Designing Command and Telemetry Systems Using MIL-STD-1553 and CCSDS

Ronnie Killough
Southwest Research Institute
6220 Culebra Rd. San Antonio TX 78238
210-522-3616 rkillough@swri.org

Michael McLelland
Southwest Research Institute
6220 Culebra Rd. San Antonio TX 78238
210-522-3360 mmclelland@swri.org

Abstract. The use of the MIL-STD-1553B (1553) communications bus and Consultative Committee for Space Data Systems (CCSDS) standards are becoming increasingly popular in the design of small satellite command and data handling systems. Experience has been gained in the use of these two standards on a number of missions, which could be of benefit to those faced with integrating them into new spacecraft.

The 1553 bus presents a number of advantages and disadvantages, both from electrical and data protocol perspectives. The 1553 bus is sometimes specified in the design of systems without considering these issues and without understanding the fundamental characteristics of the 1553 bus. The historical use of the 1553 bus provides important insights into these characteristics. The pros and cons of the 1553 bus must be considered when specifying the communications bus for a satellite. The use of the 1553 bus is often weighed against other communication protocols such as RS-422 in the design of command and data handling systems.

The CCSDS standards are now maturing and are realizing increased use in explorer-class satellites. In addition to spacecraft-to-ground communications, CCSDS data units are being used as the format for the communication of commands, messages, and telemetry data among the instruments and the spacecraft controller. Because of the common use of the 1553 bus, techniques for implementing CCSDS over the 1553 bus must be derived. Experience has produced a list of "dos and don'ts" in implementing 1553 bus command and telemetry protocols. A number of missions have developed successful strategies for integrating CCSDS and the 1553 bus.

Background

The endeavor to produce small satellites under the "faster-better-cheaper" paradigm has led to the design of loosely coupled, "open systems" architectures in which the spacecraft controller maintains a simple standards-based command and data handling (C&DH) interface to the payload (in this paper, *payload* refers to the suite of instruments onboard the satellite, rather than the satellite itself as would be the case in the context of a launch vehicle). In some architectures, the spacecraft controller communicates directly with each instrument controller on the payload as shown in Figure 1-A. Alternatively, the spacecraft controller may communicate with a payload controller, which in turn is responsible for communicating with each of the instruments. This architecture is illustrated in Figure 1-B. Regardless of the architecture, the interface protocol should be kept as simple as possible.

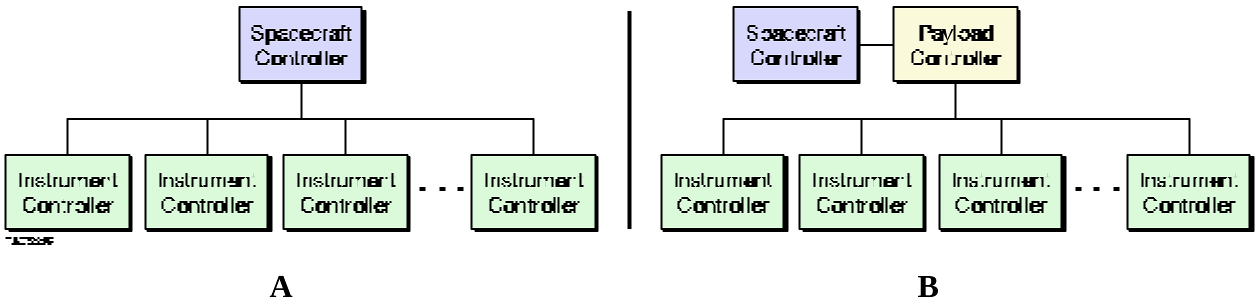

Figure 1 - Alternative Spacecraft Architectures

Open Systems Advantages

The advantages of an open systems, standards-based design include:

(a) Reduced program risk by building on top of existing and proven standards. The design and development of custom electrical interfaces and application-layer protocols is time consuming and risk-laden.

(b) The C&DH interface between the spacecraft and the science payload is simple enough that it becomes reasonable to capture all of the interfaces between the spacecraft and the payload (i.e. C&DH, mechanical, thermal, electrical) in a single interface control document (ICD). Keeping the number of individual documents to a minimum (but appropriate) set is an obvious advantage in a faster-better-cheaper program.

(c) Allows the development of the spacecraft and payload to proceed in parallel since they are not tightly integrated. This provides necessary schedule overlap which is the key to developing sophisticated observatories on a small satellite schedule.

(d) As a corollary to (c), the open systems approach facilitates the reuse of existing spacecraft bus designs across a wide variety of missions.

(e) The simplicity of the C&DH interface makes it reasonable, even in such short development schedule missions, to develop (and in some cases, rework existing) payload and spacecraft simulators to facilitate pre-observatory integration testing.

(f) Finally, integration of the spacecraft and the payload is expedited due to the simplicity of the interface between them. The use of interface simulators significantly reduces the amount of time required for observatory integration, and reduces the risk of finding interface misunderstandings late in the game.

Open-Systems Standards

The use of the MIL-STD-1553[1] (1553) communications bus and Consultative Committee for Space Data Systems[2] (CCSDS) standards are becoming increasingly popular in the design of small satellite C&DH systems. The 1553 bus has distinct advantages from mechanical, electrical, and reliability standpoints, but presents some challenges when using the interface for the transmission of streaming science data. CCSDS is attractive because it is not only becoming the defacto standard for communicating between spacecraft and ground systems, but is also easily extended for use in implementing the C&DH interface between the spacecraft and the payload.

In spite of these advantages, the system engineer should consider the pros and cons of the 1553 bus before selecting it in the design of a new mission, and should take advantage of the experience of other missions in applying CCSDS to 1553 when those selections are made. The historical foundation of the 1553 bus provides important insights into the characteristics of the 1553 bus.

History of MIL-STD-1553

The MIL-STD-1553 was first issued on August 30, 1973. Revision A to the standard was issued on April 30, 1975. Revision B was issued on September 21, 1978 and, in conjunction with Notices 1 and 2, is the standard referenced by most current applications. However, Revision B is actually up to Notice 4. Table 1, provided for reference, summarizes the differences between the base MIL-STD-1553B and the various Notices.

1553 Basics

The basic structure of the 1553 bus is illustrated in Figure 2.

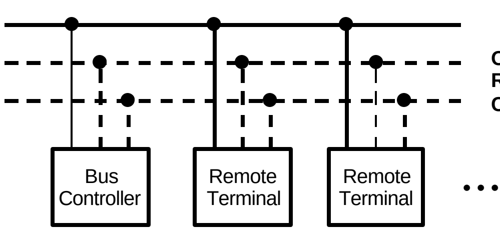

Figure 2 – 1553 Bus Structure

The bus consists of a single Bus Controller (BC) connected to a maximum of thirty-one Remote Terminals (RT). Both the BC and the RT have local controller memory spaces which are divided into thirty subaddresses (SA). Each SA is sized to contain a maximum of thirty-two 16-bit words. The BC initiates all bus transactions, and the RT responds to those commands as a slave. The primary commands include the Transmit Data command and the Receive Data command. The Transmit Data command causes data to be transmitted from a specified RT/SA to the BC. The Receive Data command causes data to be transmitted from the BC to a specified RT/SA. The transmission bit rate on the bus is one megabit per second. While there are other aspects to the 1553 bus operation, this represents the basic structure and operation of the bus.

Table 1 – Summary of 1553 Revisions

Revision Issue Date	Summary of Changes
Revision B Sept. 21, 1978	Defined implementation options to enhance compatibility between designs by different manufacturers.
Notice 1 Feb. 12, 1980	Specified certain options to be required in Air Force avionics, and restricted others.
Notice 2 Sept. 8, 1986	Title change. Notice 1 changes largely replaced by other specifications designed to enhance tri-service (Army, Navy, Air Force) interoperability. Defined certain required mode codes, introduced data wraparound test, and clarified use of busy bit.
Notice 3 Jan. 31, 1993	Administrative notice validating 1553B for use in acquisition.
Notice 4 Jan. 15, 1996	Title change only.

1553 Characteristics

It is insightful to study the titles of the various revisions of the MIL-STD-1553, as they reveal the original intended use for the bus, as well as other characteristics which are important to understand when applying the standard to command and data handling systems. The original title for the MIL-STD-1553B was “Military Standard Aircraft Internal Time Division Command/Response Multiplex Data Bus”. This title reveals the following:

(a) The bus was originally designed for use in military aircraft.

(b) The bus is fundamentally a shared, time division multiplex (TDM) interface.

(c) The bus operates on a command and response basis.

The author of the preface to the *MIL-STD-1553 Designer’s Guide*[3] notes that “MIL-STD-1553 has become the standard its proponents had hoped it would be.” This is probably an understatement, as the standard has gone beyond its original aircraft avionics application into other uses, most notably (in the context of this paper) the design of space data systems. This expansion in utility was recognized in MIL-STD-1553B Notice 2, when the title of the standard was changed to “Military Standard Digital Time Division Command/Response Multiplex Data Bus”. In this change, the words “Aircraft Internal” were dropped, in favor of the in-vogue term “Digital”. Notice 4 has further altered the title by replacing “Military Standard” with “Interface Standard”. It is certainly clear, then, that 1553 has ventured outside both its original purpose (aircraft) and its original domain (military). However, the implications of its role in spacecraft data systems are not immediately obvious to the uninitiated.

Implications of Aircraft Heritage

The design of the 1553 bus was evidently driven primarily by transmission reliability, response time, and maintaining central bus control, as opposed to the simple and efficient transmission of large quantities of data. This point is relevant, since the primary function of a spacecraft data system (for a scientific spacecraft) is the transmission of substantial quantities of science data. Science data is characteristically streaming data, but the 1553 bus is not a streaming interface. Rather, messages traversing the 1553 interface consist of a multiple of 16-bit data words, with a maximum single-command transmission size of 32 words.

The design of the application data interface protocol for a given mission must take this into account and determine a mechanism by which streaming data (or, stated another way, variable-length packets) can be transmitted in fixed-size segments. Examples of how this can be done in various spacecraft configurations are presented later.

Implications of TDM Design

Perhaps the most common mistake made when initially designing a 1553 data protocol is the failure to recognize that the 1553 bus is a time division multiplex data bus. Designing an application-level protocol which ignores this will result in many head-scratching hours in the laboratory when the protocol is found to be unreliable. The authors can personally attest to this, as an asynchronous handshaking protocol was designed for an early version of a space station experiment control unit. The protocol proved unreliable, but a detailed peer review of the protocol revealed no design flaws. The failure mode was such that, in the midst of a data transmission, the BC and the RT could become deadlocked – each waiting on the other. Results of an analysis of the problem suggested that the BC would read a given data set from the RT, but the RT controller would fail to “see” it occur. A query to the local controller by the RT would result in incorrect status information being reported, establishing the deadlock at the application layer. It was further speculated that the condition was caused when the RT happened to query its local controller coincidentally with a Transmit Data command issued by the BC. The 1553 controller manufacturer was contacted, who responded that, in essence, 1553 was never designed for this type of asynchronous protocol, and that the controller itself does nothing to prevent simultaneous access by the RT and the BC.

A second manifestation of this problem occurs when a BC reads a set of data from a given RT/SA when the RT is simultaneously in the midst of updating the data in that same SA. The result is the BC receives a data set containing partially new and partially stale data (refer to the next subsection for additional discussion regarding stale data detection). This results in confusion in the data stream, and if not detected could result in incorrect science data being transmitted to the ground, or perhaps a garbled message to the spacecraft. This is one motivation for including some kind of checksum in data segments transmitted across the 1553 bus (in the RT-to-BC direction). Often, a checksum is not seen as necessary due to the reliability of the 1553 bus. However, without it the scenario just described may go undetected.

It is imperative in the design of 1553-based communication protocols that a mechanism be established to minimize the possibility of BC/RT "controller collisions", and ensure that the protocol can detect and recover in the event a collision does occur.

The TDM design of the 1553 bus has an additional implication. The 1553 bus is a shared bus; it is not possible to communicate with more than one instrument (or other subsystem) simultaneously. When a spacecraft design employs a dedicated bus to each instrument (such as RS-422), it is possible for each instrument to transmit its data to the spacecraft without regard to whether other instruments are also transmitting (obviously limited by the capacity of the receiver). This allows the designer of the protocol to focus most of the effort on message types and formats. However, in the case of a 1553-based C&DH system, the protocol becomes more difficult to specify since the designer must consider the peculiarities of each of the RTs (which could include instruments, star trackers, sun sensors, solid state recorders, etc.), system priorities, and total bus loading in addition to the basic elements of the data protocol itself. While all of these issues must be addressed in some form regardless of the bus type, the more information that must be included in an ICD the longer the design will take and the more difficult it will be to get agreement from the various participants.

Implications of Command/Response Design

As previously described, the 1553 bus operates in such a way that all bus transactions are centrally controlled by a single BC. When considered in conjunction with the original domain of aircraft avionics, it is clear that the BC (corresponding to the master avionics component) wants to be able to poll the RTs (corresponding to the various remote avionics components distributed throughout the aircraft) and get their *current* status at any time. The implication is that each RT must maintain its current status in its local controller memory at all times for ready access by the BC. The design of the 1553 bus is such that the BC will receive the exact same data each time it queries an RT/SA, until that RT places new data in that SA. This works quite well in the scenario described, but does not work well when attempting to transmit streaming data. In a serial interface such as RS-422, once a set of data has been transmitted, the same data will not be seen again by the receiver (unless the application specifically resends it). Therefore, a 1553-based application data protocol must include a provision for the BC to detect "stale" data, to avoid confusion in the data stream.

A second implication of the command/ response design (which was addressed from a different perspective in a prior section) is that an RT has no control over when data is transmitted. This is controlled exclusively by the BC, and the convention for when it occurs must be established as part of the application data protocol ICD. Therefore, it is not sufficient for an instrument to just send data at will, limited only by the speed of the interface and the size of its local transmit buffer. Rather, the RT must be keenly aware of when the BC will poll it and when the BC has successfully retrieved a given set of data, so that the RT can safely place a new data set in the controller memory.

It should be noted that the 1553 bus does not suffer from the same pitfalls in the BC to RT direction as it does in the RT to BC direction. For one, in a satellite C&DH system the amount of data transmitted in the BC to RT direction is typically very small (relatively low rate commands) as compared to the RT to BC direction (relatively high rate telemetry). Second, since the BC initiates all bus transactions, the issues of stale data and data transfer synchronization are diminished. It is sufficient that the protocol define the maximum frequency at which the BC will send commands to each RT, and to ascertain that each RT is able to keep up with the specified rate. Finally, there are also issues regarding message error handling and bus retries, but in the interest of brevity these issues are not addressed here.

Motivation For Use of MIL-STD-1553

With all the pitfalls of using 1553 in the C&DH domain, why use it at all? The answer lies in the electrical characteristics of 1553. 1553 is a robust, fail-safe bus that has provisions for redundancy, which are desirable traits in a spacecraft data bus.

Because of the dirty electrical environment characteristic of avionics coupled with the need for high-reliability, the 1553 bus was designed to be robust from a data transmission viewpoint. Even though the 1553 specification only makes provisions for parity error detection, the 1553 bus boasts a bit error rate of 10^{-12} from a practical multiplex system built to MIL-STD-1553B. This is accomplished through the use of a Manchester Encoded signal distributed over a twisted shielded pair at a one megabit-per-second rate. This level of bit error rate increases overall system throughput since minimum bus bandwidth is utilized for retransmissions. These same features make 1553 attractive for space systems as well, due to the need for reliable transfers in a high radiation environment.

A second and perhaps most significant characteristic is the fail-safe features built into the 1553 bus. This is accomplished through galvanic isolation and a fail-safe timer. Each node on the common medium is galvanic ally isolated from the other. Because of this, a catastrophic failure in one subsystem on the bus does not lead to complete bus shutdown as seen in other bus structures such as RS-485. This allows the entire system to gracefully degrade over the life of the mission. This is especially important in science missions where the loss of one instrument, while unfortunate, does not automatically result in complete mission loss. This isolation is complemented by the fact the 1553 specification requires all transmitters on the bus to provide a terminal fail-safe timer. This is a hardware-implemented time-out that precludes a signal transmission of longer than 800 microseconds. This prevents any single transmitter from dominating the bus with chatter.

Finally, the 1553 standard makes provisions for a completely redundant bus and/or a backup bus master. Because this feature is provided for in the specification, standard chip sets from several vendors are available to systems designers that enable redundancy to be implemented without placing undo burden on the board designer.

Overview of CCSDS

While 1553 is a common hardware selection for C&DH interfaces, CCSDS is achieving dominance as the basis for telemetry and telecommand communication protocols. While CCSDS is most readily viewed as a set of standards for the transmission of data between a spacecraft and a ground system, its utility easily extends to the implementation of interfaces between spacecraft subsystems. Hence, methods for implementing CCSDS-based spacecraft-to-payload application data protocols over 1553 are needed. Before exploring this combination in more detail, a brief overview of some of the more commonly employed aspects of CCSDS are presented.

The CCSDS "standards" are actually not standards at all, but rather recommendations generated by an international committee. However, many of the recommendations have become International Organization for Standardization (ISO) standards, and so this paper frequently refers to "CCSDS recommendations" as "CCSDS standards". The basic elements of the CCSDS standards consist of formats for telecommand[4] and telemetry packets and frames[5]. CCSDS also proposes standards for items such as time formats[6] and data compression[7] algorithms. When dealing with end-to-end protocols (spacecraft to ground to end user), additional concepts are needed to ensure reliable transfer such as data encoding and telecommand verification protocols. These concepts are also addressed by CCSDS. The CCSDS telemetry and telecommand standard is patterned after the familiar ISO layered network model[8]. However, in the design of C&DH protocols within the confines of the spacecraft itself, the packetization and segmentation layers are paramount and, in some applications, portions of the transfer layer as well.

The CCSDS standards are precisely defined and carefully configuration controlled. The exactitude of the specifications ensures consistency in interpretation, but requires large numbers of pages to convey relatively simple concepts and straightforward formats. Combined with the array of acronyms and terminology, the new reader is easily intimidated. In the process of producing new versions of existing standards, and new recommendations to accommodate advanced applications of those standards, the terminology seems to swell further, such that multiple terms are sometimes used to refer to essentially the same entity. This adds to the difficulty in selecting the appropriate terminology to use when specifying CCSDS elements in a mission ICD. Without going into excessive detail, the following sections introduce some of the basic elements of CCSDS and make reference to the more common alternate terms.

Packetization Layer

The packetization layer essentially consists of the CCSDS Source Packet. Figure 3 illustrates the format of the Source Packet.

Primary Header	***Secondary Header***	***Packet Data***
Version # & APID Sequence Control Packet Length	*Optional* Timestamp Format Info.	User Data
4 Bytes	***User-Defined***	***Variable***

Figure 3 – Source Packet Format

Rather than define each of the packet fields here, the reader is referred to the CCSDS documents[4, 5, 9]. These documents and others are available at http://ccsds.org. The Source Packet contains a standard primary header, an optional user-defined secondary header, and the application data. Source Packet types for a given mission are distinguished from one another by unique Application Process Identifiers (APIDs). The Source Packet can be fixed or variable length. The standard supports a maximum packet size of 65540 bytes (a four byte primary header plus up to 65536 bytes of secondary header and source data). However, the maximum length for any given application is mission-specific. Longer packets provide the advantage of reduced overhead; however shorter packets are often preferred since less data is lost if a single packet is dropped. The CCSDS formats were designed with efficiency in mind[8], and so the various elements of the header are not always word or even byte aligned. The Source Packet is often referred to as a Telemetry Packet, or simply Packet. In the Advanced Orbiting Systems (AOS) standard[9] the Source Packet is referred to as the CCSDS Path Protocol Data Unit (CP_PDU), although the version number of the packet format remains at Version 1.

The Telecommand Packet is identical in format to the Source Packet. In the AOS standard, which was designed to address the special needs of more sophisticated systems such as satellite constellations and space stations, it is recognized that the distinction between a telemetry packet and a telecommand packet becomes less obvious as messages are exchanged between orbiting spacecraft, or between the various subsystems of larger spacecraft such as the space station.

Segmentation Layer

The packetization layer facilitates the creation of variable-length entities. However, not all interfaces support the transmission of a variable-length packet and so the packet must be segmented into fixed-size pieces. The primary header of the CCSDS Source Packet includes the necessary control bits (the Grouping Flags plus the Source Sequence Count) to support the segmentation of packets. The concept of segmentation is illustrated in Figure 4. Packet segmentation is one (but not the only) mechanism which can be used to accommodate the transmission of variable-length Source Packets over the 1553 bus.

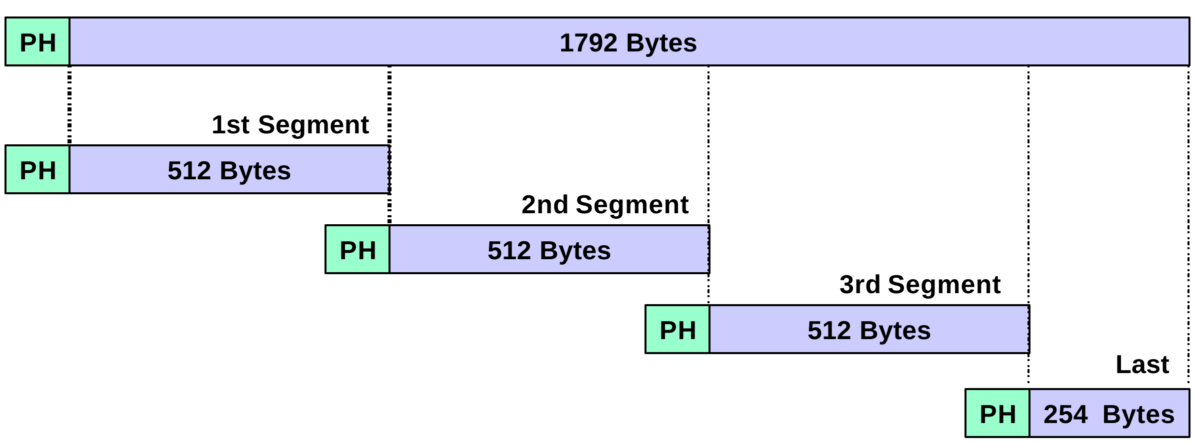

Figure 4 – Packet Segmentation Concept

In prior issues of the Packet Telemetry specification[10] an entire section was devoted to packet segmentation, and a special Packet Version number was specified for segmented packets. However in the current issue[5] the section on packet segmentation and the special version number have been withdrawn. Therefore, the terms "packet grouping" and "packet segmentation" are often used interchangeably, even though that may or may not have been intended by the CCSDS.

Transfer Layer

The transfer layer introduces a second entity called the Transfer Frame. Unlike the Source Packet, the Transfer Frame is a fixed-length entity. The length of the Transfer Frame is mission-specific, but is limited to a maximum length for any mission of 8920 bits[11]. There are two formats (versions) of the Transfer Frame: Version 1 and Version 2.

Primary Header	***Insert Zone***	***Data Zone***	***Trailer***
VCDU I.D. Sequence Ctr	*Optional*	User Data e.g. M_PDU	*Optional*
48 or 64 Bytes	***Mission-Specific/Fixed-Length***		

Figure 5 – Transfer Frame (VCDU) Format

Version 2 is introduced in the AOS specification[9] as the Virtual Channel Data Unit (VCDU). The format of the Version 2 Transfer Frame (VCDU) is shown in Figure 5. The primary role of the Transfer Frame is to facilitate the transmission of data from the spacecraft to the ground. Because the Transfer Frame is fixed-length, it provides an alternative means for the transfer of variable-length data across the 1553 bus. However, a mechanism for stuffing variable-length packets into fixed-length frames must be established. In prior versions of the Packet Telemetry specification, it appears that the use of packet segmentation was intended for this purpose. However, the special section on packet segmentation was withdrawn from the latest issue of the Packet Telemetry specification, even though the format of the Source Packet still facilitates it. An alternative mechanism for multiplexing Source Packets into VCDUs is provided for in the AOS standard, called the Multiplexing Protocol Data Unit (M_PDU). It may be that the use of the M_PDU field is now preferred over the use of packet segmentation, or it could be that the unique segmented packet version number and issues related to packet sequence numbering caused segmentation to be withdrawn, but this is not clear (more on this later). The format of the M_PDU is shown in Figure 6.

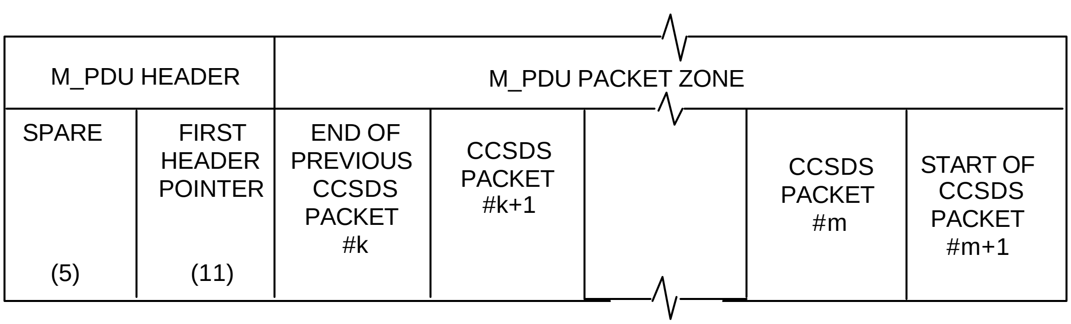

Figure 6 – Muliplexing Protocol Data Unit Format

In this brief introduction to CCSDS it is appropriate to mention one final point. CCSDS provides a number of elements, some of which have more than one version, and each of which leave a certain amount of flexibility in definition to the protocol designer. This has implication in the selection (or accommodation) of the mission ground support equipment (GSE) as not all systems which claim CCSDS compatibility support all aspects of the standard. For example, not all existing systems support both versions of the Transfer Frame/VCDU. Other systems have tighter constraints on the maximum size of a Source Packet. Specific systems may or may not support packet segmentation or use of the M_PDU. It is important for the system engineer to ascertain and consider the constraints of the particular ground systems (including ground and space-based tracking and data relay systems, control center systems, and integration and test consoles) when selecting which aspects of CCSDS to employ.

Applying CCSDS to MIL-STD-1553

The designer of a CCSDS-over-1553 protocol must consider:

- the physical (i.e. data bus) architecture of the observatory,
- the potential pitfalls of each standard,
- which aspects of CCSDS are supported by the spacecraft and instrument GSE, and
- the required level of coupling between the spacecraft and the payload.

With the exception of the last point all of these issues have been discussed. This section will present two examples of how these two standards can be combined in the implementation of on-board C&DH protocols. Each example will include a discussion of the observatory architecture and level of coupling, which aspects of CCSDS were employed, and how the protocol addresses the various pitfalls which have been presented. The two observatories chosen as examples are both medium-size explorer (MIDEX) satellites, but have different physical architectures, utilize different GSE, and require a different level of coupling between the spacecraft and the payload. Therefore, while there are similarities in the two protocols, there are also some unique aspects which serve to highlight some alternatives.

IMAGE Observatory Implementation

The Imager for Magnetopause-to-Aurora Global Exploration (IMAGE) satellite is a MIDEX observatory which uses neutral atom, ultraviolet and radio imaging techniques to study the relationship between the solar wind and the earth's magnetosphere, and to measure Coronal Mass Ejection (CME)-related neutral atom fluxes and radio emissions as forecasting tools for geomagnetic storms. The IMAGE observatory was launched on March 25, 2000, and is currently on-orbit and operating in full science mode. The IMAGE web site is at http://image.gsfc.nasa.gov.

The architecture of the IMAGE satellite resembles that pictured in Figure 1-B. The spacecraft communicates with a single payload controller over the 1553 bus, which in turn communicates with the instrument controllers over RS-422. This design results in some architectural overhead, but provides substantial advantages in the design of the 1553/CCSDS protocol and in terms of spacecraft/payload decoupling.

The IMAGE spacecraft and payload controllers are very loosely coupled; the spacecraft serves primarily as a "bent pipe" for science data produced by the payload. The only detailed communications between the two include a spacecraft time and attitude message and a payload heartbeat indicator.

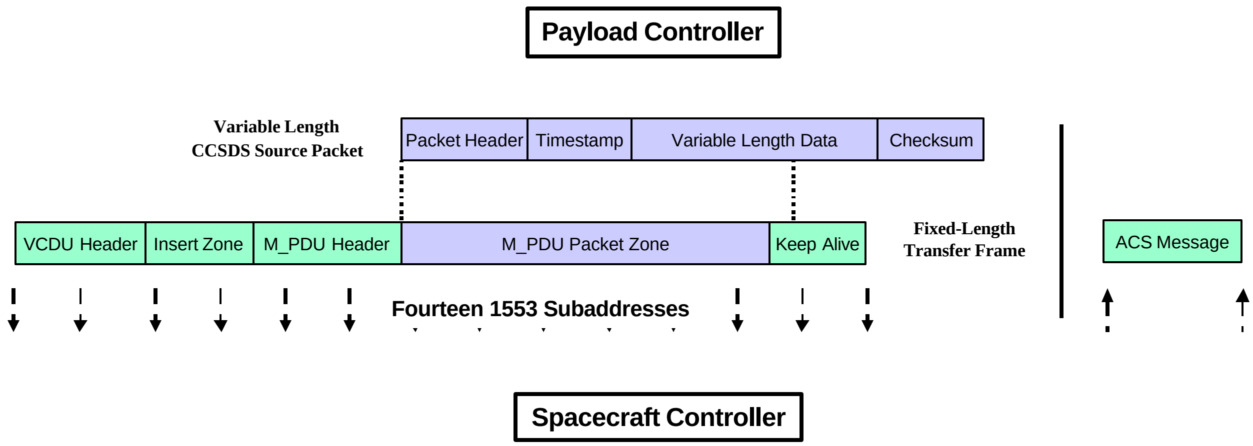

Figure 7 – IMAGE Spacecraft-to-Payload Telemetry Protocol

The observatory operates in a straightforward "single data mode". That is, all science and housekeeping data are simultaneously sent to the solid-state recorder (SSR) and transmitted over the radio frequency link. During a real-time contact, the ground systems identify and process the housekeeping packets while commanding a playback of the SSR to capture science and pre-contact housekeeping data.

One of the challenges of using 1553 for a science payload data bus is that science data is typically variable-length in nature, while the 1553 bus requires data to be transmitted in fixed-length frames. The simplicity of the IMAGE architecture provided an opportunity to take advantage of the CCSDS Transfer (framing) Layer in the implementation of the telemetry protocol between the spacecraft and the payload. The IMAGE telemetry protocol essentially requires the payload controller to multiplex the variable-length CCSDS Source Packets into 888–byte Version 2 Transfer Frames (VCDUs) using the M_PDU mechanism. The entire VCDU is then divided across fourteen sequential 1553 subaddresses to facilitate the transfer of the whole VCDU to the spacecraft as one "virtual transfer" (using fourteen 1553 commands). These fourteen subaddresses are treated as one contiguous transfer buffer. The VCDU sequence counter is exploited to provide a means by which the spacecraft can detect stale data. In operation, the spacecraft reads all fourteen subaddresses every 100 milliseconds, synchronized to a 10Hz time and attitude message sent to the payload controller by the spacecraft. The spacecraft then checks the VCDU sequence counter contained in the first subaddress. If the sequence counter has incremented, the VCDU is accepted and forwarded to the SSR and to the transmitter. If the sequence counter has not incremented, the VCDU is deemed to be "stale" and is discarded. Finally, a "payload heartbeat" counter is contained within the unused bytes at the end of the last subaddress, so that the spacecraft can ascertain that the payload is alive during extended periods of stale data. The IMAGE telemetry protocol scheme is illustrated in Figure 7.

Payload commands are implemented as 64-byte entities—the maximum length of a single 1553 transfer. The spacecraft receives payload commands from the ground as CCSDS Telecommand Packets, extracts the command from the packet data field, and forwards it to the payload controller over the 1553 subaddress indicated in the Telecommand Packet header. The payload processor determines which instrument to forward the command to, based on the 1553 subaddress the command is received on. The spacecraft time and attitude message is transmitted in a like manner over its own subaddress.

The IMAGE spacecraft, payload, and instrument controllers all contain local clocks which maintain mission elapsed time (MET). On IMAGE, time synchronization between the spacecraft and the payload is accomplished exclusively over the 1553 bus. The spacecraft transmits a message to the payload controller every 100 milliseconds which contains the current spacecraft MET. That MET is adjusted by the payload controller to account for processing and transmission latency and then latched into its local clock. These latencies were measured to be 1.38 milliseconds during integration testing and coded into the payload controller software; latency jitter was also measured and was determined to be well below the 6 millisecond time accuracy requirement. Time synchronization between the payload controller and the instrument controllers is accomplished using a time message transmitted over RS-422, coupled with a separate pulse line to provide a "time at the bell will be …" type design.

Swift Observatory Implementation

The Swift satellite is a MIDEX observatory designed to detect gamma ray bursts (GRBs) using a wide-angle x-ray telescope, and then respond by rapidly slewing in the direction of the GRB bringing it into view of two narrow-field telescopes for higher resolution, multi-wavelength observation. The Swift observatory is currently under development, and is scheduled for launch in 2003. The Swift web site is at http://swift.gsfc.nasa.gov.

The Swift observatory differs from IMAGE in both architecture and coupling. The architecture of the Swift observatory is shown in Figure 1-A. In the Swift architecture, the spacecraft controller communicates directly with each instrument controller over the 1553 bus. The selection of this architecture for Swift may be driven by its rapid response requirement. While this provides the advantages of architectural simplicity and minimal communications latencies, it carries with it the accompanying challenges in protocol definition and bus bandwidth allocation.

A second characteristic of the Swift observatory is that the instruments and the spacecraft have increased coupling between them. That is, the instruments and the spacecraft exchange various messages which serve to maintain modal synchronization among the instruments. For example, when a GRB is detected, the event sets off a series of messages such as a burst notification message, a request to slew message, and spacecraft attitude and GRB position messages. In order to facilitate rapid notification of a GRB to scientists on the ground, selected high-priority packets are sent to the ground via the Tracking and Data Relay Satellite System (TDRSS) rather than being routed to the SSR for later playback. The significance of this coupling in the context of the spacecraft-to-payload protocol is that the payload cannot reasonably multiplex Source Packets into VCDU/M_PDU frames as IMAGE did. Rather, the spacecraft must have access to the Source Packets so that it can readily distinguish, using the APID, low-priority packets that will be sent to the SSR from high priority packets that will be sent to the TDRSS.

The Swift telemetry protocol is similar to the IMAGE protocol in that it utilizes multiple subaddresses as a transfer buffer to "parallelize" the transmission of a block of data from an instrument to the spacecraft. However in the Swift protocol, one or more Source Packets are placed into the transfer buffer rather than a single VCDU. The VCDU sequence counter is not available for use in stale data detection, therefore a special counter is placed in the first two bytes of the first subaddress in the series to provide stale data detection. On IMAGE, BC/RT collisions are avoided by synchronizing telemetry transfers from the payload controller with the 10 Hz attitude message sent from the spacecraft controller. This is not possible on Swift since each instrument will be polled at a different rate. On Swift, polling rate synchronization between the spacecraft and the instruments is maintained by way of a special spacecraft message sent to each instrument at its unique polling rate, and this message doubles as a “transfer complete” indication to the instrument. The Swift protocol defines the instrument heartbeat as the transmission of new telemetry; each instrument controller is required to produce a housekeeping packet at a minimum specified rate in order to be considered alive. The Swift telemetry protocol is illustrated in Figure 8.

Swift utilizes fifteen subaddresses for the transfer buffer; this places an upper limit on the size of a Source Packet to 958 bytes (960 bytes less 2 bytes for the stale data detection counter). Larger packets can be accommodated by using packet segmentation, but the spacecraft controller ignores the grouping flags and therefore packet reassembly is relegated to the ground systems. For Swift, this is significant because the selected test and operations ground system does not handle packet reassembly. Since the control center system is only concerned with housekeeping telemetry (separate systems are responsible for constructing science data products), a requirement was levied back to the observatory to avoid segmenting housekeeping packets. Packet reassembly has inherent problems in any case because a method of dealing with packet sequence numbers has to be devised. In the previous issue of the Packet Telemetry standard[10], the sequence counter of each segment of a segmented packet was defined to contain the sequence number of the original packet. When packets were eventually reassembled, a given sequence of packets would then contain consecutive sequence counters. However, in this definition it is not possible to determine how to properly order a group of packet segments which arrive out of order. On the other hand, if each segment of a segmented packet is assigned a new sequence number, then when the packets are reassembled there will be large gaps in the sequence, and it will not be possible to determine whether all the gaps are the result of packet reassembly or packet loss. It may be that this difficulty is why the CCSDS redefined “packet segmentation” as “packet grouping” and perhaps implicitly removed the concept of packet reassembly altogether. Some protocol designers have chosen to view packet segmentation as merely a convenient protocol mechanism for the transmission of data from instruments to the spacecraft, and therefore require the spacecraft to reassemble the packets. This is a cleaner approach, since the problems with packet sequence numbering are dealt with before the packets are ever transmitted to the ground. But this approach puts additional processing burden on the spacecraft processor.

The Swift payload command protocol is very similar to IMAGE, except that all commands are transmitted to the instrument on the same 1553 subaddress, and the entire Telecommand Packet is forwarded to the instrument.

Like IMAGE, the Swift spacecraft and instrument controllers all maintain local copies of spacecraft time, and synchronization with the spacecraft clock makes use of a 1553-born message. However, Swift couples the 1553 message with a one pulse-per-second (1pps) signal to each instrument. This is of great advantage, as it eliminates the need for the instruments to factor in processing and transmission latencies. These adjustments were required on IMAGE since there was no

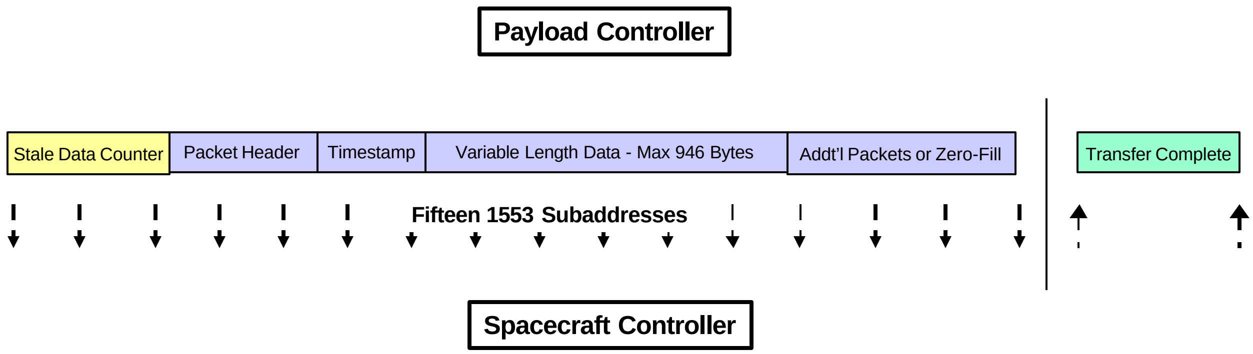

Figure 8 – Swift Spacecraft-to-Payload Telemetry Protocol

1pps signal from the spacecraft. However, the burden of these adjustments was minimal since the payload controller was the single time-cognizant remote terminal resident on the 1553 bus. Because the Swift spacecraft controller communicates directly with each instrument, synchronizing all the clocks without the 1pps would be unwieldy at best, and of questionable accuracy.

Conclusion

The MIL-STD-1553B is a stable and established bus standard with roots in military aircraft avionics. As a result of its reliability and certain electrical and mechanical advantages, 1553 has become a common component in the design of satellite C&DH systems. However, it also presents some challenges when designing telemetry data interface protocols. The system engineer should carefully consider the fundamental characteristics of 1553 before adopting it as the sole spacecraft data bus.

Many of the CCSDS recommendations are reaching maturity and have found acceptance in the space community. Consequently, 1553 and CCSDS are often paired together and used as the foundation of spacecraft-to-payload data interface protocols. However, the protocol designer must determine which aspects of CCSDS to employ in the protocol, based on the spacecraft architecture and the capabilities of available ground systems.

The 1553 and CCSDS standards are powerful tools when used in concert in the development of spacecraft data bus protocols. However, a variety of pitfalls must be avoided. System engineers exploring this open systems alternative should investigate the design successes and failures of other missions that have implemented protocols based on 1553 and CCSDS.

References

1. "Interface Standard for Digital Time Division Command/Response Multiplex Data Bus", MIL-STD-1553B, Notices 1-4, United States Department of Defense, January 15, 1996.
2. "An Introduction to CCSDS", Administrative Report A10.1-Y-3.1, Yellow Book, Consultative Committee for Space Data Systems, May 1997.
3. "MIL-STD-1553 Designer's Guide", Third Edition, DDC ILC Data Device Corporation, Bohemia, New York, 1990.
4. "Telecommand Part 3 – Data Management Service Architectural Specification", Recommendation CCSDS 203.0-B-1, Blue Book, Consultative Committee for Space Data Systems, January 1987.

5. “Packet Telemetry”, Recommendation CCSDS 102.0-B-4, Blue Book, Consultative Committee for Space Data Systems, November 1995.
6. “Time Code Formats”, Recommendation CCSDS 301.0-B-2, Blue Book, Consultative Committee for Space Data Systems, April 1990.
7. “Lossless Data Compression”, Recommendation CCSDS 121.0-B-1, Blue Book, Consultative Committee for Space Data Systems, May 1997.
8. “Report Concerning Space Data System Standards – Telemetry Summary of Concept and Rationale”, Report CCSDS 100.0-G-1, Green Book, Consultative Committee for Space Data Systems, December 1987.
9. “Advanced Orbiting Systems, Networks and Data Links: Architectural Specification”, Recommandation CCSDS 701.0-B-2, Blue Book, Consultative Committee for Space Data Systems, November 1992.
10. “Packet Telemetry”, Recommendation CCSDS 102.0-B-3, Blue Book, Consultative Committee for Space Data Systems, November 1992.
11. “Telemetry Channel Coding”, Recommandation CCSDS 101.0-B-4, Blue Book, Consultative Committee for Space Data Systems, May 1999.