losses.

IV. DSCC Galileo Telemetry (DGT) Analysis

From the reported and plotted Block V Receiver (BVR) SNR and buffered telemetry demodulator (BTD) part of the DGT SNR, there was no apparent qualitative difference between these quantities during times of frame losses and any other time during the DSS-43 pass. There were no consistent shifts in mean value, no change in peak-to-peak value, or spikes in either SNR.

The frame losses were occurring only at DSS 43, on the operational DGT. This DGT, called the "40 string" and operated with the so-called Build 4 software, was intended for single-station support. The stations also had a developmental DGT, with the one at DSS 43 called the "40t string" and operated with Build 5 software intended for arrayed-station support.

On May 28, Jeff DeWeese (of the DGT development team) ran a parallel track on the 40t string, using Build 5 software. Both were configured exactly the same, except for the finite impulse response amplitude (FIR AMP) setting, which was set to 1.25 dB, rather than the nominal setting of 6.25 dB, at the suggestion of Jim McComb. The result was that the 40 string had frame losses as predicted, whereas the 40t string did not.

V. Tiger Team Meeting

On May 30, 1 week after the spacecraft sent its first packet data, Joe Statman, Manager of JPL's Communications Ground Systems Section, which was in charge of the DGT implementation team, asked Bill Hurd to set up a tiger team to resolve the twice-a-day phenomenon. On Monday, June 3, Bill Hurd documented the results of the first brainstorming session in an e-mail message, summarized here.

What we know:

(1) The problem is only at DSS 43.

(2) It occurs at an elevation of approximately 43 deg and an azimuth of ±88 deg.

(3) DSS 14 and DSS 63 do not get above 30 deg elevation.

(4) We do not know if the problem occurs ahead of or after the IF distribution assembly, i.e., if it is an RF or signal processing problem.

(5) The signal at DSS 14 is OK when the signal at DSS 43 is bad, so it is probably not a spacecraft problem.

(6) The Reed–Solomon decoder makes errors over approximately 3000 to 4000 symbols.

(7) Data loss occurs at all data rates.

(8) It is always 4 symbols that are added or deleted, in the data checked at 32 bits per second. Ray Piereson (Flight Communications Systems Group Supervisor) will check the recent data at other rates.

(9) The SNR estimate in the BTD is low; the SNR estimate in the full-spectrum recorder (FSR) portion of the DGT is *not* low (in a 10-second averaging time); and the SNR estimate in the BVR may or may not be correlated with the BTD.

(10) The performance is different with the two different FSR/BTD strings at DSS 43 (the operational and array strings). Performance is better with the array string, which has newer software. Performance depends on FIR filter gain and is better with a nonoptimal, lower gain. (Note that this could impact the tracking loop bandwidths, which could impact cycle slipping.)

(11) There are no obvious spikes at the FIR filter output.

(12) There is no obvious radio frequency interference (RFI).

What we can do:

(1) Look at earlier BVR data to see if there was a problem at the same angles.

(2) Get more BVR data, perhaps at two different sets of loop bandwidth settings simultaneously, using two BVRs.

(3) Inject very long baseline interferometry (VLBI) phase calibration tones while tracking Galileo at the critical angles, and see if there are glitches on the tones.

(4) Run test tones at the same angles when Galileo is not being tracked.

(5) See if other users have problems at the same elevation or azimuth.

(6) Turn off the subreflector controller while tracking through the critical angles, so the subreflector does not move. (If there is a pointing or loss-of-RF signal problem, it could be due to a subreflector glitch.)

Not all of these things were done before the mystery was solved, and the ones that were done did not yield any answers. An additional clue that came to light within a few days as Galileo started transmitting at higher data rates was that the number of symbols inserted or deleted varied with the data rate. Specifically, whereas four symbols were always inserted or deleted at a data rate of 32 bits per second, at 40 bits per second, it was five symbols, and at 60 bits per second it was eight. The constant now seemed to be a time interval of approximately 32 msec.

VI. Full-Spectrum Recorder Team Analysis

Prior to the tiger team meeting, individuals on the FSR team had heard various details about the problem and had several discussions to determine what might be wrong with the FSR itself, whether hardware or software, that might cause such a problem. One idea that persisted in the minds of some of the team is that the problem (failure to decode pairs of frames each day) somehow had to do with the delay line that is in the FSR. It appeared from a cursory look at the data that the failed frames occurred at equal times on either side of meridian transit. These are times when the altitude of the antenna is the same, and also the delay line has the same value.

The purpose of the delay line is to delay the signal at each antenna by an amount equal to the light travel time from that antenna to the center of the Earth. Thus, the geometric delay due to the signals arriving at different points on the Earth are removed. It should be noted that the delay line comes *after* the samples have been digitized and timed tagged. In a sense, it works by changing the time tags on the samples by the appropriate amount. Of course, this amount changes during the course of the pass since the Earth is rotating. The amount of the delay is given by the delay predicts, which are interpolated by the software and updated in the hardware every millisecond.

Dave Rogstad attended the tiger team meeting and felt that nothing he heard ruled out the delay lines. He mentioned the idea, but promised to look further into other possible explanations. There was one thing at this meeting that pointed away from the delay lines. This was Mignon Belongie's statement that, except for the first outage, the symbols did not appear to be deleted or inserted all at once, but rather over an interval of thousands of symbols. This led Roger Lee and Robert Kahn to consider models in which the delay or the frequency somehow "jiggled" several times over a frame. Immediately after the meeting, they checked the Network Operations and Control Center (NOCC) Support Subsystem (NSS) frequency and delay predicts files for any “funny” values. They also plotted out the values to see if there was anything unusual at the time of the outage. They also examined the frequencies that were actually applied to the rotators, which were contained in the data records once a second. They found nothing odd in any of these data. However, they did notice that the delays were the same values at the time of the two outages in the track. Another clue that deepened the mystery was that, on one pass, DSS 14 decoded the data while DSS 43 did not.

On June 4th, Dave Rogstad wrote an e-mail suggesting the possibility of gaining some further insight by swapping delay boards between the two FSR strings, mentioning again the occurrence of the problem at symmetric places around local meridian. By this time, Ismael Cruz had a theory about how differences in the delay boards could produce such an effect. Thus, swapping the tone boards made a certain amount of sense. The experiment was tried, and to almost everyone’s surprise and disappointment, it had no effect.

Around this time, Rogstad and Andre Jongeling from the FSR team came up with the idea that maybe the delay lines were somehow going over their maximum limits. They determined that this would introduce a large phase shift, corresponding to 32 msec (the total length of the delay line) at the two symmetric points around meridian. A possible cause of exceeding the delay limits would be if the software somehow miscalculated an offset. In discussing this idea with Roger Lee, it turned out that he had already figured out from data given to him by Mignon Belongie that there was a jump of 32 msec in the data time. He had also determined that the delay offset had a nonzero value (5 microseconds) in the configuration file. (For a time, it was not clear which configuration file was being used at the station.) An obvious way for the delay line to wrap around would be if the offsets were not scaled correctly. Lee called Robert Kahn and asked whether he was *absolutely sure* that he was computing the offsets correctly. In particular, he wanted him to make sure he had all the scale factors correct. Kahn said he was sure, and in fact, he was looking at the code. He read off the relevant lines to Lee, and it all sounded fine. Lee sent out e-mail with some of his conclusions and let it go for the time being.

Bill Hurd called another meeting on June 6th with the FSR team, together with some others, partly because he agreed that the delay line suggestion just seemed right as the most likely source of the problem. Rogstad explained their theory, and Hurd became even more convinced. Rogstad also explained their check of the software, and they agreed that they just needed to continue looking for other ways the delay lines could be corrupted. In the meantime, various board swapping tests continued. Rogstad did not expect any of them to provide the solution because he was convinced that the real answer had to do with the delay line going over its limit, although he was basically stymied at this point.

Roger Lee was also completely perplexed. He went through all the available data and log files. This led him back to looking at the code to see for himself how the offsets were handled. He realized that the code was not what Kahn had read off to him a couple of days before. Puzzled (and worried), he began to search methodically for every line in which the delay was referenced, and noticed one of the form

$$x = a + \frac{b}{k}$$

whereas he thought it should be

$$x = \frac{a + b}{k}$$

He quickly figured out what the effect would be if the offsets were treated as seconds, rather than microseconds (as they were specified). From the predicted delay when the outages occurred, it had long before been computed how much extra delay would be required to wrap around the delay line. It turned out that 5 seconds modulo 32 milliseconds was exactly this number. He called Kahn, who concurred with the analysis. It turned out that when he had read the code a couple of days before, he was mistakenly looking at the array code, not the single-antenna code. They called Byron Yetter to tell him to simply set the offset to zero to fix the problem. They also set all of the frequency offsets to zero as well just to be on the safe side.

VII. Success At Last

On June 7, a triumphant message arrived from Canberra: No frames were lost around the predicted time of 12:43. The problem appeared to have been fixed.

However, there was one last scare on June 9, when a frame was lost at the predicted afternoon time. Ray Piereson carefully checked the specifics of this frame and determined that this frame was probably lost coincident with a data rate change. (Early in Phase 2, a number of frames occurring at or near rate changes were lost. These rate-change frame losses were due to problems unrelated to the twice-a-day phenomenon.)

Most of the frames lost were lost forever. Fortunately, it was only a few percent of the data stream, and at that early stage of Phase 2, the data rate was low and not much high-priority data were being sent. It is not surprising that there were problems resulting from such a drastic change in both onboard and ground-based software, and we can be happy that the losses were not very big.

One remaining puzzle is why a long stretch of symbols was affected instead of just 32 milliseconds worth. Perhaps someone will figure this out some day, but for all practical purposes, the mystery is solved.

References

[1] J. Statman, "Optimizing the Galileo Space Communication Link," *The Telecommunications and Data Acquisition Progress Report 42-116, October–December 1993*, Jet Propulsion Laboratory, Pasadena, California, pp. 114–120, February 15, 1994.

[2] K.-M. Cheung, M. Belongie, and K. Tong, "End-to-End System Consideration of the Galileo Image Compression System," *The Telecommunications and Data Acquisition Progress Report 42-126, April–June 1996*, Jet Propulsion Laboratory, Pasadena, California, pp. 1–11, August 15, 1996.
http://tda.jpl.nasa.gov/tda/progress_report/42-126/126E.pdf

[3] S. Dolinar and M. Belongie, "Enhanced Decoding for the Galileo Low-Gain Antenna Mission: Viterbi Redecoding With Four Decoding Stages," *The Telecommunications and Data Acquisition Progress Report 42-121, January–March 1995*, Jet Propulsion Laboratory, Pasadena, California, pp. 96–109, May 15, 1995.
http://tda.jpl.nasa.gov/tda/progress_report/42-121/121Q.pdf