# Managing the On-Board Data Storage, Acknowledgement and Retransmission System for Spitzer

Marc A. Sarrel*, Carlos Carrion† and Joseph C. Hunt, Jr.‡
*Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, 91109-8099 USA*

**The Spitzer Space Telescope has a two-phase downlink system. Data are transmitted during one telecom session. Then commands are sent during the next session to delete those data that were received and to retransmit those data that were missed. We must build sequences that are as efficient as possible to make the best use of our finite supply of liquid helium. One way to improve efficiency is to use only the minimum time needed during telecom sessions to transmit the predicted volume of data. But, we must also not fill the on-board storage and must allow enough time margin to retransmit missed data. We describe tools and procedures that allow us to build observatory sequences that are single-fault tolerant in this regard and that allow us to recover quickly and safely from anomalies that affect the receipt or acknowledgment of data.**

## I. Introduction

In this paper we develop a simplified model of one segment of the Spitzer Space Telescope downlink system. We discuss strategies, procedures and tools used to mitigate the risk of filling the on-board storage and to recover from anomalies affecting receipt or acknowledgement of data.

Our goal is to tolerate single failures that affect the downlink system without filling the on-board storage and to retransmit any resulting missed data within a short amount of time. We must manage these two observatory resources independently: the amount of on-board storage to hold data, and the amount of time required to transmit data. This management starts during sequence planning and continues through sequence execution.

The Spitzer on-board storage is slightly under-sized for how we use it. The actual science data volumes per observation that we have seen in operations are larger than were predicted before launch. In addition, on-going efficiency improvements have increased the number of observations made. If the on-board storage were to fill, the observatory has a significant chance of entering standby mode or, less likely, safe mode. Soon we will be implementing a flight-software patch to avert filling the on-board storage by replacing observations with idle time if there is not enough room for the predicted volume of data.

Spitzer has a two-phase downlink system. Data are down linked during one telecom pass. Then on the next pass, commands are sent to acknowledge (i.e. delete) those data that were successfully received and to retransmit those data that were missed. This strategy gives us the longest time possible to attempt retransmission of missed data and frees space on board as soon as possible, but makes management and analysis of the system more complex.

We discuss the calculations we use during sequence planning and generation to assess the risk of filling the MMC, and the speed with which we would clear any backlog. These calculations tell us how and where we can mitigate those risks both on board the observatory and on the ground. They also tell us where we may increase efficiency by combining adjacent telecom sessions when the risk is low enough. However, the analysis and tools used for this purpose currently measure and predict only the quantity of data, not the identity of those data. They tell use *how much* data, but not *which* data.

We discuss the software tool that we have developed to analyze the current and future state of the on-board storage. The tool calculates the size, nature and urgency of any backlogs. It guides the choice of actions to recover

* Downlink Systems Engineer, Spitzer Space Telescope, Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Dr., M/S 264-767, Pasadena, CA 91109-8099.
† Mission Operations Systems Engineer, Spitzer Space Telescope, Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Dr., M/S 264-767, Pasadena, CA 91109-8099.
‡ Flight Control Team Lead, Spitzer Space Telescope, Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Dr., M/S 264-767, Pasadena, CA 91109-8099.

from an anomaly during sequence execution. We also discuss the procedures themselves and the failure-mode analysis used to develop them.

We discuss the future development of the tool. Generally, it assumes that the most aggressive approach for acknowledgement and retransmission will be taken. This is not always necessary. In the future versions of the tool, we hope to relax some of these constraints and allow easier analysis of alternative approaches.

We review actual problems that have affected the downlink system in flight. We review the consequences of those problems and the way we have either solved them or minimized their effect.

We examine potential changes to Spitzer flight software that would avert filling the on-board storage. We retrospectively identify some areas of observatory telemetry and telecommand that might have been designed differently to simplify operation of the downlink system. We also examine alternative downlink system designs and how the same sort of analysis described here might be applied.

## II. The Spitzer Downlink System

### A. The Life Cycle of Storage Units

The fundamental component of storage on the Spitzer observatory is the storage unit. Data are written to, read from and deleted from the on-board storage in quanta of storage units. A storage unit is fixed size: sixteen transfer frames. The transfer frames are stored in the on-board storage rather than being built on the fly at the time of transmission. However, there is no header information for storage unit and there is no way to tell on the ground which transfer frames came from which storage units. Storage units exist only on the observatory.

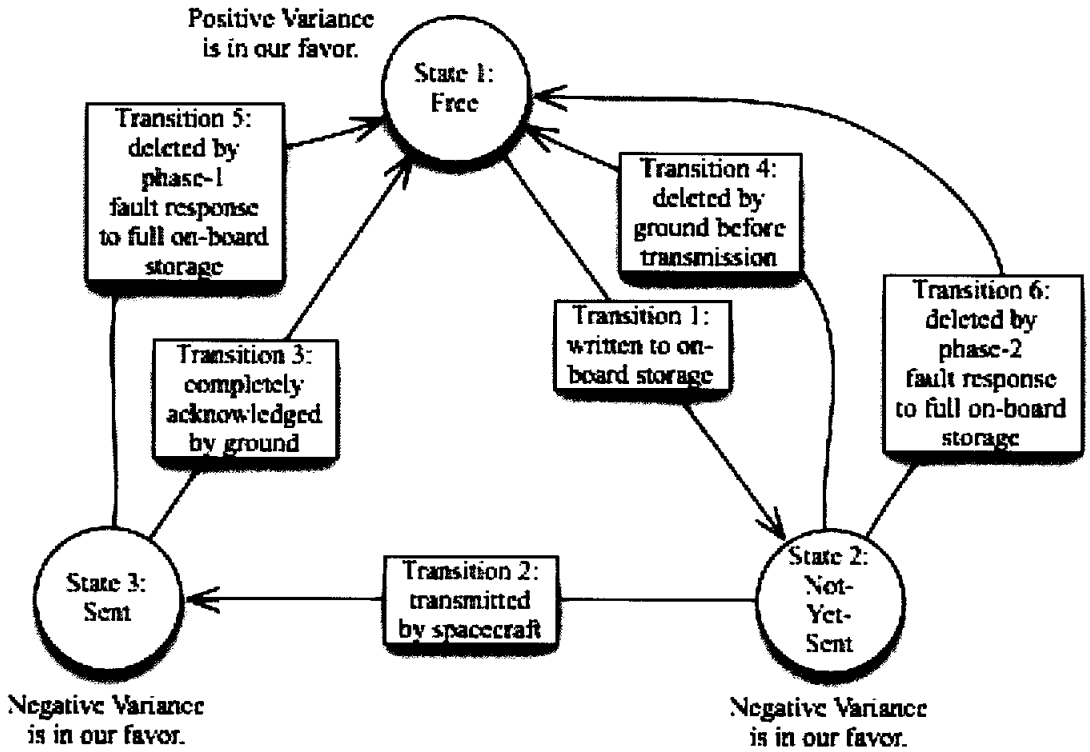


**Figure 1. The state diagram for the management of storage units on the observatory.**

A particular storage unit on the on-board storage is in one of three states: Free, Sent and Not-Yet-Sent (NYS). A storage unit is said to be in the free state if it contains no data. A storage unit is in the not-yet-sent state if data have been written into it, but it has not yet been transmitted. A storage unit is in the sent state if it contains data that have been transmitted by the observatory, but not yet acknowledged (i.e. deleted) by command from the ground. Figure 1 shows the relationship among these three states and the possible transitions between them. The nominal set of three transitions is shown by the three straight arrows in Fig. 1. That is, in the normal case, a free storage unit becomes not-yet-sent when data are written to it (Transition 1). A not-yet-sent storage unit becomes sent when transmitted

for the first time by the observatory (Transition 2). And, finally, a sent storage unit becomes free when the ground sends commands to delete all the data it contains (Transition 3). Transitions 4, 5 and 6 are made only under anomalous conditions.

The sum of the number of storage units marked as free, the number marked as not-yet-sent and the number marked as sent will always equal the size of the on-board storage. The minimum number of storage units in each state is zero. The variance is defined as the number of storage units actually in a given state minus the number predicted to be in that state at a particular time. A *positive* variance of free storage units is in our favor, while a *negative* variance of either not-yet-set or sent storage units is in our favor.

The observatory maintains several queues of storage units. The first queue is just a list of the storage units marked as free. For each APID, there are two queues, one for storage units marked as not-yet-sent, and one for storage units marked as sent. Each storage unit contains packets of only a single application identifier (APID). There is also another queue of storage units, holding those storage units that have been requested by the ground for retransmission.

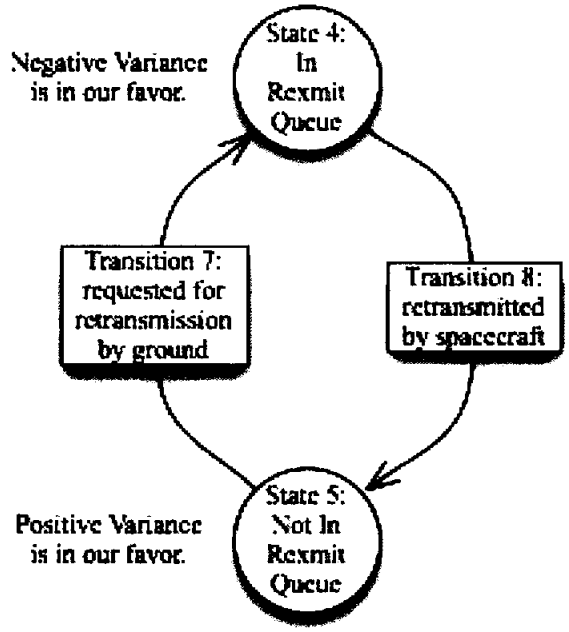


**Figure 2. The state diagram for the retransmission of storage units.**

Figure 2 shows the relationship between these two states of either being in the retransmit queue or not being in the retransmit queue. The retransmit queue may contain packets of all APIDs. The storage units are not copied when they are put in the retransmit queue. They are always in one of the other queues mentioned above as well as being in the retransmit queue. A *negative* variance of storage units in the retransmit queue is in our favor, while a *positive* variance of storage units not in the retransmit queue is in our favor. Storage units are put in the retransmit queue when they are requested for retransmission by the ground (Transition 7). They are removed from the retransmit queue whey they are retransmitted by the observatory (Transition 8). The minimum number of storage units in the retransmit queue is zero. The maximum number is, in effect, unlimited since storage units can be in the retransmit queue multiple times. But, in practice the highest values we see are one third to one half the size of the on-board storage. And, even that many is rare. We don't track the number of storage units not in the retransmit queue, but the minimum would be zero and the maximum is the size of the on-board storage.

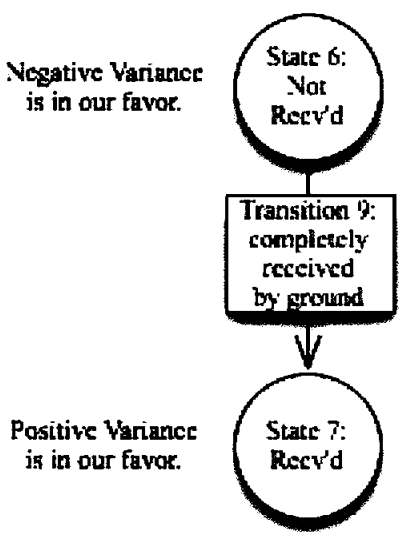


**Figure 3. The state diagram for the receipt of storage units.**

For the purposes of the tool and analysis described in this paper, it is not in general necessary to model the number of storage units in the pairs of queues for each APID. This might be a useful extension in the future.

Finally, we must track how many storage units we have received on the ground. The state diagram in Fig. 3 shows the relationship between storage units that have been received and those that have not. Until a storage unit has been received on the ground it is in the not received state. Once it has been received (Transition 9), it changes to the received state.

The variances are calculated a little differently than in the previous two state diagrams in Fig. 1 and Fig. 2. In those two state diagrams, the observatory sends us telemetry from which we can in theory calculate the number of storage units in each state at any moment in time. But, since it is not possible to uniquely identify a storage unit or transfer frame on the ground or to make a mapping from one on

the ground to one on the observatory, it is not possible to keep a running total of the number of unique storage units received. Although individual packets can be mapped from the ground back to the observatory (via the spacecraft clock and source packet sequence count) since packets are variable size even within an APID, they are not very useful in tracking the amount of storage used on board. So, we compare the number of storage units predicted to be received and actually received for each telecom session individually. We also make a distinction between those storage units received the first time they were transmitted and those received on a subsequent transmission (retransmission).

### B. Measuring Numbers of Storage Units During the Observation and Telecom Cycle

We receive seven principal measurements of the number of storage units from the observatory. They are as shown in Table 1. The first four measurements correspond directly to the number of storage units in states 1, 2, 3 in Fig. 1 and state 4 in Fig. 2. The last three measurements do not correspond to any of the states in the same way. But, they are used later on in the data-volume prediction process to calculate other values. We will focus on the first four measurements for the rest of this section.

| Num | Name | Description |
|---|---|---|
| 1 | Free storage units | The number of storage units currently marked as free. Corresponds to the number of storage units in State 1, Fig. 1. |
| 2 | Not-yet-sent storage units | The number of storage units currently marked as not-yet-sent. Corresponds to the number of storage units in State 2, Fig. 1. |
| 3 | Sent storage units | The number of storage units currently marked as sent. Corresponds to the number of storage units in State 3, Fig. 1. |
| 4 | Storage units in the retransmit queue | The number of storage units currently in the retransmit queue. Corresponds to the number of storage units in State 4, Fig. 2. |
| 5 | Storage units retransmitted | The total number of storage units ever retransmitted since the last reboot. |
| 6 | Storage units discarded by acknowledgement | The total number of storage units ever deleted by commands sent from the ground since the last reboot. |
| 7 | Storage units discarded by de-allocation | The total number of storage units ever deleted by a fault protection response to a full on-board storage since the last reboot. |

**Table 1. The Seven Principal Telemetered Quantities of Storage Units On Board the Observatory.**

The observatory's time is divided between two distinct phases. The first is the period of autonomous operation. During the period of autonomous operation, the observatory is out of contact with the ground and makes science observations. The second phase is the telecom session. During the telecom session, the observatory transmits data to Earth and receives commands from Earth. During these alternating periods, the number of storage units in each of the states from Fig. 1 and Fig. 2 follows regular patterns. Generally, the measurements behave one way during the periods of autonomous operation and another way during the telecom sessions.

During the period of autonomous observation, nominally only one transition from the state diagrams in Fig. 1, Fig. 2 and Fig. 3 is made. A large number of storage units are written to the on-board storage. This corresponds to Transition 1. These are mostly the storage units from science observations, with a small number of engineering and housekeeping storage units. During the telecom session, five transitions are made. A small number of engineering and housekeeping storage units are still written to the on-board storage, Transition 1. Storage units are read from the on-board storage and are transmitted for the first time, Transition 2. The commands are executed to acknowledge storage units received during the previous downlink session, Transition 3. The commands are executed to retransmit any storage units from the previous telecom session that were missed (i.e. not received), Transition 7. Finally, those storage units are retransmitted, Transition 8.

During the period of autonomous observation, the following things happen to the four measurements. The number of free storage units decreases due to Transition 1. The number of not-yet-sent storage units correspondingly increases due to Transition 1. The number of sent storage units remains the same because neither Transition 2 nor Transition 3 is made. The number of storage units in the retransmit queue remains the same because neither Transition 7 nor Transition 8 is made.

During the telecom session, the following things happen to the four measurements. The number of free storage units decreases slightly due to Transition 1, but increases much more due to Transition 3. In the nominal case, there

is always a net increase of free storage units over the telecom session. The number of not-yet-sent storage units increases slightly due to Transition 1, but decreases much more due to Transition 3. In the nominal case, there is always a net decrease of not-yet-sent storage units, ending at zero, over the telecom session. The number of free storage units and the number of not-yet-sent storage units mirror each other during the downlink of data. The number of sent storage units increases because of Transition 2, but decreases due to Transition 3. In the nominal case, there can be either a net gain or a net loss, depending on the relative volumes written to the on-board storage during the previous two periods of autonomous operation. If the most recent period of autonomous operation had a larger data volume than the previous one, then there will be a net increase in the number of sent storage units and vice versa. The number of storage units in the retransmit buffer decreases because of Transition 7 and increases because of Transition 8. In the nominal case, the number of storage units is small and the number of storage units in the retransmit buffer always starts and ends the downlink session with a value of zero.

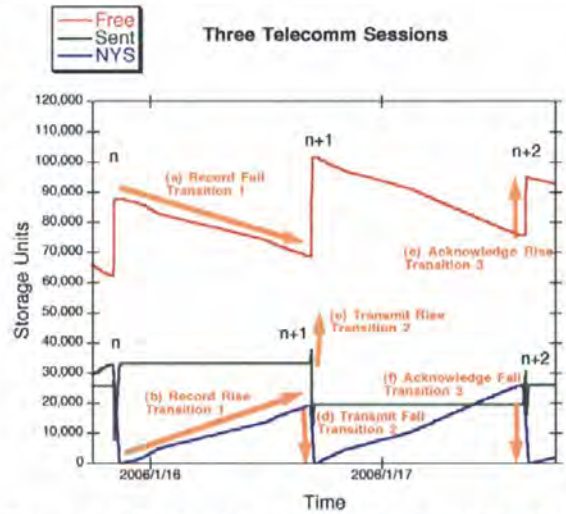


**Figure 4. The Number of Free, Sent and Not-Yet-Sent Storage Units Over Three Telecomm Sessions.**

### C. Visualization of the Observation and Telecom Cycles

We can plot the changes in the number of storage units in each state to see the patterns that develop. The graph in Fig. 4 shows how the number of free storage units, sent storage units and not-yet-sent storage units changes over the course of three telecom sessions. The red line shows the number of free storage units. The green line shows the number of sent storage units and the blue line shows the number of not-yet-sent storage units. The sum of these three values is always equal to the size of the on-board storage, i.e. a little more than 120,000 storage units. The number of storage units in the retransmit queue is not visible on this graph because, in this nominal case, its value when not zero is too small to be visible on the scale of the y-axis.

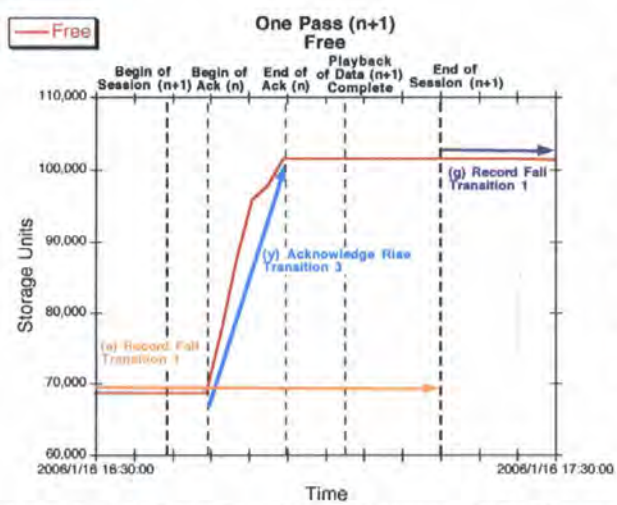


**Figure 5. The Number of Free Storage Units During One Telecom Session.**

The orange arrows show the course of one set of data as it is recorded, transmitted and then acknowledged. During the period of autonomous operations between telecom sessions n and n+1, the orange data are recorded. Transition 1 causes the number of free storage units to fall, Fig.4 arrow (a), and the number of not-yet-sent storage units to rise in mirror image, Fig. 4 arrow (b). During this period that the number of sent storage units remains constant. During telecom session n+1 the orange data are transmitted. Transition 2 causes the number of sent storage units to rise, Fig. 4 arrow (c), and the number of not-yet-sent storage units to fall, Fig. 4 arrow (d). Also during telecom session n+1, the previous data are acknowledged, causing the number of free storage units to rise and the number of sent storage units to fall. The arrows for these transitions are not shown. In this case, there is a net drop in the sent storage units, although just as often there will be a net rise. Finally, during telecom session n+2, the orange data are acknowledged. Transition 3 causes the free storage units to rise, Fig 4. arrow (e), and the sent storage units to fall, Fig. 4 arrow (f). The heights of all six arrows in Fig. 4 are equal to each other.

We can now examine more closely what happens to the number of free, sent and not-yet-sent storage units during one telecom pass. Figure 5 shows what happens to the number of free storage units during telecom session n+1. There are five events shown as vertical dashed lines. From left to right, they are: the beginning of the telecom

session, the beginning of acknowledgement, the end of acknowledgement, the end of playback of data and the end of the telecom session. The time of the beginning and end of the telecom session is controlled by the background sequence on the observatory. The time end of the playback of data is governed by the volume of data on the on-board storage to be transmitted. The times of the beginning and end of acknowledgement depend on when the real-time commands for acknowledgement are sent to the observatory.

Transitions 1 and 3 from Fig. 1 affect the number of free storage units. Transition number 1 happens all the time. Data are always being written to the on-board storage. This is illustrated by arrows (a) and (g) in Fig. 5. We make an artificial boundary at the end of the telecom session to name the data being recorded. In this case, Fig. 5, arrow (a) represents the data recorded from the end of telecom session n to the end of telecom session n+1. Fig. 5, arrow (g) represents the data recorded from the end of telecom session n+1 until the end of telecom session n+2. So, whether it is represented by Fig. 5 arrow (a) or (g), transition 1 is always reducing the number of free storage units.

Figure 5, arrow (y) represents transition 3, the acknowledgement of storage units. It occurs between the beginning and end of acknowledgement. Simultaneously during this time, transition 1 is causing the number of free storage units to fall and transition 3 is causing the number of free storage units to rise.

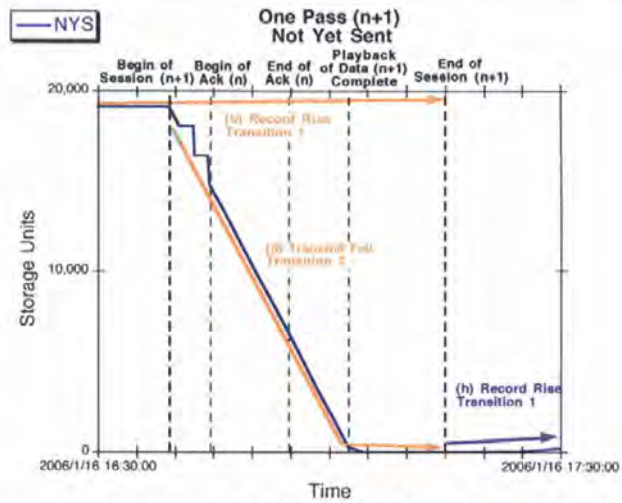


**Figure 6. The Number of Not-Yet-Sent Storage Units During One Telecom Session.**

Figure 6 shows what happens to the number of not-yet-sent storage units during telecom session n+1. As before, transition 1 occurs all the time. In Fig. 6, it is represented by arrows (b) and (h). Transition 1 causes the number of not-yet-sent storage units to rise. From the beginning of the telecom session until the end, transition 2, represented by Fig. 6, arrow (d), causes the number of not-yet-sent storage units to fall. They fall steeply from the start of the telecom session until the the playback of data is complete. Then, they fall very slowly until the end of the telecom session. As transition 1 causes a few storage units to be written to the on-board storage, transition 2 immediately causes them to be transmitted.

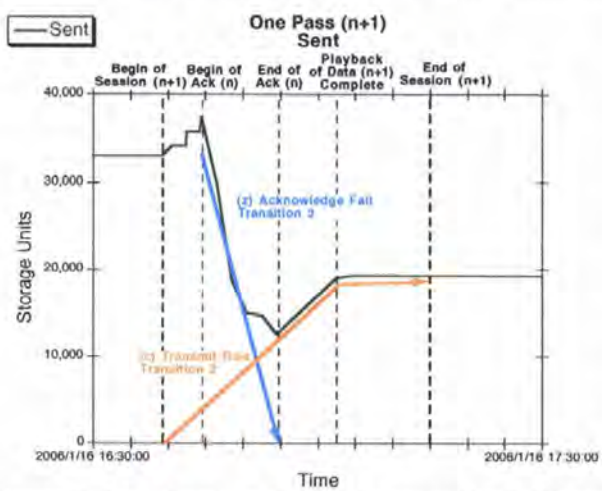


**Figure 7. The Number of Sent Storage Units During One Telecom Session.**

Figure 7 shows what happens to the number of sent storage units during telecom pass n+1. The number of sent storage units changes only during the telecom session itself. It does not change before the beginning or after the end of the telecom session. During the entire telecom session, transition 2, represented by Fig. 7 arrow (c), causes the number of sent storage units to rise. As with arrow (d) in Fig. 6, the rate of transition 2, Fig. 7, arrow (c), decreases drastically once the playback of data is complete. Between the beginning of acknowledgement and the end of acknowledgement, transition 3, Fig. 7, arrow (y) causes the number of sent storage units to fall. The interaction of transitions 2 and 3 give the curve in Fig. 7 its characteristic "s" shape. In this case, there was a net fall in the number of sent storage units from the beginning of the telecom session to the end. This is because a greater number of storage units were acknowledged than transmitted during telecom session n+1. But, just as often, the curve will exhibit a net increase.

## III. Managing Downlink Risk and Efficiency During Sequence Construction

The main purpose of this analysis is to prevent the on-board storage from filling. During sequence construction, we can manage that risk by means of a few simple calculations based on predicted data volumes. Our goal is to manage data volumes in such a way that the sequences we build are single-fault-tolerant. That is, we must withstand the loss of an uplink, loss of a downlink or loss of an entire telecom session and not fill the on-board storage. We must also leave enough margin in the amount of downlink time to insure that, if we must retransmit a large volume of data, we can do so quickly. Until that backlog is cleared, the consequences of a second failure are increased and we are at greater risk of filling the on-board storage. Our guideline is that we must clear the backlog within six telecom sessions after a failure.

| | Start of Telecom Session | | End of Telecom Session | |
|---|---|---|---|---|
| **Session** | **Not-Yet-Sent** | **Sent** | **Not-Yet-Sent** | **Sent** |
| n | Set n | Set n-1 | - | Set n |
| n+1 | Set n+1 | Set n | - | Set n+1 |
| n+2 | Set n+2 | Set n+1 | - | Set n+2 |

**Table 2. Transmission and deletion of data in the nominal case over three telecom sessions.**

### A. Consequences of Single Failures

A single failure is defined as the failure to receive the data that were transmitted by the spacecraft during a telecom session, the failure to uplink and execute the real-time commands to acknowledge the data received during the previous downlink session, or both simultaneously. Normally, at the beginning of a telecom session, e.g. session n, there are two sets of data stored on the on-board storage. Set n-1 is marked as sent and set n is marked as not-yet-sent. At the end of a downlink session, there is normally only one set of data stored on the on-board storage. Set n is marked as sent. Then, that pattern continues for telecom sessions n+1, n+2, etc. See Table 2.

| | Start of Telecom Session | | End of Telecom Session | |
|---|---|---|---|---|
| **Session** | **Not-Yet-Sent** | **Sent** | **Not-Yet-Sent** | **Sent** |
| n | Set n | Set n-1 | - | Set n & Set n-1 |
| n+1 | Set n+1 | Set n & Set n-1 | - | Set n+1 |
| n+2 | Set n+2 | Set n+1 | - | Set n+2 |

**Table 3. Transmission and deletion of data in the case of a missed uplink during session n.**

But, if during session n, we are unable to send the real-time command to acknowledge set n-1, i.e. transition 3 in Fig. 1 does not happen, then we will end session n with both set n-1 and set n marked as sent. Then, at the start of telecom session n+1, we will have three sets of data on the on-board storage. Set n-1 and set n will be marked as sent and set n+1 will be marked as not-yet-sent. If all goes well during session n+2, we will send real-time commands to acknowledge both sets n-1 and n and at the end of session n+2 the system will have returned to normal. See Table 3.

| | Start of Telecom Session | | End of Telecom Session | |
|---|---|---|---|---|
| **Session** | **Not-Yet-Sent** | **Sent** | **Not-Yet-Sent** | **Sent** |
| n | Set n | Set n-1 | - | Set n |
| n+1 | Set n+1 | Set n | - | Set n+1 & Set n |
| n+2 | Set n+2 | Set n+1 & Set n | Set n+2 | Set n+1 & Set n |

**Table 4. Transmission and deletion of data in the case of a missed downlink during session n.**

If we are unable to receive the data transmitted by the observatory during telecom session n, i.e. transition 9 in Fig. 3 does not happen, then we will still end session n with one set of data on the on-board storage. Set n is marked as sent. But, since we received no data from the observatory during telecom session n, we will be unable to build a real-time command to delete any data to be sent during telecom session n+1. Therefore, we will end telecom session n+1 with two sets of data on the on-board storage. Sets n and n+1 will be marked as sent. Then we will begin telecom session n+2 with three sets of data. Sets n and n+1 will be marked as sent and set n+2 will be marked as not-yet-sent. Set n+1 will need to be retransmitted. That retransmission creates a backlog of data which must be cleared over the succeeding telecom sessions before we will have returned to normal. See Table 4.

In this case, the states at the ends of telecom sessions n+1 and n+2, and the beginnings of n+2 and n+3, are somewhat idealized. At the end of session n+1 for example, it is possible that some of set n+1 will remain not-yet-sent. Similarly, at the end of session n+2, some of set n+2 may have been transmitted and moved to sent. The exact results will depend on the relative data volumes of sets n, n+1 and n+2 and the relative durations of sessions n+1 and n+2. However, the important point remains true that, regardless of whether the data are marked as sent or not-yet-sent the volume of that data on the on-board storage is as shown at the start and ends of the telecom sessions. The same holds true in the next example, the case when we miss both an uplink and downlink at the same time.

If we miss an entire telecom session, i.e. if we miss both the uplink and the downlink during session n, the individual results described above will be combined. We will begin both telecom sessions n+1 and n+2 with three sets of data on the on-board storage. At the start of session n+1, set n-1 and set n will be marked as sent and set n+1 will be marked as not-yet-sent. Then, at the start of session n+2, sets n and n+1 will be marked as sent and set n+2 will be marked as not-yet-sent. Just as in the missed-downlink example above, the data from set n will need to be retransmitted. That retransmission creates a backlog of data which must be cleared over the succeeding telecom sessions before we will have returned to normal. See Table 5.

| | Start of Telecom Session | | End of Telecom Session | |
|---|---|---|---|---|
| **Session** | **Not-Yet-Sent** | **Sent** | **Not-Yet-Sent** | **Sent** |
| n | Set n | Set n-1 | - | Set n & Set n-1 |
| n+1 | Set n+1 | Set n & Set n-1 | - | Set n+1 & Set n |
| n+2 | Set n+2 | Set n+1 & Set n | Set n+2 | Set n+1 & Set n |

**Table 5. Transmission and deletion of data in the case of a missed uplink and a missed downlink during session n.**

The above examples illustrate the equivalence of missing an uplink and missing a downlink. Missing a downlink on pass n is equivalent to missing an uplink on pass n+1. They have identical effects in that they cause three sets of data to be on the on-board storage at the start of session n+2. They have different effects in that if the original failure was a missed uplink, then no data must be retransmitted and the recovery will be complete at the end of telecom session n+1. If the original failure was a missed downlink, then that set of data must be retransmitted and the recovery will be strung out over several succeeding passes.

Exactly how many sessions it will take to clear the backlog of data will depend on the volumes of the sets of data and the lengths of the telecom sessions. Each telecom session is nominally designed to be slightly longer than needed to transmit that set of data. Typically, we keep each downlink session no more than ninety-seven or ninety-eight percent full. Sometimes they are significantly less full because of other rules governing the minimum length of a telecom session. So, this extra time can be used to retransmit data that were not received the first time. We also have the capability to start the transmission of data early via real-time command at the beginning of telecom sessions. After the observatory comes to Earth-point, it waits seven minutes before starting the transmission of data. It does this to allow the establishment of a good communications link with the ground. However, the majority of the time, the full seven minutes is not needed and so can be used as extra time to clear data backlogs. Typically, we gain six of those seven minutes as extra time.

### B. Analysis During Sequence Construction

We produce one set of data volume predictions for each telecom session. Each set of predictions contains five pieces of information. It contains: 1) the time at which the background sequence on the observatory will begin the downlink session, 2) the time at which it will end the telecom session, 3) the total number of storage units it would be possible to transmit from the on-board storage during this time, 4) the number of storage units that will have been recorded from the end of the previous session to the start of this session, and 5) the number of storage units to be recorded between the start and end of this telecom. The third item is calculated by taking the total transmission rate of the observatory and subtracting off a small portion to account for real-time data that are never recorded to the on-board storage. For the purposes of this early analysis during sequence generation, the fourth and fifth items are usually added together and used as a single number.

From the previous section, we can see that the most important requirement for single-fault tolerance is that the volumes of data from each set of three consecutive sessions must be smaller than the size of the on-board storage. So, the first step in the analysis is calculate a rolling sum of each set of three consecutive data volumes and subtract it from the size of the on-board storage. This quantity is called the missed-pass risk. It says how many free storage units would remain at the start of a given pass in case of a single failure. We could instead have limited each downlink session to one-third of the size of the on-board storage, after allowing some margin. That strategy would have limited us to about 37,000 storage units per telecom session. But this would be unnecessary, would provide no reduction in the complexity of calculation compared to our current system and would impose a severe inflexibility in scheduling science observations.

In Table 6, the rows below the dashed line, rows n through n+12, represent the telecom sessions for a single background sequence. The rows above the dashed line, n-6 through n-1, come from the previous sequence. They are included because data from the previous week are needed to do the calculations for the current week and vice versa. Columns A, B and C are supplied as part of the data volume predictions. Column A is shown in place of the beginning and end times of the telecom session, items 1 and 2 above. For our purposes, the exact times are not important, so we have simply numbered the telecom sessions. Column B is the predicted capacity of the telecom session, or the total number of storage units it would be possible to transmit from the on-board storage during the

length of that telecom session, item 3 above. Column C is the total volume of data predicted to be recorded and transmitted from then end of the previous telecom session to the end of the current one, the sum of items 4 and 5 above.

Column D is the first calculated column. It simply shows what percentage of the predicted capacity will be used by the predicted volume. It is just column C divided by column B expressed as a percentage. We use this as our first guideline and try to keep it below ninety-eight percent. This serves several purposes. It allows for some under prediction in the predicted volumes, column C, due to uncertainty in how well the data will compress on board the observatory. Although, we are somewhat conservative in this regard. It also allows us to make sure that we leave some margin in the telecom session to retransmit data if needed. Columns F and G do this calculation more directly. Finally, if a downlink session is more than one-hundred percent full, the simplified calculation for column E gives incorrect results.

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| **Telecom Session** | **Predicted Capacity** | **Predicted Volume** | **Percentage Full** | **Missed-Pass Risk** | **Six-Pass Margin** | **Six-Pass Percentage** |
| | **(SU)** | **(SU)** | **(Percent)** | **(SU)** | **(SU)** | **(Percent)** |
| n-6 | 34,718 | 23,232 | 66.92% | | 47,395 | 80.23% |
| n-5 | 33,562 | 26,249 | 78.21% | | 38,360 | 83.80% |
| n-4 | 39,570 | 38,943 | 98.42% | | 25,531 | 88.46% |
| n-3 | 34,781 | 27,905 | 80.23% | | 30,648 | 86.26% |
| n-2 | 34,072 | 21,052 | 61.79% | | 24,887 | 89.05% |
| n-1 | 37,343 | 26,224 | 70.22% | | 15,556 | 93.30% |
| n | 31,398 | 28,705 | 91.42% | 44,894 | 10,657 | 95.61% |
| n+1 | 30,595 | 29,301 | 95.77% | 36,645 | 8,809 | 96.44% |
| n+2 | 24,000 | 23,508 | 97.95% | 39,360 | 19,632 | 92.63% |
| n+3 | 36,714 | 35,759 | 97.40% | 32,307 | 8,774 | 96.78% |
| n+4 | 38,346 | 37,941 | 98.94% | 23,667 | 10,328 | 96.07% |
| n+5 | 42,017 | 35,056 | 83.43% | 12,120 | 17,755 | 93.17% |
| n+6 | 42,017 | 41,742 | 99.35% | 6,136 | 14,748 | 94.29% |
| n+7 | 35,409 | 35,368 | 99.88% | 8,709 | 21,081 | 90.54% |
| n+8 | 42,833 | 37,311 | 87.11% | 6,454 | 13,616 | 92.43% |
| n+9 | 42,833 | 40,485 | 94.52% | 7,711 | 8,094 | 94.10% |
| n+10 | 28,556 | 24,414 | 85.50% | 18,665 | 20,023 | 81.56% |
| n+11 | 39,306 | 27,803 | 70.74% | 28,173 | 5,131 | 92.59% |
| n+12 | 40,304 | 36,350 | 90.19% | 32,308 | -7,370 | 125.43% |

**Table 6. Example calculation of the various measures of downlink risk for a single sequence.**

Column E is the missed-pass risk number. It shows the number of free storage units that would remain if there were a single failure in the downlink system. It is just the sum of the three previous predicted volumes subtracted from the size of the on-board storage. So, column E for telecom session n is the sum of column C for sessions n, n-1 and n-2 subtracted from 120,875. Two telecom sessions from the previous sequence are needed to calculate all of column E for the current sequence. None of the telecom sessions from the next sequence are needed, though. Therefore, no part of column E must be re-analyzed, as must columns F and G, see below.

If any session has a value in column D that exceeds one-hundred percent full, then the affected numbers in column E will be incorrect. In that case, full modeling and prediction of telemetry values, see Table 1, would be needed. The basic problem is that any new storage units that remain untransmitted at the end of a telecom session must be counted twice, once for the current session on once for the next. This can create a bow-wave effect that may ripple forward across several telecom sessions. In any case, this effect causes the numbers shown in column E to be higher than the true values.

Column F shows how much extra capacity, in storage units, would remain if the data from this telecom session needed to be retransmitted over the following six sessions. This calculation assumes that each of the following six downlink sessions can be lengthened in real-time by six minutes, i.e. about 4,830 storage units, as described in section III A. So, column F for telecom session n is the sum of column C for sessions n through n+6 subtracted from the sum of column B for sessions n+1 through n+6, each lengthened by six minutes.

We will never have complete information to calculate column F for the last six telecom passes in a given sequence when the predicts for that sequence first become available because we need the predicts from the next sequence. In this example, column F for sessions n+7 through n+12 are calculated based on incomplete information.

We make the worst-case assumption that all sessions from the following sequence are exactly one-hundred percent full, but that we are still able to extend each session by six minutes. This makes column F for the last few sessions look artificially bad. So, we always re-analyze the last six sessions of the previous week to make sure that they are acceptable now that complete information to calculate them is available. That is why we include sessions n-6 through n-1 in this table.

Column G is a similar measure to column F, except expressed as a percentage full similar to column D. It shows the percentage full that the following six sessions would be if the data from the current session needed to be retransmitted. Like column F, this calculation assumes that the following six passes will each be lengthened in real-time by six minutes. So, column F for telecom session n is the sum of column C for sessions n through n+6 divided by the sum of column B for sessions n+1 through n+6, each lengthened by six minutes, expressed as a percentage. We make the same worst-case assumption for column G as we do for column F. In addition, we must re-analyze column G for the last six sessions of each sequence when the predictions for the following sequence become available. Fortunately, because we build our sequences far enough in advance and work on several sequences at once, we can still correct problems in the last six sessions of columns F and G once the predicts for the next sequence become available.

### C. Definition of Allowable Risk

Our criteria for what constitutes allowable risk are empirical, and they have changed over time to reflect changes in the quality of the data-volume predictions. The limits are guidelines. Violation of them requires analysis that is more detailed, mitigation or a waiver.

The definition for column D, the percentage full, originally stood at 93%. That allowed us to predict only the data volume for the science data in column C, but to omit the volume for the routine engineering data. Most of the 7% margin was budgeted to account for that engineering data. However, when the other calculations of risk were introduced, columns E, F and G, this implicit accounting of engineering data volume caused the numbers in those columns to be incorrect.

Since we have begun the explicit inclusion of predicted volumes for engineering data volume, not only have the values in columns E, F and G become more reliable, but we have been able to increase the limit for column D to 97%. Pending some further operation at the 97% limit, we anticipate and increase to 98%. We aim to have the only source of uncertainty in the predicted volume be that of on-board compression. Having said that, we do systematically over-predict the science data volumes slightly to be conservative. If we were to make the mean of the prediction equal the mean of the actual volumes, then we would probably have to lower the number. The same goes for the other risk columns, E, F and G.

The prediction of the capacity of a telecom session is very accurate. All that is needed is to determine the duration of the session and multiply by the rate at which storage units are read from the on-board storage and transmitted. The major source of uncertainty is the duration of the pre-session slew back to Earth point. That can affect the duration by a few percentage points.

For the missed-pass risk column, we set a yellow-alarm limit at 10,000 storage units and a red-alarm limit at 5,000 storage units. If we are unable to raise the number above these limits, then we must take some other form of mitigation as described below. We take different actions depending on whether we violate the yellow limit or the red limit.

Columns F and G are more recent additions. As such, we do not have well defined limits for them. Right now for column G, we are using the same 97% limit as for column D. That seems to work well and it may be possible to change from six passes to five and still meet this 97% requirement. We do not currently have a separate limit for column F, although we do use the number as auxiliary information.

The above limits are meant to mitigate risk. We do not have explicit limits to improve efficiency. However, we do have some informal guidelines. These will be described in the next section.

### D. Mitigating Risk and Increasing Efficiency During Sequence Construction

We can use the numbers in columns D through G of Table 6 to analyze where we are at too great a risk of single failures. Remember that our two requirements are that we not fill the on-board storage and be able to retransmit any single session's data within the next six sessions. Once we have identified the areas of risk, we can take specific steps to mitigate that risk. Those steps fall into two categories.

In the first category, we can increase or decrease the values in columns B and C by changing the background sequence. If that fails to completely mitigate the risk, we can take steps in the second category like getting backup ground antennas and bypassing the normal two-phase acknowledgement cycle and deleting data from the observatory during the telecom session in which it is transmitted. Conversely, this same analysis can be used to

identify where there is too much margin and risk can be increased. This allows us more time to make science observations. The gains can take the form of simply adding more science observations to replace idle time, or shortening or removing telecom sessions to allow more science observations. In the following discussion, to decrease risk means to lower the numeric values in columns D and G, and to increase the numeric values in columns E and F. To increase risk means the values change in the opposite directions.

The first way to mitigate risk or increase efficiency is to manipulate the values in columns B and C. That is, either lengthen or shorten the telecom session, column B, or add or remove data volumes, column C. The exact methods used to accomplish these changes are limited by the constraints that the scientists place in the specification of their observations. Optimizing the selection and placement of science observations is a subject for a different paper. Therefore, we will focus on the effects to risk and efficiency of making those changes. In the following examples, we will primarily make changes to telecom session n.

If we increase the predicted volume of data for telecom session n, i.e. increase the numeric value in column C, that will increase the risk in column D for session n, column E for the three sessions n through n+2, and in columns F and G for the seven sessions n-6 through n. Conversely, if we decrease the value of the predicted volume for session n, the risk will decrease in the same places. The affected rows in columns E and F change in value by the same magnitude as the change to column C. If the value in column C increases by 1,000, the values in columns E and F decrease by 1,000.

If we shorten telecom session n, i.e. decrease the numeric value in column B, which will increase the risk in column D for session n, and in columns F and G for the six sessions n-6 through n-1. Conversely, if we increase the numeric value in column B for session n, the risk will decrease in the same places. As before, the magnitude of the change in column F is equal to the magnitude of change in column B. Notice also that changes to column B do not affect the risk at all in column E. This is true because changes to the length of the telecom session do not affect the amount of data stored on the on-board storage. However, this rule fails if the length of the telecom session is shortened to the extent that the percentage full, column D, exceeds one-hundred percent. In that case, the values in column E are no longer valid. To get correct values, we would need to do full modeling and prediction of the telemetry values in Table 1.

Using a combination of the above direct and inverse analyses, we can easily identify for which telecom session to change the values of the predicted capacity and volume in order to reduce the risk in columns D through G. For example, if the missed-pass risk in session n+2, column E is too low, it may be mitigated by decreasing the predicted volume, column C, in any of sessions n through n+2. If the six-pass percentage in session n is too high, then it can be reduced by either decreasing the values in the predicted volume column B for any or all of the seven sessions n through n+6, or by increasing the predicted capacity, column B for any or all of the six sessions n+1 through n+6.

There are also several combinations of changes that can be made to reduce, or at least balance risk. Shifting storage units from one telecom session to a neighboring one is a common example. It can be easily accomplished by shifting the time of a telecom session and moving an observation from one side to the other. If we move five-hundred storage units from session n+1 to session n. That will increase the risk in column D and E for session n, and columns F and G for column n-6. It will also decrease the risk for columns D, F and G in session n and column E in session n+3. Column E for sessions n+1 and n+2 and columns F and G for sessions n-5 through n remain unchanged. Or, more precisely, the changes cancel each other out.

However, we can also use this analysis to increase efficiency. For example, if the missed-pass risk is around than 60,000 or 70,000 for a couple of sessions in a row, then it is likely that one of the sessions can be removed and the time used for science observations. And, even the data volumes would allow removal of a telecom session, their may be other flight rules and procedures that require us to keep it. Choosing exactly which session to forgo can be tricky and usually requires some trial and error. As a second example, if the percentage full column is below around 60% or 70%, it may be possible to shorten the telecom sessions and use the additional time for science observations. However, we have other operational rules that govern the minimum length of telecom sessions, so it is not always possible to do this. In any case, the new sequence would still have to meet all the risk-mitigation limits described above.

If we cannot lower the risk sufficiently using the methods from the first category, then we can take measures from the second category. Our rules in this second category are primarily aimed at mitigating risks in the missed-pass risk column. Methods in the second category are not capable of mitigating risks in the percentage full column. In addition, we have not yet developed methods in the second category to mitigate risks in the six-pass margin and six-pass percentage columns. They were added recently and we need to gain more experience with them first.

We have one set of methods to use when a value in the missed-pass risk column is below the yellow-alarm limit and another set to use when it is below the red-alarm limit. The strategy is that below the yellow-alarm limit we will

provide a redundant uplink capability to make sure that we can send the commands to acknowledge and retransmit the data from the previous telecom session. When we are below the red-alarm limit, in addition to providing a redundant uplink capability, we also provide a redundant downlink capability. The logic behind this order is that missing an uplink can put us in danger of filling the on-board storage on the next pass, while missing a downlink can put us at risk of filling the on-board storage on the second pass after.

a)

| | Missed-Pass Risk (SU) | Mitigation |
|---|---|---|
| n-2 | no alarm | |
| n-1 | no alarm | ground antenna |
| n | Yellow | |
| n+1 | no alarm | |
| n+2 | no alarm | |

b)

| | Missed-Pass Risk (SU) | Mitigation |
|---|---|---|
| n-2 | no alarm | pre-emptive delete |
| n-1 | no alarm | |
| n | Yellow | |
| n+1 | no alarm | |
| n+2 | no alarm | |

c)

| | Missed-Pass Risk (SU) | Mitigation |
|---|---|---|
| n-2 | no alarm | ground antenna |
| n-1 | no alarm | ground antenna |
| n | Red | |
| n+1 | no alarm | |
| n+2 | no alarm | |

d)

| | Missed-Pass Risk (SU) | Mitigation |
|---|---|---|
| n-2 | no alarm | pre-emptive delete & ground antenna |
| n-1 | no alarm | |
| n | Red | |
| n+1 | no alarm | |
| n+2 | no alarm | |

**Table 7. Options to mitigate single yellow and red alarms in the missed-pass risk column.**

The next question is during which telecom session to provide the backup capabilities. See Table 7. If there is a yellow alarm in the missed-pass risk column in telecom session n, we must provide a redundant backup capability during telecom session n-1. This is because a missed uplink will cause the number of free storage units to reach the missed-pass risk number on the next pass. See Table 3. If there is a red alarm in session n, in addition to a backup uplink in session n-1, we must provide a redundant downlink capability in session n-2. This is because a missed downlink will cause the number of free storage units to reach the missed-pass risk number on the second pass after. See Table 4. The way that we provide a redundant uplink is to schedule a second ground antenna in session n-1. Then, if the primary antenna cannot transmit commands, we can switch to the backup. Similarly, to provide redundant downlink, we schedule a second ground antenna in session n-2. If we have several alarms that we must mitigate, a single backup antenna can serve several roles. A backup station during session n provides a redundant uplink for session n+1 and a redundant downlink for session n+2.

At this point in our mission, we must use a 70 m antenna to support our downlink rate of 2.2 Mb/s. However, it is generally not possible for us to get a second 70 m antennas as a backup. Similarly, even though we have the capability to array two or more 34 m antennas as our primary, it is generally not possible to get more than a single antenna as a backup. To make the uplink redundant, however, we do not need a 70 m antenna. We can use the smaller 34 m antennas. For redundant downlink, we can use the 34 m antennas at a lower data rate to give graceful degradation. Combining this with lengthening the downlink session with a real-time command as described before, we can usually mitigate the risk sufficiently. If we were to receive no data during a missed downlink in session n-2, the free storage units on the observatory would hit the value in the missed-pass risk column at the start of session n. However, we can delete in session n-1 any data we receive during session n-2 and thereby increase the number of free storage at the start of session n by that amount. We have done the telecommunications analysis to determine what downlink bit rates we can support as a function of time and have the real-time commands and procedures in place to drop the downlink bit rate when needed.

If we have a yellow alarm and a backup ground antenna is not available, we do have an alternate method to mitigate the risk. We mitigate the risk not by providing redundancy, but by bypassing the two-phase transmission and acknowledgement system. We do this by deleting data during the same pass in which it was transmitted. To mitigate against a yellow alarm in the missed-pass risk column in session n, we must do a pre-emptive delete in session n-2. This immediately increases the number of free storage units on the observatory. Then, if we miss the

uplink in session n-1, the number of free storage units at the start of session n will be increased by that amount. We need not pre-emptively delete the entire set of data. We need only delete a sufficient amount to raise the low point of the free storage units above the yellow limit at the start of session n.

Our ground data system is not currently capable enough to send us at JPL the large volume of data before the telecom session has ended. So, we are unable to run our normal data-acknowledgement software to generate the deletion commands. We must build a pre-emptive delete command in advance of sequence execution It must delete only a sufficient volume of data to mitigate the risk. Then, we must use our knowledge of the order and rate at which the observatory transmits data to predict the time when it is safe to send the pre-emptive delete command. We predict indications in telemetry that tell us when the data have been transmitted. In addition, we monitor the performance of the ground antenna during the telecom session to see if it is performing well and therefore likely to have received a good copy of the data. Ultimately, though, when we send the pre-emptive delete command, we do so before we have the data ourselves at JPL and trust that a complete copy is saved upstream in the ground data system.

We are hopeful that, in the near future, planned improvements to the network capacity of the ground data system will allow us to receive some data during the telecom session, build a delete command with our normal software during the session and then send it. That would have the advantage of only deleting data that we know for sure we have received.

### E. Benefits of Missed-Pass Risk Analysis

The primary benefit of this of analysis is an operational one. It lets us design our sequences to be single-fault tolerant to problems with the downlink system. It also lets us determine how and when to mitigate risk if we are unable to maintain single-fault tolerance. However, we have used it in the development of and process improvement of our mission operations system. Here are a few case studies.

Spitzer has three different science instruments. The first, the Spitzer Infrared Spectrograph (IRS), produces data at a relatively low rate. The second, the Spitzer Infrared Array Camera (IRAC), varies in its rate of production. Depending on how it is used, it can produce data at a low or high rate. The third, the Spitzer Multi-band Imaging Photometer produces data at a high rate. Only one instrument is powered on at a time. Each instrument remains on for a campaign that lasts from about one week to as many as three weeks. For the first part of science operations, we would have two telecom sessions per day spaced approximately twelve hours apart. Each session was up to one hour long.

The science teams asked if it would be possible to reduce the number of telecom sessions per day. The IRS would typically produce 15,000 storage units per session space at twelve-hour intervals. It only takes about nineteen minutes for the observatory to transmit this volume of data. This was well below our minimum length of forty minutes during prime shift and thirty minutes otherwise for a telecom session. In addition, the observatory had to spend a certain amount of time slewing to and from Earth point for each session. The observatory takes about 15 minutes to slew a full 180°. Therefore, we were not making efficient use of time. The missed-pass risk in that case would be about 75,875 storage units (i.e. $75{,}875 = 120{,}875 - 3\times15{,}000$). If we double the interval length to twenty-four hours and the data volume to 30,000 storage units, then the missed-pass risk value becomes about 30,875 storage units.

This is still well above our yellow and red-alarm limits for the missed-pass risk. We, of course, had to consider other engineering factors before approving the decision to change the spacing of the telecom sessions to twenty-four hours. From the perspective of a data volume, on-board storage and missed-pass risk, we could say that the change was acceptable. Now, we only ever have one telecom session per day during IRS campaigns. We make an additional fifteen to twenty minutes per day available for science observations for every day during which we have only one telecom session. It also reduces the project's demand for time on the ground antennas. This is especially important because our telecom sessions are very short compared to the set-up and tear-down time required by the ground stations.

A proposal to change the spacing of telecom passes to thirty-six hours was rejected because it did not allow enough time for the uplink of sequences and frequent enough contact with the observatory for health and safety. Such a schedule with about 40,000 to 45,000 storage units per session, worst case, would also lower the missed-pass risk value below zero.

The IRAC can produce a larger amount of data than does the IRS. On about half to three-quarters of days during IRAC campaigns, however, the data volumes are small enough and the numbers in the missed-pass risk column large enough that we can eliminate one of the telecom sessions. The MIPS data volumes are consistently too large to allow this.

At one time, we investigated the possibility of scheduling the telecom sessions asymmetrically during the day. Instead of one telecom session about every twelve hours, we might alternate between eight- and sixteen-hour spacing. The idea was that a single operator could work both telecom sessions in a single twelve-hour shift. We found that, among other things, the risk of filling the on-board storage became too great, even for the same net data volume. We can model this problem in a simple way like this. Say that for twelve-hour spacing we produce 35,000 storage units per session. That translates to a missed-pass risk of 15,875 storage units each session. That's well above our yellow-alarm limit with some margin to account for variations in data volumes. If we change to the eight- and twelve-hour spacing, the data volumes alternate between 23,333 storage units and 46,667 storage units accordingly. That in turn causes the missed-pass risk value to alternate between 27,542 storage units and 4,208 storage units. It is 27,542 storage units when we must store the data from two short sessions and one long, and 4,208 storage units when we must store two long sessions and one short. The 4,208 storage units value is below our red-alarm limit. While this is a simple example, the asymmetry of the missed-pass risk was one consideration in deciding not to implement asymmetric telecom-session spacing.

Speculatively, once the fill-avoidance patch has been fully implemented, we may revisit our policy on mitigations to be taken based on the value of the missed-pass risk. For example, we may allow the missed-pass risk to reach some negative value before taking any steps toward mitigation. We have not yet done the trade study for this option. But, if we were to choose this course, we would trade some fault tolerance for a gain in efficiency and ease of scheduling observations. The fill-avoidance patch would reduce the lost observing time due to filling the on-board storage. It would provide graceful degradation. Instead of losing several days of time to recover from standby or safe mode, we would lose only the time of several individual science observations. In addition, this second option would require no special response from the mission operations system. A similar line of analysis might also allow a change to one telecom session per day during all instrument campaigns.

## IV. Responding to Anomalies That Affect the Downlink System

The sort of pre-flight analysis described above can be extended to track the history and current state of the on-board storage and to extrapolate from that current state into the future. We have built an on-board storage prediction tools that measures the accuracy of data-volume predictions, calculates predicted and actual measures of risk, predicts if the on-board storage is in danger of filling in case of an anomaly, predicts how many telecom sessions it will take to clear any backlogged data, and to support the choice of the best recovery strategy. We have a functional version of the tool. We have identified some enhancements that could be made as incremental improvements or to account for analysis techniques and recovery strategies designed since the tool was built.

The primary functions of the tool is to take predictions of the data volumes and convert them to predicted telemetry values, see Table 1, and to take actual telemetry values and convert them back into actual data volumes. The predicted data volumes are then compared to the actual data volumes and the predicted telemetry is compared to the actual telemetry. From these comparisons, the tool calculates the number of backlogged storage units, or variances, in the different states shown in Figs. 1, 2 and 3.

### A. The Architecture of the On-Board Storage Prediction Tool

The on-board storage prediction tool is divided into several pages. The primary pages all calculate the same set of information, but starts with different values as inputs. That is, sometime the calculation is done from data volumes to telemetry and sometimes from telemetry to data volumes. In addition to the primary pages, there are pages that calculate variances. They take the difference between pairs of primary pages. As shown in Figs. 1, 2 and 3, variance is always calculated as actual value minus predicted value. In addition to the primary pages and the variance pages, two supplemental pages do not follow the same format. The supplemental pages compile key pieces of information from the other pages and show it in a concise format. Finally, there are two pages used for input of data, with little or no calculation, and two pages used to override those values that were input. See Table 8.

Each of the primary, variance, input and override pages has the same format of information. There is one row for each telecom session. The tool models the state at two points during each session. Each row covers the span of time from the end of the previous telecom session to the end of the current one. For each row, the columns fall into six categories: 1) catalog and identifying information, 2) measures of risk, 3) actions between sessions, 4) state at start of session, 5) actions during the session, and 6) state at the end of the session. See Table 9. The special pages also have one row for each telecom session. But, the set of columns is different.

The on-board storage prediction tool has two basic inputs: predicted data volumes, and actual telemetry values. These are input into the tool via the input pages. If needed for what-if analysis, these values can be overridden by using the override pages. The predicted data volumes are ingested into the tool once or twice per cycle of the

sequence generation process. For us, this amounts to a couple of time each week. The telemetry values are ingested after every telecom session, once or twice per day. The tool maintains a history of each telecom session that has taken place and a prediction for each telecom session as far into the future as we have data volume predictions. The history can be purged if the older telecom sessions are no longer needed.

| Page | Category | Purpose |
|---|---|---|
| Predicted | Primary | Converts predicted data volumes into predicted telemetry values. |
| Actual | Primary | Converts actual telemetry values into actual data volumes |
| Extrapolation | Primary | For past sessions shows the actual telemetry values and data volumes. For future sessions, it extrapolates from the most recent Actual data using Predicted data volumes. |
| Perfect Predicted | Primary | For past sessions, uses the Actual data volumes and re-predicts what the telemetry would have been had there been not anomalies. For future sessions, it extrapolates from its most recent actual state. |
| Perfect Extrapolation | Primary | The same as Perfect Predict, except that it assumes that all backlogs can be cleared in a single session (i.e. that each session has infinite capacity). |
| Extrapolation Comparison | Variance | Compares the extrapolation and predict pages (extrapolation – predict). |
| Perfect Predict Comparison | Variance | Compares the perfect predict and predict pages (perfect predict – predict). |
| Perfect Extrapolation Comparison | Variance | Compares the perfect extrapolation and predict pages (perfect extrapolation – predict). |
| Summary | Supplemental | Displays a subset of information from predict, extrapolation and perfect extrapolation comparison pages for easy reference. |
| Mitigation | Supplemental | Displays information from predict and extrapolation pages, determines which sessions need backups for uplink and downlink, and does special calculations in support of the pre-emptive delete process. |
| Actual | Input | Ingests the actual telemetry values. |
| Actual Overlay | Overlay | Allows actual telemetry values to be overridden. |
| Predict | Input | Ingests the predicted data volumes. |
| Predict Overlay | Overlay | Allows the predicted data volumes to be overridden. |

**Table 8. Different pages of the on-board storage prediction tool.**

The Predicted page takes the data volume predicts, with any optional overlays, and calculates the predicted telemetry values and measures of risk that would result. It does not take into account any actual telemetry values or actual data volumes. It assumes that there are no anomalies that affect downlink and therefore that all storage units that are transmitted are received and no storage units need ever be retransmitted. It takes the predicted data volumes at face value and does not try to compensate for variation in compression ratios. It covers whatever time range for which we have data volume predicts. The initial condition is assumed to be that the on-board storage is empty.

The Actual page takes actual telemetry values and converts them into actual data volumes and measures of risk. The actual telemetry reflects the result of any anomalies that affect the downlink system. However, the actual calculated data volumes are independent of any anomalies since they are strictly a spacecraft event. The actual data volumes are later used in the Perfect Predict and Perfect Extrapolation pages.

The Extrapolation page calculates differently for telecom sessions that have already occurred and those that are in the future. For sessions in the past, the Extrapolation page shows exactly the same information as the Actual page. For sessions in the future, it uses the most recent actual telemetry as initial conditions and re-predicts the future telemetry values and measures of risk using the predicted data volumes from the Predict page. We use this page to predict what will happen on the spacecraft. It tells us if we are likely to fill the on-board storage and how many telecom sessions it will take to clear any backlog.

The Perfect Predict page is like the Extrapolation page in that it calculates differently for telecom sessions that have already occurred and those that are in the future. For sessions in the past, the Perfect Predict takes the actual data volumes and re-calculates what the actual telemetry values and measures of risk would have been if there had been no anomalies in the downlink system. In other words, this page shows you what the telemetry and measures of risk would have been if we were able to predict data volumes exactly and if the downlink system had operated perfectly. For sessions in the future, this page starts with the re-calculated telemetry values from the most recent telecom session as initial conditions and re-predicts the future telemetry values and measures of risk based on predicted data volumes. The primary purpose of this page is to serve as a comparison to the Extrapolation page for the purposes of measuring variances.

The Perfect Extrapolation page is the same as the Perfect Predict page except that for future telecom sessions, it assumes that the capacity of storage units that may be transmitted in the session is infinite. The original intent for this page was to measure the amount of backlog that occurs when not all the not-yet-sent storage units can be transmitted during a telecom session – that is when the percentage full is greater than one-hundred percent. For telecom sessions that have already occurred, the page works well. However, it does not work well for future telecom sessions. We may decide to remove this page in the future.

| **Category** | **Column** |
|---|---|
| Catalog Information | Day of week |
| | Sequence ID |
| | Telecom session ID |
| | Start time of session |
| | End time of session |
| Measures of Risk | Percentage Full |
| | Missed-Pass Risk |
| | Six-Pass Margin |
| | Six-Pass Percentage |
| Actions between sessions | Volume written to on-board storage |
| | Volume acknowledged |
| | Volume freed by fault protection |
| State at start of session | Free storage units |
| | Not-yet-sent storage units |
| | Sent storage units |
| | Storage units in the retransmit queue |
| | Storage units retransmitted |
| | Storage units discarded by acknowledgement |
| | Storage units discarded by de–allocation |
| Actions during sessions | New data transmitted |
| | Old data retransmitted |
| | New received on ground |
| | Old received on ground |
| | Volume written to on-board storage |
| | Volume freed by fault protection |
| State at end of session | *same as at start of session* |

**Table 9. Options to mitigate single yellow and red alarms in the missed-pass risk column.**

The Extrapolation Comparison page takes the difference between the Extrapolation page and the Predict page. The primary purpose for doing this comparison is to calculate the difference between the predicted data volumes and the actual data volumes. This is useful for tracking the accuracy of the predictions and removing any systematic errors via the Predict Overlay page if desired.

The Perfect Predict Comparison page takes the difference between the Perfect Predict page and the Predict page. The purpose is to calculate the variances in the telemetry values and in received versus transmitted storage units. We use this is the mechanism to calculate how many storage units of backlog there are at each transition in Figs. 1, 2 and 3.

The Perfect Extrapolation Comparison page is the same as the Perfect Predict Comparison page, except that it takes the difference of the Perfect Extrapolation page and the Predict page. The purpose is to calculate the variances in the telemetry values and in received versus transmitted storage units. However, like the Perfect Predict page, it does not do a good job for future telecom passes.

The Summary page gathers the catalog and identifying information, the measures of risk, the data volumes from the Extrapolation page, the free storage units at the beginning of the pass and the free, sent and not-yet-sent at the end of the pass from the Extrapolation page, and the backlogs at the end of the pass from the Perfect Extrapolation Comparison page. The purpose of this page is to collect the most frequently used information in one place.

The Mitigation page gathers the catalog and identifying information, the measures of risk and the data volumes from the Extrapolation page. Then it determines which telecom sessions need mitigation and redundancy according to the rules set forth in Table 7. Further, it calculates the volume of data that would need to be deleted by a pre-emptive delete to raise the missed-pass risk above the yellow-alarm limit value.

## B. Analysis of the Tool's Output and Limitations

As stated above, the tool is updated with new telemetry input after each telecom session, and with new predicted data volume a couple of times per week. The main output of the tool is the telemetry values and measures of risk from the Extrapolation page, and the past and future backlogs of data (i.e. variances) from either the Perfect Predict or Perfect Extrapolation pages. From this information, the most important questions to answer during any anomaly are: 1) will the on-board storage fill, 2) how long will it take to clear the backlog, if any, 3) what are the possible consequences from a second failure before we have recovered from the first, 4) what recovery strategy should we pursue.

Figure 8 shows an example from an actual anomaly presented in graphical form. In it there are thirteen telecom sessions and twelve intervening periods of making science observations. The solid lines represent the actual free storage units and missed pass risk, measured and calculated after the fact. The dashed lines represent the predicted values, from the Perfect Predict page of the tool. The solid red line is continuous data. The dashed red line and both blue lines are only calculated at the solid dots.

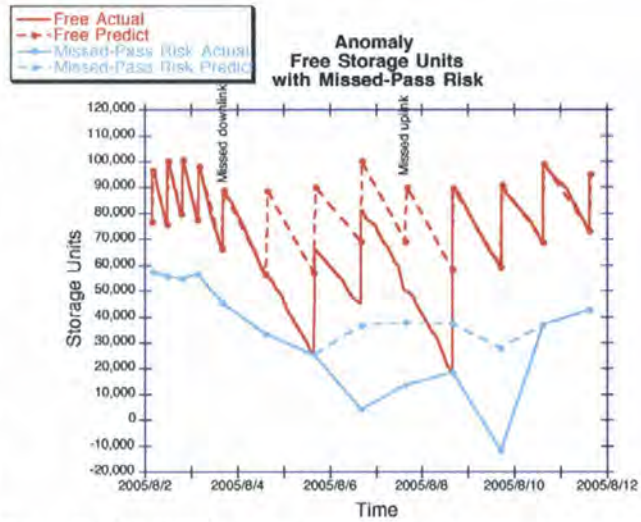


**Figure 8. Predicted and actual free storage units and missed-pass risk during overlapping downlink anomalies.**

Let us call the telecom session with the missed downlink session n. During session n, we missed the downlink. That is, we received no storage units. We were able to uplink the commands to acknowledge the data received during session n-1. Hence, the number of free storage units rose as predicted during session n. During session n+1, though, we have no acknowledgement commands to send because we missed the previous downlink. The number of free storage units does not rise during that session. So, during session n+2, the number of free storage units bottoms out at the predicted missed-pass risk number, about 25,000 free storage units. This is consistent with the rule that a missed downlink will cause the free storage units to reach their minimum two telecom sessions after the anomaly. A missed downlink during session n is equivalent to a missed uplink during session n+1 and causes the free storage units to reach a minimum during session n+2.

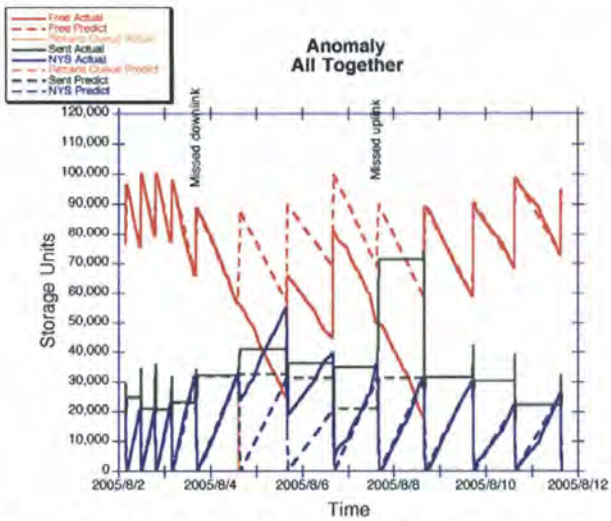


**Figure 9. Predicted and actual free, sent, not-yet-sent and retransmit-queue storage units.**

During telecom sessions n+2 and n+3, we can see that the actual number of free storage units is gradually returning to the predicted values. Although it is not visible looking only at Fig. 8, with session n+1, we began to retransmit the missed data and clear the backlog. We were able to transmit more storage units than predicted starting with session n+1 because there was margin built into the original session, and we extended the sessions via real-time commands. The rise in free storage units during sessions n+2, and n+3 is slightly larger than would otherwise have been.

During session n+4 we missed the uplink. We were unable to send the command we had built to acknowledge the data received during session n+3. Like in session n+1, the number of free storage units did not rise. At the start of session n+5, the number of free storage units bottoms out at the new, revised missed-pass risk of about 19,000 free storage units. If we had further missed the uplink during session n+5, we would clearly have filled the on-board storage. The missed-pass risk for session n+6 was less than -10,000 free storage units. Session n+5 had no problems. We were able to acknowledge the storage units received during sessions n+3 and n+4.

By pure co-incidence, during session n+4 we received the last of the backlogged not-yet-sent storage units. So, during session n+5, the actual number of free storage units agreed again with the predicted number. We recovered simultaneously from the missed downlink during session n and the missed uplink during session n+4. During this whole incident, we demonstrated that we were single-fault tolerant and that we recovered in fewer than the required six telecom sessions from the original incident.

We can view the four primary telemetry values, both predicted and actual, together on one graph. See Fig. 9. The missed-pass risk has been omitted from this plot. There are several things to note about the telemetry values. We can see that during telecom session n+1, there was a large retransmission of data, the solid orange line. These are the data that were not received due to the missed downlink during telecom session n. The retransmission did not fill the entire downlink capacity of telecom session n+1. We can see this because the solid blue line representing not-yet-sent storage units did drop slightly during telecom session n+1. By looking at the actual not-yet-sent storage units, we can see more easily that we are making a net gain at clearing the backlog. It reaches zero at the end of session n+4. Each telecom session ends with the number of not-yet-sent storage units closer to zero. The number of actual Sent storage units, the solid green line, remains greater than the predict value all the way until telecom session n+5.

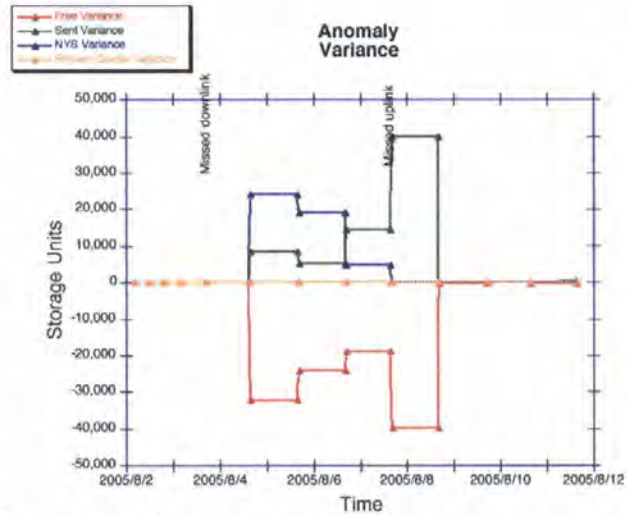


**Figure 10. Variances of free, sent, not-yet-sent and retransmit-queue storage units.**

We can look at this in a different way if we plot the variances themselves. See Fig. 10. The sum of the variances of free, sent and not-yet-sent is always zero. This is true here because we have shown the variances relative to the Perfect Predict page retrospectively. Notice that the variances are calculated only at the start end of the telecom sessions. The variance for number of storage units in the retransmit queue is always zero. The prediction for the number of storage units in the retransmit queue is always zero. Because the large retransmission during session n+1 completed before the end of the session, no variance appears in Fig. 10. If that retransmission had not finished in session n+1 but carried over into session n+2, then a variance would be visible.

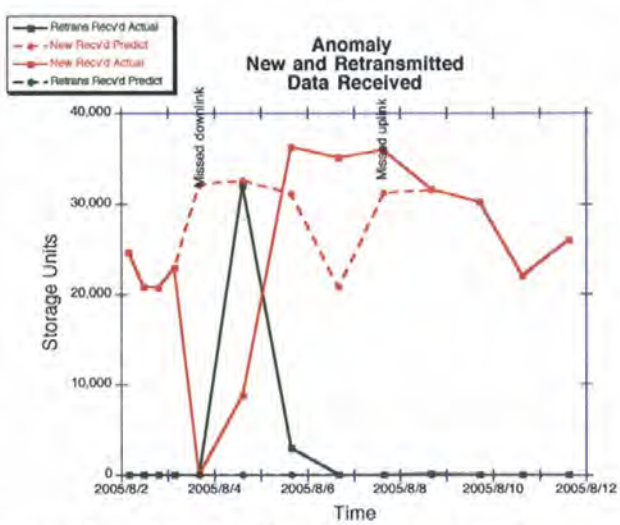


**Figure 11. Actual and predicted numbers of new and retransmitted storage units received.**

We can also do a similar analysis for the number of storage units received on the ground during each telecom session, see Fig. 3. Figure 11 shows the number of new and retransmitted storage units actually received compared to the number predicted to be received. The number of retransmitted storage units predicted to be received is always zero. We can see that the number of storage units received was zero during telecom session n, the missed downlink. Then, for telecom sessions n+1 through n+4, the missed uplink, the total number of storage units received was greater than predicted. The number received was greater than predicted because there was some built-in margin in the downlink, the percent full was less than one-hundred percent, and we extended each of these telecom sessions by a few minutes via real-time command.

We can see the same pattern in Fig. 12. Figure 12 shows the variances of new and retransmitted storage units received over the course of this anomaly. During session n, the new storage units showed a large negative variance. Then, during session n+1, the retransmitted storage units showed a large positive variance while the variance for

new storage units was still negative. Then, for sessions n+2 and n+4, the variance in the number of new storage units received was positive, indicating that we were making progress clearing the backlog of data. Figure 12 shows most clearly of all the recent graphs that the missed uplink did not cause any retransmission of storage units. Also, we can see that Figs. 11 and 12 are the only graphs that show any indication of a problem during telecom session n. All the other measures indicate a problem starting with either session n+1 or n+2.

### C. Recovery Strategies

We have developed many operational strategies and techniques to recover from anomalies that affect the downlink system. The most important goals during any recovery are: 1) determine if and when we will fill the on-board storage assuming that there are no further anomalies, 2) start the retransmission of any missed data during the next telecom session, 3) lengthen the time available for downlink in each session until the transmit backlog is cleared, 4) predict how long it will take to clear the backlog, 5) if there is a backlog of data to transmit, pre-emptively acknowledge data during the recovery to reduce the risk of filling the on-board storage due to a second failure.

Many of our techniques to accomplish the above goals result from the peculiarities of our downlink system, both on the observatory and on the ground. We have come up with ten to fifteen detailed scenarios for single-point failures.

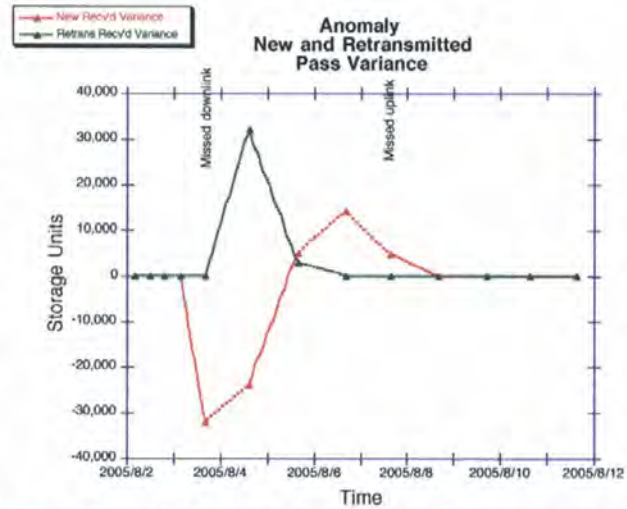


**Figure 12. Variances of the actual and predicted numbers of new and retransmitted storage units received.**

One of the primary problems we face is that of the end gap. The ground software that generates the acknowledgement and retransmission commands will not request retransmission of data unless and until newer data of that same type has been received. Several attempts to work around this problem have caused other problems of their own, resulting in the unnecessary transmission of the same data twice in a single telecom session, thus wasting downlink time and creating a backlog. The ground software cannot know if the latest data it has received is also the latest data that the observatory has transmitted. One way to fix this in future missions would be to add a telemetry channel for each data type that would periodically tell the ground the most recent spacecraft clock time it has ever transmitted. That would solve the problem in most of the single-point failure scenarios. The only time that the problem would remain is in the case of a completely missed downlink.

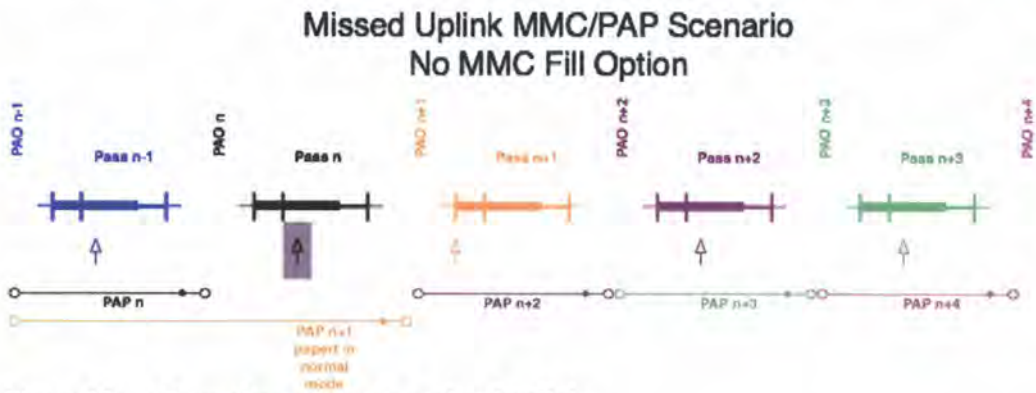


**Figure 13. Example response to a missed uplink.**

With our current system, in the case of a missed downlink, we must manually build a retransmit command to start the retransmission in the following telecom session. This is because a missed downlink creates a large end gap. Our normal retransmission process would not start retransmission until one telecom session later than that. Since not all data types are generated during every telecom session, a subset of end gaps may linger for several days to several weeks without manual intervention.

Sometimes we must not send commands to retransmit data. If the retransmit queue is not empty at the beginning a telecom session, a retransmit command stands a chance of requesting the retransmission of data that are already in the retransmit queue. This will cause those data to be placed twice in the retransmit queue. On the ground, we do

not have visibility into the order and contents of the retransmit queue. Skipping the retransmit commands in this case does not often cause a problem, however. There is usually enough data waiting to be transmitted or retransmitted on board the observatory that no downlink time during the telecom session is left idle. However, simpler rules for the order of transmission of data in the retransmit queue and more visibility on the ground into the contents of the queue would make this process simpler.

Before launch, we had developed a different technique to lengthen the downlink time during telecom sessions. We would gain more downlink time by delaying the end of the telecom session. We did this by changing the on-board sequence to remove and re-order science observations that immediately followed the telecom session. We have used this procedure once during or nominal mission and it worked well. It has several practical limitations that will probably preclude its use in the future. It takes use from twenty-four to forty-eight hours to build the revised sequence prior to uplink. Since our telecom sessions can be as close as twelve hours apart, this technique produces its effect too late. The procedure to upload and activate the modified sequence requires several real-time commands during the telecom session. It will become more difficult to manage that process as our one-way light time increases over the remainder of our mission. The risk of filling is always greatest during the one or two telecom sessions following the anomaly. The main benefit of the technique is that it allows us to clear the backlog more quickly. Clearing the backlog more quickly by itself does not reduce the risk of filling the on-board storage in the next two sessions. It is operationally simpler and quicker to start the playback of data early using our real-time command as described before.

Our downlink system is particularly sensitive to a stream of data that has many small gaps. Remember that on board, we must delete all the data in a single storage unit with a single command. Otherwise, the storage unit is not freed. Since each storage unit is sixteen transfer frames long, if we have a problem that deletes every twelfth transfer frame for example, we will be unable to delete any of those storage units even though we have eleven twelfths of the data on the ground. We have had this problem several times in flight. In cases like this, we must consider any such gappy to have been missed. In this example, we would have to retransmit twelve times as much data as we were actually missing. We must also follow special procedures with our ground tools because such gappy data causes the ground acknowledgement software to generate sequences that are too long to complete execution during the telecom session.

The fifth recovery strategy listed at the beginning of this section is particularly difficult to implement. We have used it a few times, but it has proved difficult. The problem with it is that we must build commands in advance to pre-emptively delete the data that we think will be transmitted during the next telecom session. There are very tight timing and other constraints on sending such commands. The state of the data on-board the observatory during an anomaly is not always clear. We have very good insight into *how much* data are on-board, but not *which* data are on-board. The commands we have on the observatory must be told which data to delete. If we are ever able to increase our network capacity and get enough data to JPL during the telecom session to build and send a pre-emptive delete command based on actual data received, that will greatly improve this process.

### D. Benefits of the Tool

The on-board storage prediction tool makes managing a complex process possible. Before we developed the tool, we would have to do an *ad hoc* analysis for each anomaly. Those analyses were never as rigorous or systematic as the analysis done by the tool. The tool gives us the data we need quickly to determine the best recovery strategy to use for a given anomaly. It has had other, more specific benefits as well.

It has let us do a routine comparison of the actual data volumes to the predicted data volumes. This comparison revealed a number of ways in which the accuracy of our data-volume predicts could be improved. Most improvements were simple, like correctly accounting for packet headers, etc. Others were more complex like correctly estimating the volume of engineering and housekeep data. We were also able to introduce a mechanism whereby we can scale the data volume predictions via factors contained in a configuration file. This allows us fine-grained control to adjust for the unpredictable compression ratios on board the observatory.

The tool has also alerted us to several other issues related to the on-board storage. As part of our launch sequence, about nine-hundred storage units were put in a protected state and taken out of regular use. The tool allowed us to discover that fact and subsequently make a plan on if and when to recover them.

Occasionally, we receive a request to give up our previously scheduled time with ground antenna so that the antenna may be used for some high-priority maintenance or in support of another mission, which has had some anomaly. The tool allows us to analyze the effect of missing that telecom session. The results from the tool let us negotiate use of a different ground antenna for the same time. The second antenna might only be able to support a reduced downlink bit rate, or might only be available for a portion of our telecom session. The analysis provided by the tool lets us give quick and meaningful responses to such requests.

One of the capabilities of the Spitzer mission is to make observations of targets of opportunity. We schedule some such targets of opportunity as part of the normal sequence process weeks in advance. Other times, we must change the sequence that is currently executing and observe the target within forty-eight hours. The on-board storage prediction tool has been used as part of that process to analyze the impact of the change. What we have found is that scheduling a target of opportunity on such short notice usually reduces the amount of data generated. This is to be expected since observing a target of opportunity will usually introduce more time to re-point the observatory and perhaps even change science instruments.

## V. Future Possibilities and Applicability to Other Downlink Systems

One of the most important lessons learned from doing this analysis and building this tool is that time to transmit data, and space to store data must be managed as separate resources. Making sure that we have sufficient time to downlink the data during each telecom session does nothing to insure that we will not fill the on-board storage if we miss an uplink or downlink. Similarly, making sure that the missed-pass risk is always above our yellow-alarm limit will not insure that we always have enough time to downlink all the data during each telecom session. It will create a backlog and eventually fill the on-board storage.

We would like to extend the functionality of the on-board storage prediction tool. We would like the tools to support more simply all of the recovery procedures outlined in the previous section. Some of these changes only need to be made to the user interface. For example, many analyses must be made manually through the two overlay pages in the tool. It would be more convenient and more reliable if the tool supported those options with special-purpose features. The primary features that would benefit from these are: 1) planning a pre-emptive delete command, 2) extending the length of a telecom session via real-time command, 3) planning for a specific missed uplink or downlink in advance, and 4) planning for a data-rate change in advance.

We would also like to add some new fundamental capabilities to the tool. The tool currently does not measure the actual duration of telecom sessions. When the tool was first built, we did not have our current procedure to start the downlink early via real-time command. We only had the forty-eight-hour procedure to extend the downlink by removing science observations. Now that we regularly use the real-time commands to start the playback early, it would be useful to have the tool measure the amount of time by which we actually lengthened the playback.

We would like to have some capability in the tool to infer the presence and perhaps the specifics on an end gap in the data. This would help the flight controllers locate missing data. We sometimes have a small end gap due to a minor problem in the downlink. It is not large enough to cause concern about filling the on-board storage. However, it may elude us for several days and any way to detect the issue earlier would be helpful. The end gap is a specific example of a more general problem. The tool assumes that we always take the most aggressive approach to recovering data and clearing backlogs. It assumes that we always immediately request for retransmission all data that we did not receive. For a variety of reasons, we do not always do that. The magnitude of the missing data may not be large enough to justify the action.

Currently, the tool can model the state of the on-board storage only twice per telecom session, once at the beginning and once at the end. If we add the capability to model at more times than that, we can increase the functionality of the tool. The tool cannot now correctly understand when we use a pre-emptive delete command. Following the execution of a pre-emptive delete command, the tool will show incorrect results. The magnitude of this problem is usually minor, and we know how to compensate for it manually, so it does not pose a big risk. But, we could eliminate this confusion if we were to add third point at which to model the state of the on-board storage. That point would be after the execution of the nominal acknowledgement commands and before the execution of the pre-emptive delete commands. We could then measure the actual performance of our pre-emptive delete commands and reflect their effect correctly in the tool both before execution and after.

We could take the concept further and model the state of the on-board storage at times during the collection of science data. This could be used to track the accuracy of the data volume predictions for each scientific observation request. There is some uncertainty in the execution time of the science observations. This capability would need to be tied to the actual timing of science observations on the observatory. We could also connect this sort of analysis to the fill-avoidance flight-software change. We could predict which observations were most at risk to be skipped in case of an anomaly. This sort of information might be used to schedule observations in such a way that certain ones were less likely to be skipped.

Currently the tool reports on the difference between predicted and actual data volumes generated for the telecom session as a whole. If it were able to report the same information for each scientific observation, those results might be used to improve the data-volume prediction process. We have done this sort of per-observation analysis by hand

for several representative spans of time. That has helped us choose the previously mentioned scale factors for data volumes. A routine report at this level might improve the results even further.

We do not currently have requirements on the accuracy or precision of our data volume predictions. A statistical analysis based on the per-observation reporting mentioned above would help use refine the limits on which we base our choices of risk mitigation during the sequence-generation process. Our limits right now are empirical, but we have a long history using them and feel comfortable that they are sufficient. Our history shows that our data-volume predicts are somewhat conservative. We tend to over-predict the data volumes slightly. A more rigorous approach would potentially let us reduce the levels at which we choose to take action and thereby make the observatory more efficient.

Once the fill-avoidance flight-software change has been implemented and if we can make the capability to receive and acknowledge data pre-emptively during the course of a single telecom session, we may be able to reduce the frequency of our telecom sessions significantly. The first capability is scheduled to become available in the next few months. The second capability is currently not possible, but may become so as the network bandwidth of our ground data system is improved. Acknowledging a portion of the data pre-emptively in effect reduces the data volume of that telecom session. The size of the on-board storage often limits the amount of data that we can collect per telecom cycle. Pre-emptively deleting some data that we have actually received would allow us to collect more data per telecom cycle, without increasing the risk of filling the on-board storage or skipping observations. This would increase our efficiency by allowing us to schedule telecom sessions less frequently.

It seems that there are many different approaches to solving the problem of data storage and retransmission. This approach to risk management and analysis can be applied to many of them. Some missions do not provide for the retransmission of data. They have one chance to receive it on the ground and then it is automatically overwritten. Missions that use that scheme need only make sure that they do not overfill their on-board storage in any given telecom cycle and that they allow enough time for transmission. If such a mission allows data that is not transmitted in the first telecom session to be carried over to the next, then there would be a greater need for the techniques described here. Even without a retransmission system, they would need to make sure they did not fill.

Some missions use a ring buffer coupled with a retransmission system. In such a system, data may be requested for retransmission as long as it has not been overwritten. This sort of analysis could predict how long data would be stored on-board and be available for replay.

This sort of analysis is useful regardless of the mechanism used to acknowledge and retransmit data. Most downlink systems now use either the time of data collection, i.e. the spacecraft clock, to do the job. Sometimes it is done directly, and sometimes it is inferred from the time of transmission or receipt on the ground. An even better solution would be to use a file-based system, like CFDP. But regardless, as long as the data volumes, the time of storage needed and the duration of transmission, this sort of analysis can be put to good use.

## Acknowledgments

The work described in this paper was carried out at the Jet Propulsion Laboratory and the California Institute of Technology under contract to the National Aeronautics and Space Administration.

## References

TBS