

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

Marc A. Sarrel<sup>\*</sup>, Carlos Carrion<sup>†</sup> and Joseph C. Hunt Jr.<sup>‡</sup>

*Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, 91109-8099 USA*

**The Spitzer Space Telescope has a two-phase downlink system. Recorded data are transmitted during one telecom session. Then commands are sent during the next session to delete those data that were received on the ground and to retransmit those data that were missed. We must build science sequences that are as efficient as possible to make the best use of our 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 science 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 (i.e. deletion) 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.

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. 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.

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 triggering a fault-protection response. 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 in the on-board storage for the predicted volume of data.

This is further complicated because the observatory has a fixed high-gain antenna. It is not possible to make scientific observations and point the high-gain antenna at Earth simultaneously. In addition, the CPU on board the observatory is not capable of doing both simultaneously even if we were to disregard the fixed high-gain antenna.

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 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 on-board storage, and the time it will take to 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 us *how much* data, but not *which* data.

---

<sup>\*</sup>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.

<sup>†</sup>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.

<sup>‡</sup>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.

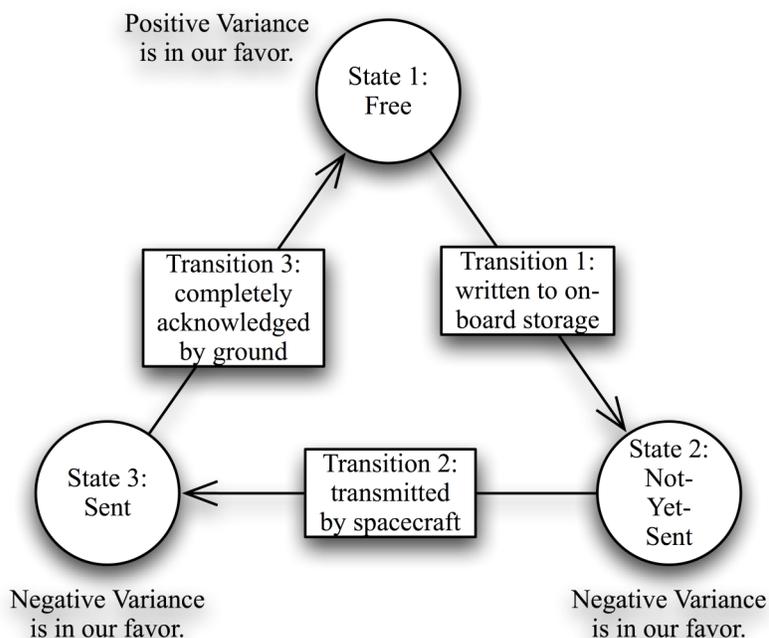
We discuss the software tool that we have developed to analyze the current and future state of the on-board storage. The tool calculates the size, type and urgency of any backlogs, and guides the choice of actions to recover from an anomaly during sequence execution. We also discuss the procedures themselves and the failure-mode analysis used to develop them.

We discuss the future development of the tool. Generally, the tool assumes that the most aggressive approach for acknowledgement and retransmission of data will be taken. This is not always necessary. In 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 avoid 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 finally 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, storage units do not have headers 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.**

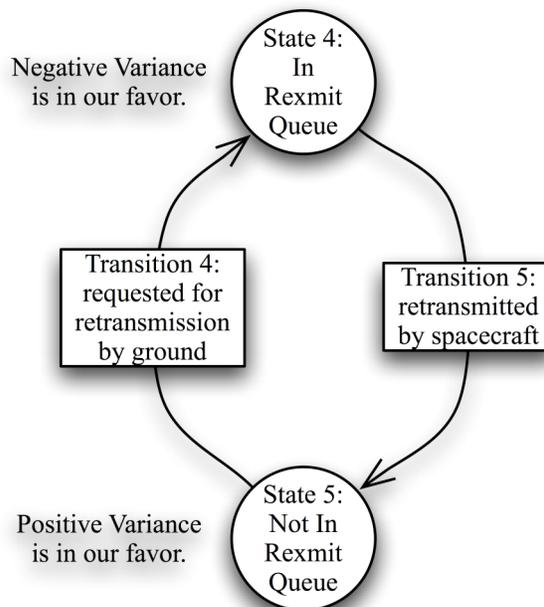
(Transition 1). A not-yet-sent storage unit becomes sent when transmitted for the first time by the observatory (Transition 2). Finally, a sent storage unit becomes free when the ground sends commands to delete all the data it contains (Transition 3).

The sum of the number of storage units marked as free, the number marked as not-yet-sent and the number marked as sent is constant and 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.

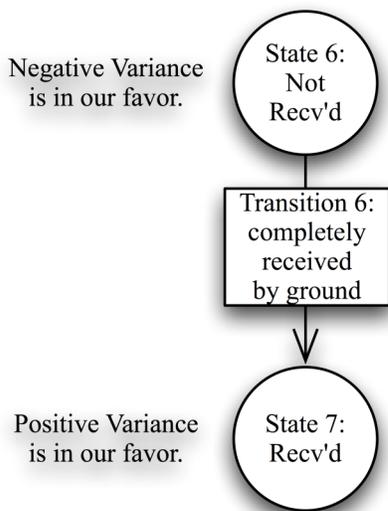
The observatory maintains several queues of storage units. The first queue is just a list of the storage units marked as free. For each application identifier (APID), there are two queues, one for storage units marked as not-yet-sent, and one for storage units marked as sent. Each storage unit contains packets of only a single APID.

There is also another queue of storage units, holding those storage units that have been requested by the ground for retransmission. 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 many APIDs. The storage units are not copied when they are put in the retransmit queue. They must be in one of the other queues mentioned above; they are never in the retransmit queue alone. A *negative* variance of storage units 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 4). They are removed from the retransmit queue when they are retransmitted by the observatory (Transition 5). 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. However, 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.

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.



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



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

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 6), 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 to the number 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 (i.e. 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 and

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.

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 and receives commands from the Earth. During these alternating periods, the numbers of storage units in each of the states from Fig. 1 and Fig. 2 follow 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 operation, nominally only Transition 1 from Fig. 1 is made. That is, the only thing that happens is that a large number of storage units are written to the on-board storage. No transitions from

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.**

Fig. 2 or Fig. 3 are made. These storage units are mostly from science observations, with a small number from engineering and housekeeping activities. 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 4. Finally, those storage units are retransmitted, Transition 5.

During the period of autonomous operation, 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 4 nor Transition 5 is made.

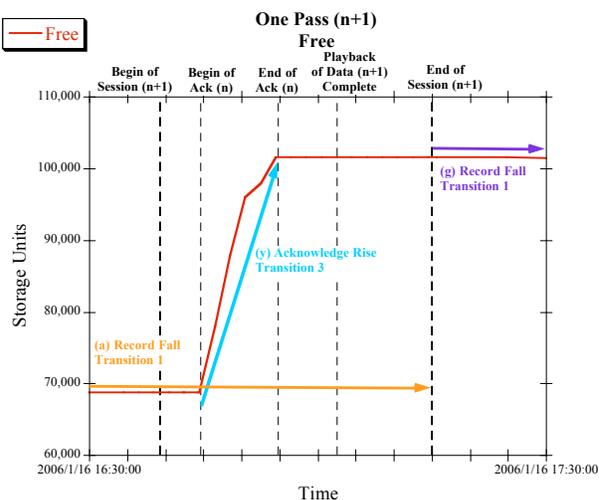
During the telecom session, the following things happen to the four measurements. The number of free storage units decreases slightly due to Transition 1, but increases much more due to Transition 3. In the nominal case, there is always a net increase of free storage units over the telecom session. The 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 4 and increases because of Transition 5. 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.

### 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 is too small to be visible on the scale of the y-axis.

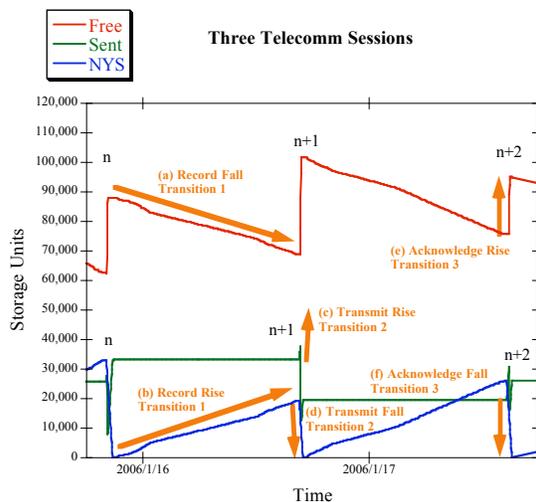
The orange arrows show the course of one set of data as it is recorded, transmitted and then acknowledged. During the period of autonomous operation between telecom sessions  $n$  and  $n+1$ , the data represented by the orange arrows are recorded. Transition 1 causes the number of free storage units 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, 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). 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 orange 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 key 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 end time 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 the acknowledgement depend on when the real-time commands for acknowledgement are sent to and executed on the observatory.



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

Figure 5, arrow (a) represents the data recorded from the end of telecom session  $n$  to the end of telecom session  $n+1$ . Figure 5, arrow (g) represents the data recorded from the end of telecom session  $n+1$  until the end of telecom session  $n+2$ . Whether it is represented by Figure 5 arrow (a) or arrow (g), Transition 1 is always reducing the number of free storage units.



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

Figure 5, arrow (a) represents the data recorded from the end of telecom session  $n$  to the end of telecom session  $n+1$ . Figure 5, arrow (g) represents the data recorded from the end of telecom session  $n+1$  until the end of telecom session  $n+2$ . Whether it is represented by Figure 5 arrow (a) or arrow (g), Transition 1 is always reducing the number of free storage units.

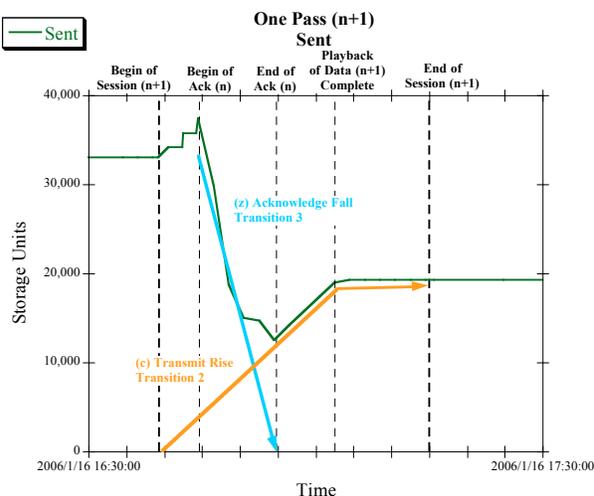
Transition 1 and Transition 3 from Fig. 1 affect the number of free storage units. Transition 1 happens all the time because data are always being written to the on-board storage. This is illustrated by arrow (a) and arrow (g) in Fig. 4 and Fig. 5. We make artificial boundaries at the ends of the telecom sessions. Figure 5, arrow (a) represents the data recorded from the end of telecom session  $n$  to the end of telecom session  $n+1$ . Figure 5, arrow (g) represents the data recorded from the end of telecom session  $n+1$  until the end of telecom session  $n+2$ . Whether it is represented by Figure 5 arrow (a) or arrow (g), Transition 1 is always reducing the number of free storage units.

Figure 5, arrow (g) represents the data recorded from the end of telecom session  $n+1$  until the end of telecom session  $n+2$ . Whether it is represented by Figure 5 arrow (a) or arrow (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.

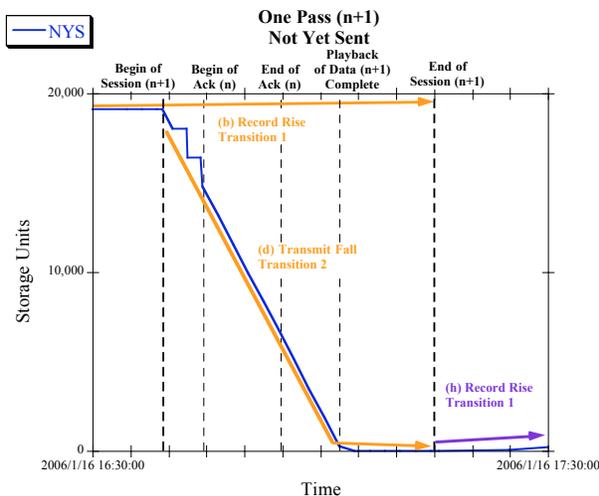
Figure 6 shows what happens to the number of not-yet-sent storage units during telecom session  $n+1$ . As before, Transition 1 occurs all the time. In Fig. 6, it is represented by arrow (b) and arrow (h). Transition 1 causes the number of not-yet-sent storage units to rise. From the beginning of the 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 playback of data is complete when they hit zero. Then, they stay at or very close to zero 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. The strange stair-step shape of the blue line at the beginning of the telecom session is just due to a small gap in the telemetry.

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.



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

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 (z) causes the number of sent storage units to fall. The interaction of Transition 2 and Transition 3 give the curve in Fig. 7 its characteristic zigzag 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. The strange stair-step shape of the green line at the beginning of the telecom session is just due to a small gap in the telemetry.



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

### 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 sequences are about one week long in

usually regain it once the next uplink is successful. In the case of a missed downlink, it may take several telecom sessions. Our guideline is that we must clear the backlog within six telecom sessions after a failure.

### 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

Telecom Session	Start of Telecom Session		End of Telecom 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.**

If during session n, we are unable to send the real-time commands 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+1, we will send real-time commands to acknowledge both sets n and n+1 and at the end of session n+1 the system will have returned to normal. See Table 3.

Telecom Session	Start of Telecom Session		End of Telecom 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.**

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 that must be cleared over the succeeding telecom sessions before we will have returned to normal. See Table 4.

Telecom Session	Start of Telecom Session		End of Telecom 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.**

retransmission of set n. Similarly, at the end of session n+2, some of set n+2 may have been transmitted and moved to sent if both the retransmission of set n and transmission of set n+1 have completed. 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

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.

If we are unable to receive the data transmitted by the observatory during telecom session n, i.e. Transition 6 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

In this example, 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 due to the

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 that must be cleared over the succeeding telecom sessions before we will have returned to normal. See Table 5.

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

Telecom Session	Start of Telecom Session		End of Telecom 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  $n$ .**

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. 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 are not needed and so can be used as extra downlink time to clear data backlogs. Typically, we can gain six of those seven minutes as extra downlink time per telecom session.

## 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 telecom 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 interval, 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 session. The third item is calculated by taking the total transmission rate of the observatory, subtracting off a small portion to account for real-time data that are never recorded to the on-board storage and then multiplying by the duration of the session. 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 to 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. However, this would provide little or no benefit. It 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 the 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 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 (SU)	Predicted Volume (SU)	Percentage Full (Percent)	Missed-Pass Risk (SU)	Six-Pass Margin (SU)	Six-Pass Percentage (Percent)
n-6	34,718	23,232	66.92%	34,081	47,395	80.23%
n-5	33,562	26,249	78.21%	35,635	38,360	83.80%
n-4	39,570	38,943	98.42%	32,451	25,531	88.46%
n-3	34,781	27,905	80.23%	27,778	30,648	86.26%
n-2	34,072	21,052	61.79%	32,975	24,887	89.05%
n-1	37,343	26,224	70.22%	45,694	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. Therefore, 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, in column E only rows from the current sequence (i.e. n through n+12) must be analyzed. This is different from columns F and G, see below. The values in column E for sessions n-6 through n-1 are simply carried over from the previous sequence.

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, a more involved analysis 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 and once for the next. This can create a bow-wave effect that may ripple forward across several telecom sessions. When this simple analysis is used and telecom sessions are greater than one-hundred percent full, the numbers shown in column E will be higher than the true values. They will erroneously under-estimate the risk.

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 until the predicts for the next sequence become available. In this example, column F for sessions n+7 through n+12 are calculated based on incomplete information. We assume that all sessions from the following

sequence (i.e. sessions n+13 and later) 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 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 we have the predicts for the next sequence.

### **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 either a more detailed analysis, some form of mitigation or a waiver.

The definition for column D, the percentage full, originally stood at ninety-three percent. 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 seven-percent 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 also we have been able to increase the limit for column D to ninety-seven percent. Pending some further operation at the ninety-seven percent limit, we anticipate and increase to ninety-eight percent. 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 slew to Earth point. A slew that is longer than predicted can delay slightly the start of the telecom session. But, the magnitude of this effect is only a few percentage points.

For the missed-pass risk column, we set a yellow-alarm limit at ten-thousand storage units and a red-alarm limit at five-thousand 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 ninety-seven percent 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 ninety-seven percent 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 the consequences of a single failure are too great. Remember that our two requirements are that we not fill the on-board storage and that we be able to retransmit any single session's data within the next six sessions in case of a missed downlink. Once we have identified the areas of risk, we can take specific steps to mitigate that risk. Those steps fall into two categories.

The first category contains all those changes we can make to the background sequence on board the observatory. We can increase or decrease the values in columns B and C by changing the background sequence. The second category contains measures to increase redundancy on the ground or real-time commands. Things in the second category include getting backup ground antennas and deleting data from the observatory during the telecom session in which it is transmitted. 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.

We will first discuss methods of mitigation in the first category. 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 to efficiency of 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, we 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, we will decrease the risk 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 one thousand, the values in columns E and F decrease by one thousand.

If we shorten telecom session n, i.e. decrease the numeric value in column B, we 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 value in column E at all. 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 a more involved analysis.

Using a combination of the above direct and inverse analyses, we can easily identify for which telecom session we must 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. See Table 7. It can be easily accomplished by shifting the time of a telecom session and moving an observation from one side to the other. Areas of increased risk are shown in red, while areas of decreased risk are shown in green. Continuing with the example in Table 6, if we move 1,000 storage units from session n to session n+1, that will decrease the risk in columns D and E for session n, and in columns F and G for session n-6. It will also increase 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. More precisely, the changes cancel each other out.

However, we can also use this analysis to increase efficiency. If the missed-pass risk is around sixty- or seventy-thousand free storage units or above for a couple of telecom sessions in a row, then it is likely that one of those telecom sessions can be removed, thereby combining the two adjacent periods of autonomous operation, and making that time available for science observations. Even if the data volumes would allow removal of a telecom session, there 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. Secondly, if the percentage-full column is below around eighty or ninety percent, we may be able to shorten the telecom session 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 or six-pass percentage columns.

We have one set of methods to use when a value in the missed-pass risk column is below the yellow-alarm limit, see Table 8a and Table 8b, and another set to use when it is below the red-alarm limit, see Table 8c, Table 8d, Table 8e and Table 8f. 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. 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.

If there is a yellow alarm in the missed-pass risk column in telecom session n, we must provide a redundant

A	B	C	D	E	F	G
Telecom Session	Predicted Capacity (SU)	Predicted Volume (SU)	Percentage Full (Percent)	Missed-Pass Risk (SU)	Six-Pass Margin (SU)	Six-Pass Percentage (Percent)
n-6	34,718	23,232	66.92%	34,081	48,395	79.81%
n-5	33,562	26,249	78.21%	35,635	38,360	83.80%
n-4	39,570	38,943	98.42%	32,451	25,531	88.46%
n-3	34,781	27,905	80.23%	27,778	30,648	86.26%
n-2	34,072	21,052	61.79%	32,975	24,887	89.05%
n-1	37,343	26,224	70.22%	45,694	15,556	93.30%
n	31,398	27,705	88.24%	45,894	10,657	95.61%
n+1	30,595	30,301	99.04%	36,645	7,809	96.84%
n+2	24,000	23,508	97.95%	39,360	19,632	92.63%
n+3	36,714	35,759	97.40%	31,307	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 7. Example calculation of the various measures of downlink risk for a single sequence – effect to risk of moving 1000 SU from session n to session n+1.**

redundant, however, we only need use a single 34-meter antenna. This strategy relies on the fact that we have done the risk analysis calculations and determined that we can miss a downlink during session n-1, but not an uplink.

If we have a yellow alarm and a backup ground antenna is not available, we have an alternate method to mitigate the risk. See Table 8b. 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 the amount we pre-emptively deleted in session n-2. 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 network 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. 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 end-to-end ground data system 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.

uplink capability during telecom session n-1, see Table 8a. This is because a missed uplink will cause the number of free storage units to reach the missed-pass risk number at the start of the next telecom session, see Table 3, so there is no intervening telecom session during which to take any action. 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 to delete data set n-2, we can transmit commands with the backup.

At this point in our mission, we must use a 70-meter antenna to support our downlink rate of 2.2 Mb/s. To make the uplink

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.

We must provide both uplink and downlink mitigation when we have a red alarm in the missed-pass risk column. We can provide the uplink mitigation in either of the two ways described above. We also have two ways to provide

a)				b)			
	Missed-Pass Risk	Uplink Mitigation	Downlink Mitigation		Missed-Pass Risk	Uplink Mitigation	Downlink Mitigation
n-2	No alarm			n-2	No alarm	Pre-emptive delete	
n-1	No alarm	Ground antenna		n-1	No alarm		
n	Yellow			n	Yellow		
n+1	No alarm			n+1	No alarm		
n+2	No alarm			n+2	No alarm		

c)				d)			
	Missed-Pass Risk	Uplink Mitigation	Downlink Mitigation		Missed-Pass Risk	Uplink Mitigation	Downlink Mitigation
n-2	No alarm		Ground antenna	n-2	No alarm	Pre-emptive delete	Ground antenna
n-1	No alarm	Ground antenna		n-1	No alarm		
n	Red			n	Red		
n+1	No alarm			n+1	No alarm		
n+2	No alarm			n+2	No alarm		

e)				f)			
	Missed-Pass Risk	Uplink Mitigation	Downlink Mitigation		Missed-Pass Risk	Uplink Mitigation	Downlink Mitigation
n-2	No alarm			n-2	No alarm	Pre-emptive delete	
n-1	No alarm	Ground antenna	Contingency pre-emptive delete	n-1	No alarm		Contingency pre-emptive delete
n	Red			n	Red		
n+1	No alarm			n+1	No alarm		
n+2	No alarm			n+2	No alarm		

**Table 8. Options to mitigate single yellow and red alarms in the missed-pass risk column.**

downlink mitigation, for four combinations. The simplest is shown in Table 8c. In this scenario, we provide uplink mitigation with a backup ground antenna during telecom session n-1 and another backup ground antenna for downlink mitigation during telecom session n-2. The idea is to give us two chances to receive data set n-2 with the backup ground antenna during telecom session n-2, and two chances to delete data set n-2 with the backup ground antenna during telecom session n-1. We can also use a backup ground antenna for downlink mitigation during telecom session n-2 along with a pre-emptive delete before for uplink mitigation. See Table 8d. Since it is not always possible to get a backup ground antenna that is capable of supporting our nominal 2.2 Mb/s downlink rate, we have contingency plans to lower our downlink bit rate via real-time commands. This allows us to receive at least some of the data. The vast majority of the time, we can receive enough so that during telecom session n-1, we will be able to delete enough so that at the start of telecom session n, the number of free storage units is above the yellow limit. This mitigation approach relies on graceful degradation rather than a full-capability backup.

If it is not possible to get a backup ground antenna during telecom session n-2, the downlink can instead be mitigated with a contingency pre-emptive delete command during telecom session n-1. See Table 8e and Table 8f. This contingency pre-emptive delete command would only be sent if there were actually a problem that interfered

with the receipt of data during telecom session n-2. In this case, we would send real-time commands during telecom session n-1 to retransmit the data we missed during telecom session n-1 and then send the contingency command to delete it. The point is to make sure that at the start of telecom session n, that the number of free storage units is at least above the red limit if not above the yellow. This protects us against an under-estimation of data volume that would otherwise cause us to fill the on-board storage. Like the regular pre-emptive delete commands used for uplink mitigation, the design of the contingency pre-emptive delete commands is tricky and relies on detailed knowledge of the time and order in which the observatory transmits data.

The contingency pre-emptive delete command can be combined with either method of uplink mitigation as shown in Table 8e and Table 8f. In the case of Table 8f, though, if there were a downlink problem during telecom session n-2, we would probably not send the regular pre-emptive delete command during that session. In effect, the contingency pre-emptive delete command during telecom session n-1 gives us a second chance to delete the data that we were unable to delete during telecom session n-2.

#### **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 in the downlink system. It also lets us determine how and when to mitigate risk if we are unable to maintain single-fault tolerance in our design. 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 (MIPS) 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 to 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. 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. 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.  $75,875 = 120,875 - 3 \times 15,000$ ). 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. But, 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 a significant amount of time 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 teardown time required by the ground stations.

A proposal to change the spacing of telecom passes to thirty-six hours was rejected because it neither allowed enough time for the uplink of sequences nor 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 one-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. 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. 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. The asymmetry of the missed-pass risk was one consideration in deciding not to implement asymmetric telecom-session spacing.

#### **F. Fill-Avoidance for the On-Board Storage**

As of this writing, the fill-avoidance patch will take effect in a few weeks. We have already revised our rules, as described above, for risk mitigation. The implementation of the fill-avoidance patch allows us to remain single-fault tolerant with respect to filling the on-board storage while scheduling larger data volumes. It does this by removing the single-fault tolerance with respect to skipping observations. The proposal and design for the fill-avoidance check system came from the Spitzer Science Center. A single, or multiple, fault that might have caused the on-board storage to fill now might only cause some observations to be skipped. It will still be possible for us to fill the on-board storage. There is no fill-avoidance check before engineering or housekeeping data are written to the on-board storage. Certain critical events, like instrument power-ons, power-offs, calibrations and anneals, are also excluded from the fill-avoidance check. However, their data volumes are small, and do not violate our single-fault tolerance with respect to filling the on-board storage.

Before each science observation is started, the sequence will check to make sure that there is enough free space remaining in the on-board storage. The minimum amount required is the sum of the actual data-volume estimate for the observation, plus constants to account for uncertainty in the data volume estimate, engineering data volume that remains to be written before the next telecom session and a bare-minimum number of free storage units below which we do not want to allow the on-board storage to go. These last three constants total 7,000 storage units. We would like to change the constants for the uncertainty in the data volume estimate and the amount of engineering data into values that are a function of time. The constants we use now are, by necessity, overestimates. Under some circumstances, they might cause science observations to be skipped that would not have filled the on-board storage.

We will eliminate the current yellow- and red-alarm limits and replace them with a single red-alarm limit of 7,000. In the future, mitigation measures during sequence generation that before were required will be optional. We will choose to implement them if the cost of rescheduling skipped observations would be high. For example, sometimes missing a single observation that is part of a larger group would significantly reduce the science value of the rest of the group of observations. These determinations will be viewed as trade offs to be made by science rather than mission operations. This fill-avoidance patch may allow us to re-visit our strategy for the frequency and spacing of telecom sessions.

### **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 predict the future state from that history. We have built an on-board storage prediction tool 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 are 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 Fig. 1, Fig. 2 and Fig. 3.

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

The on-board storage prediction tool is divided into several pages. All the primary pages calculate the same set of information, but start with different values as inputs. That is, sometimes the tool calculates data volumes from telemetry values and sometimes telemetry values from data volumes. In addition to the primary pages, there are pages that calculate variances. The variance pages take the differences between pairs of primary pages. As shown in Fig. 1, Fig. 2 and Fig. 3, the variance is always calculated as actual value minus predicted value. In addition to the primary pages and the variance pages, there are two supplemental pages. The supplemental pages compile key pieces of information from the other pages and show them in a summary 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 9.

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 of the on-board storage 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 abbreviated.

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	Data volume written (during this period of autonomous operation) to on-board storage
	Data volume freed (during this period of autonomous operation) by acknowledgement
	Data volume freed (during this period of autonomous operation) 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 (total since reboot) discarded by acknowledgement
	Storage units (total since reboot) discarded by fault protection
Actions during sessions	Data volume of new data transmitted
	Data volume of old data retransmitted
	Data volume of new data received on ground
	Data volume of old data received on ground
	Data volume written (during this session) to on-board storage
	Data volume freed (during this session) by acknowledgement
	Data volume freed (during this session) by fault protection
State at end of session	Same as at start of session

**Table 9. Categories and columns of information contained in the on-board storage prediction tool.**

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 times 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.

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 that no storage units ever need to 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 that the on-board storage is empty.

The Actual page takes actual telemetry values, with any optional overlays, 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 performs its calculations differently for telecom sessions that have already occurred and those that will occur 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 performs its calculations differently for telecom sessions that have already occurred and those that will occur 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, as it is currently implemented, for future telecom sessions. We may decide to remove this page in the future.

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 as the mechanism to calculate how many storage units of backlog there are at each transition in Fig. 1, Fig. 2 and Fig. 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. It was intended to show the extent of the backlog caused when telecom session were greater than one-hundred percent full. However, like the Perfect Predict page, it does not do a good job for future telecom passes.

The Summary page displays together 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.

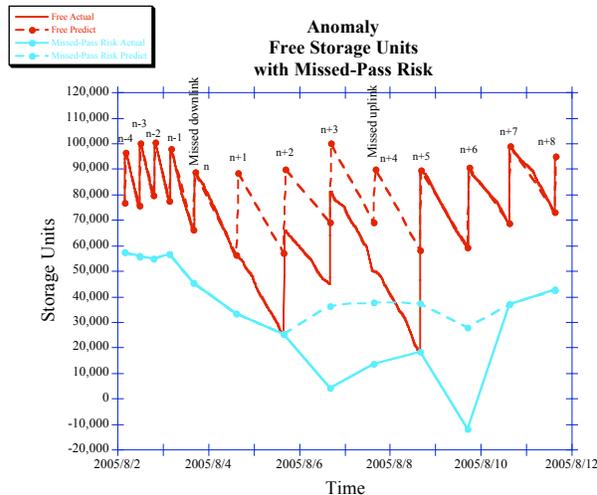
The Mitigation page gathers the catalog and identifying information, the measures of risk and the data volumes from the Extrapolation page. Then it determines which telecom sessions need mitigation and redundancy according to the rules set forth in Table 8. Further, it calculates the volume of data that would need to be deleted by a pre-emptive delete to raise the missed-pass risk above the yellow-alarm limit value.

## **H. 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 outputs of the tool are 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. There are thirteen telecom sessions and twelve intervening periods of autonomous operation. 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 before the fact. The solid red line is continuous data. The dashed red line and both blue lines are only calculated at the beginning and end of telecom sessions, represented by the solid dots.

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. However, our uplink capability was unaffected and we were able to send the commands to acknowledge the data received during session  $n-1$ . Hence, the number of free storage units rose as predicted during session  $n$ . Because we had no downlink during session  $n$ , we had to wait until session  $n+1$  to confirm this. During session  $n+1$ , though, we have no acknowledgement commands to send because we missed the



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

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. The missed-pass risk is re-calculated after each telecom session. However, Fig. 8 does not show all these intermediate versions. If we had missed the uplink during session  $n+5$  (or equivalently the downlink is session  $n+4$ ), 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. We were able to acknowledge the storage units received during sessions  $n+3$  and  $n+4$ .

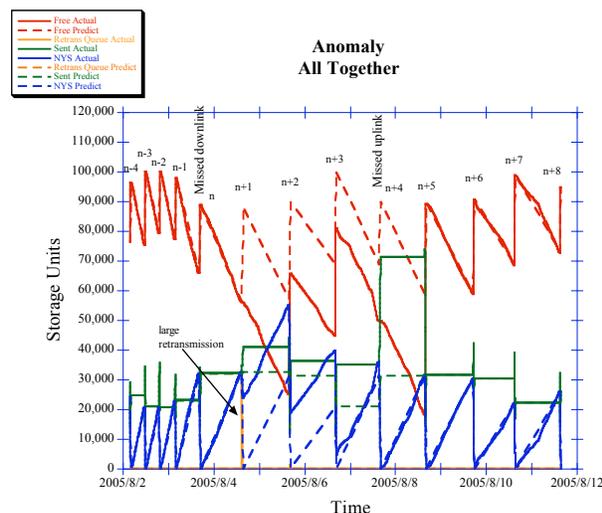
By pure co-incidence, during session  $n+4$  we received the last of the backlogged not-yet-sent storage units. So, at the end of session  $n+5$ , the actual number of free storage units again agreed with the predicted number. During session  $n+5$ , we recovered simultaneously from the missed downlink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

We can view the four primary telemetry values, both predicted and actual, together on one graph. See Fig. 9. The missed-pass risk and marker dots have been omitted from this plot for clarity. 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

previous downlink. The number of free storage units does not rise during that session. So, at the start of session  $n+2$ , the number of free storage units bottomed 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 uplink during session  $n+1$  and causes the free storage units to reach a minimum during session  $n+2$ .

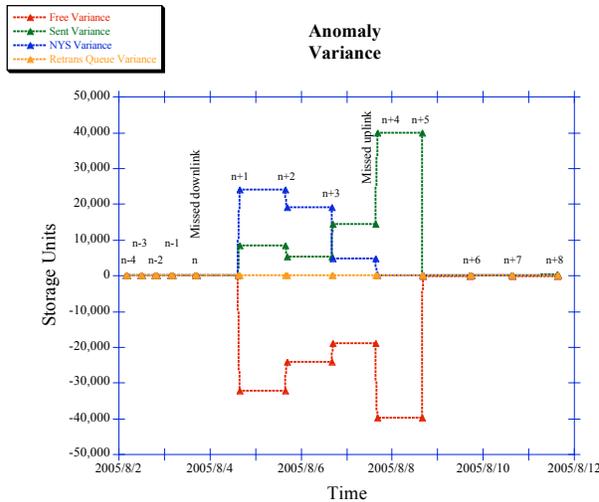
During telecom sessions  $n+2$  and  $n+3$ , we can see that the actual number of free storage units is gradually returning to the predicted values. Although it is not visible looking only at Fig. 8, with session  $n+1$ , we began to retransmit the missed data and clear the backlog. We were able to transmit more storage units than predicted starting with session  $n+1$  because there was margin built into the original session, and we increased the durations of

the sessions via real-time commands.



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

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



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

telecom session. Figure 11 shows the number of new and retransmitted storage units actually received compared to the numbers 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. This is true 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. The fact that the number of actual retransmitted storage units received during telecom session n+1 is almost equal to the number of predicted new storage units is purely co-incidental.

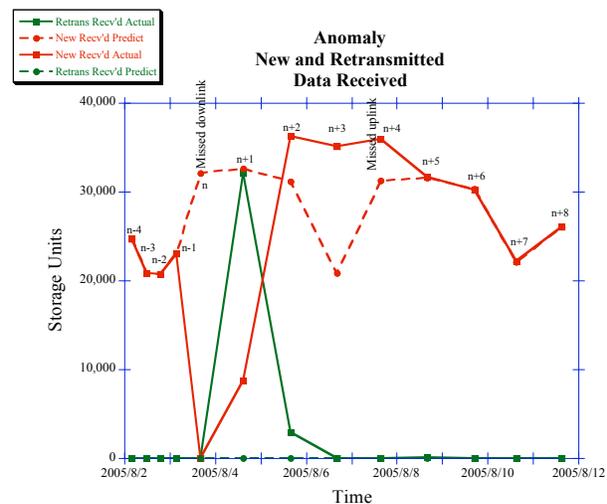
We can see the same pattern in Fig. 12. Figure 12 shows the variances of new and retransmitted storage units received over the course of this anomaly. During session n, the new storage units showed a large negative variance. Then, during session n+1, the retransmitted storage units showed a large positive 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. In addition, we can see that Fig. 11 and Fig. 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.

### I. Recovery Strategies

We have developed many operational strategies and techniques to recover from anomalies that affect the downlink system. The most important things during any recovery are:

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. We can look at this in an easier 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. That is, we are only showing the variances due to the anomaly and not including those due to errors in data-volume prediction. Notice that the variances are calculated only at the start and end of the telecom sessions. The variance for number of storage units in the retransmit queue in this example is always zero. This is because the large retransmission during session n+1 completed before the end of the session. If that retransmission had not finished in session n+1 but carried over into session n+2, then a variance would be visible.

We can also do a similar analysis for the number of storage units received on the ground during each



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

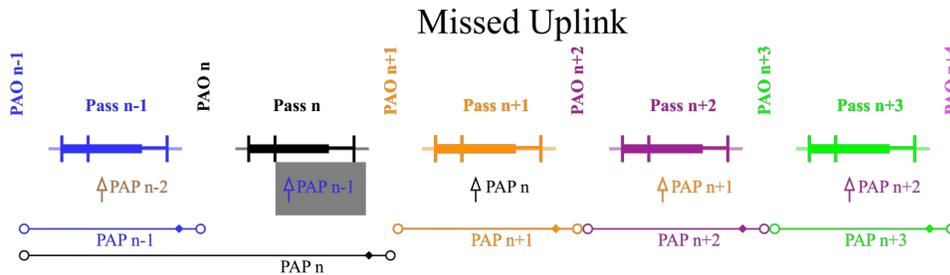
- 1) to determine if and when we will fill the on-board storage assuming that there are no further anomalies,
- 2) to start the retransmission of any missed data during the next telecom session,
- 3) to lengthen the time available for downlink in each session until the transmit backlog is cleared,
- 4) to predict how long it will take to clear the backlog,
- 5) to determine if we have lost single-fault tolerance (i.e. revised missed-pass risk is greater than zero) and when we will recover it again,
- 6) if there is a backlog of data to transmit, to 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 do not always follow the sixth rule above since building such commands is complex and we run the risk of deleting data we have never received. We have outlined ten to fifteen detailed scenarios for single failures.

An example diagram is shown in Fig. 13. In this diagram, we use the acronym for Period of Autonomous Operation (PAO), Pass means telecom session, and Packet Acknowledgement Process (PAP) is the software we use to build commands to acknowledge data on-board. The upper series of horizontal lines represent the telecom sessions. The first and last perpendicular lines indicate the beginning and end of the telecom session. The thick line represents how long it takes to transmit the data from the on-board storage. The vertical arrows indicate when the data-acknowledgement commands are sent. The long horizontal lines with the circles indicate the receive-time range covered by each PAP command.

In this scenario, we are unable to uplink the PAP commands during Pass n, PAP n-1 (the blue arrow in the grey box). However, we did receive the downlink data during that pass. So, between passes n and n+1, we build a single PAP command that covers the data received during both pass n-1 and pass n, PAP n (the black arrow). Then, during pass n+1, we will send the PAP n. We will never send the PAP n-1 command.

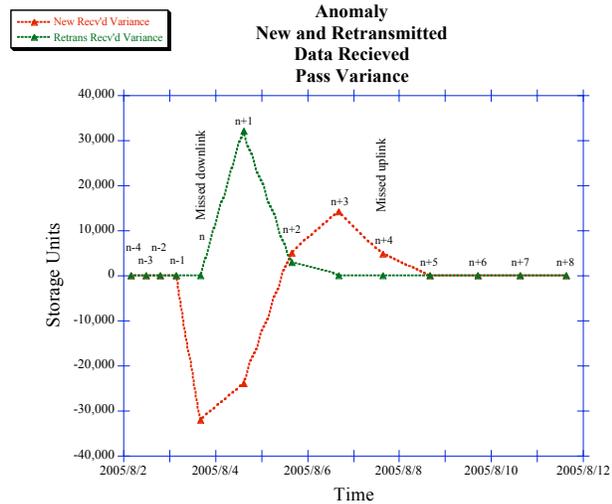
One of these primary problems we face is that of the end gap. An end gap occurs when we do not receive the last portion of data of a given APID transmitted by the observatory. The ground software that generates the acknowledgement and retransmission commands will not request retransmission of data unless 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.



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

The only time that the problem would remain is in the case of a completely missed downlink.

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.



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

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

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 of 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 our nominal mission and it worked well. However, it has several shortcomings that will probably limit its use in the future. We require 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 takes too long. 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 usually greatest during the one or two telecom sessions following the anomaly. The main benefit of delaying the end of the telecom session 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, even though we gain less time per session.

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 packets 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 all such gappy data to have been missed entirely. 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 sixth 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.

## **J. 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 now 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 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 predictions 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, a small number of storage units were put in a protected state and taken out of regular use. This was due to

a feature of the flight software. 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 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.

This tool has also made it possible to train other members of the project in the analysis of and response to anomalies of the downlink system. Before we built the tool, there had been only one member of the project who was so qualified. The project had in fact assigned itself an action to conduct the cross training to mitigate this single-point-failure risk item. The tool and the analysis and design that went into building it, made the training possible.

## **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 may create a backlog and eventually fill the on-board storage.

We would like the tools to more simply support 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 this 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.

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 details of 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 a procedure 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 per telecom session, we could 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 until inputs from the following telecom session are made. 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. However, 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 placed 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.

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. This might involve modeling the telemetry at the start and end of each science observation and by individual APID. 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 us 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.

The fill-avoidance patch will change the way we mitigate risk in the downlink system. However, it will not completely obviate the need to do the analysis and monitoring of the on-board storage. We are willing to accept the risk of skipping some observations. However, we do not want to guarantee that we will skip observations in case of a single failure. The fill-avoidance patch is a safety net that we hope we never have to use. Therefore, we will still do the analysis during sequence generation and the monitoring during sequence execution. Especially during anomaly recovery, we must rely on the analysis tool to tell us the best way to recover quickly and how long it will take us to clear the backlogged data. It will also tell us when we will regain our single-fault tolerance. Until we do so, a second failure would almost certainly cause us to skip an observation. We may also be able to glean insight into which observations are most at risk to be skipped, although that still currently requires manual analysis.

The primary reason that we delete data during the telecom session following the one in which they were received is that the data lines from the ground antennas are not large enough, and introduce too much latency, to allow us to receive the data, and build and send the acknowledgement commands quickly enough. In the next several months, the capacity of those data lines will be increased. We hope that this will be sufficient for us to send the acknowledgement commands in the same pass. We will likely only be able to do this for the first one-quarter to one-third of the data received during a given pass. However, in an anomaly situation, that would be significant and would help us reduce the risk of filling from either under estimation of data volume or a second failure.

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. The first capability is scheduled to become available in the next few weeks. The second capability is currently not possible, but may become so as the network bandwidth of our ground data system is improved over the next few months. 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.

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 overflow 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, either automatic or by-request. 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. As long as the data volumes, the duration of storage and transmission needed, the analysis we describe in this paper can be put to good use.

Missions that are science-driven, more specifically, those that are designed to take advantage of one-time critical science events, must design into their science collection and downlink processes the capability to identify, prioritize, and retransmit high-priority data. This paper shows the complexity that is inherent in such mission designs, and

provides some insights on how carefully mission designers must consider the resources that need to be managed and the tools the operations teams need to use in order to increase the success of such missions.

### **Acknowledgments**

The authors would like to thank the following people all of whom have contributed to the success of the software and processes described in this paper: Dipak Achhanni, Magdy Bareh, Ronald Beck, Lee Bennett, Carole Boyles, Robert Bunker, Sean Carey, Wendy Casiano, Joe Diep, Wayne Evenson, Mark Fisher, Paul Fleming, Michael Gallagher, Carol Glazer, David Hirsch, Henry Hotz, Wallace Hu, John Ibanez, Michael Jones, Bolinda Kahr, Joseph Kahr, Helenann Kwong-Fu, Deborah Levine, Robert Lineaweaver, Patricia Lock, William Mahoney, Jeanne Makihara, Sunny Mar, Angus McMechen, David Mittman, Bobby Naron, Michael Ochs, Victoria Scarffe, Charles Scott, Stuart Spath, Kennis Stowers, Stanley Thompson, Freia Weisner, Robert Wendlandt, Robert K. Wilson and Joshua Wood.

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