TELEMETRY

The purpose of a telemetry system is to convey measurement information reliably and transparently from a remotely located data generating source to users located in space or on earth. Typically, data generators are scientific sensors, science housekeeping sensors, engineering sensors, and other subsystems on board a spacecraft.

The advent of capable microprocessor-based hardware will result in data systems with demands for greater throughput and a requirement for corresponding increases in spacecraft autonomy and mission complexity. These facts, along with the current technical and fiscal environments, create a need for greater telemetering capability and efficiency with reduced costs.

The telemetry link is the voice of the spacecraft, providing two basic types of information: performance and experimental information. By performance measurement we mean spacecraft operating conditions, consisting of temperature, pressure, voltage, current, subsystem monitoring, and so forth. The experimental measurements are related to scientific objectives of the mission, like investigations of solar plasma, magnetic fields, and micrometeorites. Due to the large number of measurements to be performed, it is necessary to time-multiplex them according with some priority.

Telemetry data can be categorized into three basic forms: *engineering parameter data, attitude,* and *payload.*

Engineering parameter data, also known as housekeeping data, keep check on the operating status and health of the spacecraft's on-board equipment. The following are a few forms and examples:

1. *Temperature.* Thermistors are usually used to convert temperature data into their voltage analog. In case of high temperatures, thermocouples are used to detect the temperature whose output does not exceed more than few milivolts. This voltage signal corresponding to temperature is dc amplified to a level more suitable for the telemetry encoder.
2. *Pressure.* Various forms of pressure transducers are used to monitor pressure in fuel tanks and plenum chambers.
3. *Operating Status.* The operating status of a particular piece of equipment is represented by a single bit that

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*.

indicates a function mode is enabled (Logic Hi) or disabled (Logic Low). For proportional status information, such as amplifier gain setting, the bits are grouped into words of appropriate length.

4. *Redundancy Status.* The redundant status indicates which redundant side of equipment is in use. This is monitored by a status bit (either Logic Hi or Logic Low).
5. *Deployment of Mechanisms.* A microswitch is used to provide a status bit of separation from the launcher.
6. *Voltages and Currents of Equipment and Power Supplies.* Usually these voltages are scaled to a common full-scale range. Current monitoring may involve a few signal processing circuits.

These are just a few examples of engineering parameter data to be monitored on a modern satellite payload. The result of every command is checked via the telemetry. Typical sampling and updating of these data are performed from every few seconds to few minutes and hence require small bandwidths. This makes the engineering parameter data bit rate a few hundred per second for sufficient transmittal of total information.

Attitude data arise from variety of sensors, such as gyroscopes, star mappers, accelerometers, and sun and earth sensors. The data can be analog, digital, or both.

During the transfer and intermediate orbit phase, the attitude and velocity will change rapidly, and frequent sampling is needed (i.e., one to four times per second). High sampling rates needed during some mission phases may lead to a bandwidth that exceeds that which can be provided conveniently by a standard PCM data system.

Payload data are variable and require individual consideration. For example, applications and scientific payloads are likely to need only few channels of data, but their rates may be high. Data compression may be required to reduce the downlink rate.

After launch the command link is required in conjunction with telemetry link to supply information needed by the various spacecraft subsystems. Many commands are necessary for the routine operation and control of spacecraft functions; some are provided for changing the mission emphasis if unusual or unexpected conditions are encountered, while others are required to correct erratic operations or to salvage the mission if a spacecraft failure occurs. Command operation varies in complexity from simple on-off operations to trajectary-correction factors (based on tracking) for maneuvers. Usually, all of the commands will fall into two classes: (1) the discrete command for switching functions, which requires only an address for identification and execution, and (2) the quantitative command, which requires a magnitude in addition to the address.

TELEMETRY DATA ENCODING

All data considered arise from three basic forms: analog, digital bilevel, and digital serial. In conditioning *analog data* the first step is scaling it to a common full-scale range; 0 V to 5 V is the usual range. The frequency components greater than half the sampling frequency are then removed by a low-pass filter to prevent aliasing errors. Filtering channels are then sampled and converted to a digital word. An overall accuracy of about 1% is sufficient, and an 8 bit A/D converter is used to achieve this.

In *digital bilevel* data the off state is represented by zero voltage and the on state by +5 V for complementary metal–oxide semiconductor (CMOS) and 2.4 V for transistor–transistor logic (TTL) logic. Individual groups are then arranged into 8 bit words and sampled by logic gates whose outputs are serialized and mixed with the main PCM data stream.

Digital data are usually acquired in serial form for reasons of simplifying the spacecraft's cable harness. Such data are initially stored as an 8 bit or 16 bit word in the shift register located in the equipment generating the data.

The usual spacecraft telemetry system is broken down into two major conceptual categories: a packet telemetry concept (1) and a telemetry channel coding concept (2).

Packet telemetry is a concept that facilitates the transmission of space-acquired data from source to user in a standardized and highly automated manner. Packet telemetry provides a mechanism for implementing common data structures and protocols that can enhance the development and operation of space mission systems. Packet telemetry addresses the following two processes:

1. The end-to-end transport of space mission data sets from source application processes located in space to distributed user application processes located in space or on earth.
2. The intermediate transfer of these data sets through space data networks; more specifically, those elements that contain spacecraft, radio links, tracking station, and mission control centers as some of their components

Packet telemetry along with telemetry channel coding services provide the user reliable and transparent delivery of telemetry information.

TELEMETRY SYSTEM CONCEPT

The system design technique known as layering was found to be a very useful tool for transforming the telemetry system concept into sets of operational and formatting procedures. The layering approach is patterned after the International Organization for Standardization's Open Systems Interconnection layered network model, which is a seven-layer architecture that groups functions logically and provides conventions for connecting functions at each layer. Layering allows a complex procedure such as the telemetering of spacecraft data to the users to be decomposed into sets of peer functions residing in common architectural strata.

Within each layer, the functions exchange data according to established standard rules, or protocols. Each layer draws on a well-defined set of services provided by the layer below and provides a similarly well-defined set of services to the layer above. As long as these service interfaces are preserved, the internal operations within a layer are unconstrained and transparent to other layers. Therefore, an entire layer within a system may be removed and replaced as dictated by user or technological requirements without destroying the integrity of the rest of the system. Furthermore, as long as the appropriate interface protocol is satisfied, users can interact with the system/service at any of the component layers. Layering is therefore a powerful tool for designing structured systems that change due to the evolution of requirements or technology.

A companion standardization technique that is conceptually simple, yet very robust, is the encapsulation of data within an envelope or header. The header contains the identifying information needed by the layer to provide its service while maintaining the integrity of the envelope contents.

Packetization Layer

Within packet telemetry, spacecraft-generated application data are formatted into end-to-end transportable data units called TM (transfer frame) source packets. These data are encapsulated within a primary header that contains identification, sequence control, and packet length information and an optional trailing error control field. A TM source packet is the basic data unit telemetered to the user by the spacecraft and generally contains a meaningful quantity of related measurements from a particular source.

Segmentation Layer

To provide assistance with data flow control, the Packet Telemetry Recommendation provides the capability to segment large packetized transportable data units into smaller communication-oriented TM source packets (Version 1 format) or TM segments (Version 2 format) for transfer through the space data channel. Consequently, the TM source packets and/or TM segments are of proper size for placement into the data field of the data unit of the next lower layer.

Transfer Frame Layer

In spacecraft communication the TM (transfer frame) is used to transport source packets and segments through the telemetry channel to the receiving telecommunications network. TM transfer frame protocols offer a range of delivery service options. An example of such a service option is the multiplexing of TM transfer frames into virtual channels (VCs).

The TM transfer frame begins with an attached frame synchronization marker and is followed by a primary header. The primary header contains frame identification, channel frame count information, and frame data field status information. The transfer frame data field may be followed by an optional trailer containing an operational control field and/or a frame error control field. The first of these fields provides a standard mechanism for incorporating a small number of real-time functions (e.g., telecommand verification or spacecraft clock calibration). The error control field provides the capability for detecting that which may have been introduced into the frame during the data handling process. The delivery of transfer frames requires the services provided by the lower layers (e.g., carrier, modulation/detection, and coding/decoding) to accomplish its role.

Channel Coding Layer

Since a basic system requirement is the error-free delivery of the transfer frames, telemetry channel coding is used to protect the transfer frames against telemetry channel noise-induced errors. Reference 2 describes the Consultative Committee for Space Data Systems (CCSDS) Recommendation for Telemetry Channel Coding, including specification of a convolutionally encoded inner channel concatenated with a Reed–Solomon block-oriented outer code (4). The basic data units of the CCSDS Telemetry Channel Coding that interface with the layer below are the channel symbols output by the convolutional encoder. These are the information bits representing one or more transfer frames as parity-protected channel symbols.

The RF channel physically modulates the channel symbols into signal patterns interpretable as bit representations. Within the error detecting and correcting capability of the channel code chosen, errors that occur as a result of the physical transmission process may be detected and corrected by the receiving entity.

Telemetry source packets may be segmented and placed into the data field of telemetry segments, which are preceded by a header. The source packets and/or the segments are placed into the data field of the transfer frame, which is preceded by a transfer frame header. If the specified Reed–Solomon code is used in the channel coding scheme, the transfer frame is placed into the Reed–Solomon data space of the Reed–Solomon codeblock, and the codeblock is preceded by an attached synchronization marker.

Relationship Between Telemetry and Telecommand Systems

A different level of understanding is revealed by considering interactions between the telemetry system and other systems in the operational environment. There is a balanced relationship between the telemetry system and the uplink telecommand system. The two systems work hand in hand to ensure the transfer of user directives from the sending end (traditionally on the ground) to the receiving end (controlled process, device, or instrument). Of course, the telemetry system does a great deal more than simply returning command receipt status information to the sender: Its usual function is to provide reliable, efficient transfer of all spacecraft data (housekeeping, sensor readings, etc.) back to users.

TELEMETRY DATA FORMATTING

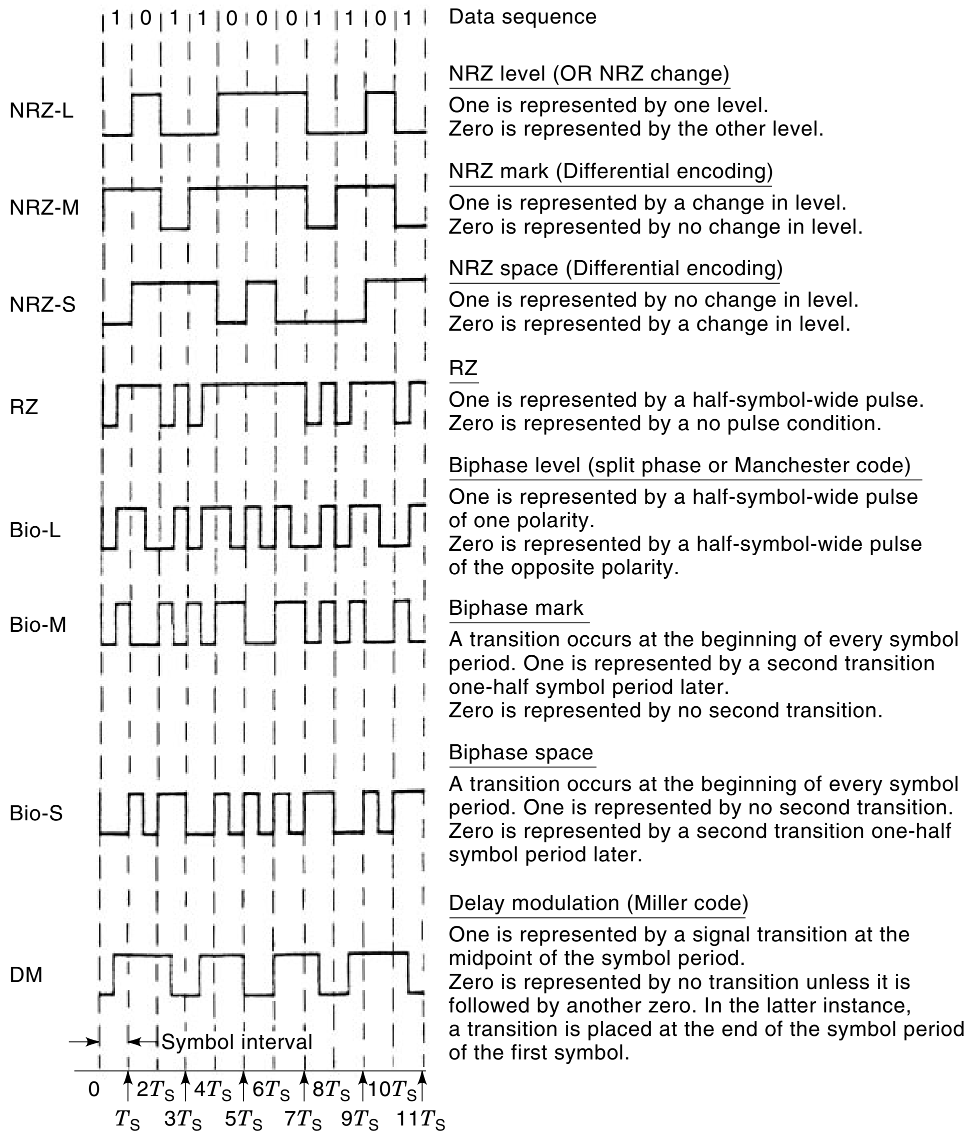
The baseband data $d_i(t)$ can have different formats, as illustrated in Fig. 1. With a nonreturn to zero-level (NRZ-L) data format, a logical one is represented by one level and a logical zero by the other. With NRZ-M (mark), a logical one is represented by a change in level and a logical zero by no change. Two other formats, biphase and Miller, are also defined in Fig. 1.

Referring to Fig. 1 and assuming that binary waveform levels are $\pm A$, the NRZ signaling format falls into the class of signals (5) whose spectrum is given by

$$S_m(f) = \frac{1}{T} p(1-p)|S_1(f) - S_2(f)|^2 + \frac{1}{T^2} \sum_{n=-\infty} \left| pS_1\left(\frac{n}{T}\right) + (1-p)S_2\left(\frac{n}{T}\right)\right|^2 \delta\left(f - \frac{n}{T}\right) \tag{1}$$

Since the elementary signal is a rectangular pulse of width T, its Fourier transform is

$$S_1(f) = -S_2(f) AT \exp(-j\pi fT) \frac{\sin(\pi fT)}{\pi fT} \tag{2}$$



NRZ level (OR NRZ change)

One is represented by one level.
Zero is represented by the other level.

NRZ mark (Differential encoding)

One is represented by a change in level.
Zero is represented by no change in level.

NRZ space (Differential encoding)

One is represented by no change in level.
Zero is represented by a change in level.

RZ

One is represented by a half-symbol-wide pulse.
Zero is represented by a no pulse condition.

Biphase level (split phase or Manchester code)

One is represented by a half-symbol-wide pulse of one polarity.
Zero is represented by a half-symbol-wide pulse of the opposite polarity.

Biphase mark

A transition occurs at the beginning of every symbol period. One is represented by a second transition one-half symbol period later.
Zero is represented by no second transition.

Biphase space

A transition occurs at the beginning of every symbol period. One is represented by no second transition.
Zero is represented by a second transition one-half symbol period later.

Delay modulation (Miller code)

One is represented by a signal transition at the midpoint of the symbol period.
Zero is represented by no transition unless it is followed by another zero. In the latter instance, a transition is placed at the end of the symbol period of the first symbol.

Figure 1. Various binary PCM waveforms.

Substituting Eq. (2) into Eq. (1) and letting $E = A^2T$, we get

$$\frac{S_m(f)}{E} = \frac{1}{T}(1-2p)^2\delta(f) + 4p(1-p)\left[\frac{\sin^2(\pi fT)}{(\pi fT)^2}\right] \tag{3}$$

When $p = 1/2$, the dc spike at the origin disappears and the NRZ signaling format falls into the noise equivalent power (NEP) class with

$$\frac{S_m(f)}{E} = \left[\frac{\sin^2(\pi fT)}{(\pi fT)^2}\right] \tag{4}$$

RZ Baseband Signaling

In the RZ case we have $S_1(f) = 0$, and $S_2(f)$ corresponds to the Fourier transform of a half-symbol-wide pulse; that is,

$$S_2(f) = \frac{AT}{2}\exp\left(\frac{-j\pi fT}{2}\right)\left[\frac{\sin\left(\frac{\pi fT}{2}\right)}{\left(\frac{\pi fT}{2}\right)}\right] \tag{5}$$

Since the source is again purely random, substituting Eq. (5) into Eq. (1) gives

$$\frac{S_m(f)}{E} = \frac{1}{4T}(1-p)^2\delta(f) + \frac{1}{4T}(1-p)^2\sum_{\substack{n=-\infty\\ n\neq 0}}^{\infty}\left(\frac{2}{n\pi}\right)^2\delta\left(f-\frac{n}{T}\right) + \frac{1}{4}p(1-p)\left[\frac{\sin^2\left(\frac{\pi fT}{2}\right)}{\left(\frac{\pi fT}{2}\right)^2}\right] \tag{6}$$

Biphase (Manchester) Baseband Signaling

Two rudimentary signals in Manchester baseband signaling are defined by

$$\begin{aligned} &s_1(t) = A; \quad 0 < t < T/2 \quad \text{and} \quad -A; \quad T/2 < t < T \\ &s_2(t) = -s_1(t) \end{aligned} \tag{7}$$

Replacing the Fourier transform of Eq. (7) into Eq. (1) will yield

$$\frac{S_m(f)}{E} = \frac{1}{T}(1-2p)^2 \sum_{\substack{n=-\infty \\ n\neq 0}}^{\infty} \left(\frac{2}{n\pi}\right)^2 \delta\left(f-\frac{n}{T}\right) + 4p(1-p)\left[\frac{\sin^4\left(\frac{\pi fT}{2}\right)}{\left(\frac{\pi fT}{2}\right)^2}\right] \tag{8}$$

for $p = 1/2$, the line spectrum disappears, and

$$\frac{S_m(f)}{E} = \left[\frac{\sin^4\left(\frac{\pi fT}{2}\right)}{\left(\frac{\pi fT}{2}\right)^2}\right] \tag{9}$$

The Miller coding scheme can be modeled as a Markov source with four states whose stationary probabilities all equal to 1/4 and whose transition matrix is given by

$$P = \begin{matrix} 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 1/2 & 1/2 \\ 1/2 & 1/2 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 \end{matrix} \tag{10}$$

Another property of the Miller code is that it satisfies the recursion relation

$$P^{4+i}\Gamma = -\frac{1}{4}P^{i}\Gamma, \qquad i \geq 0 \tag{11}$$

where Γ is the signal correlation matrix whose *ik*th is defined by

$$\gamma ik \equiv \frac{1}{\sqrt{E_i E_j}} \int_0^T s_i(t)s_k(t)d(t) \qquad i,k = 1,2,3,4 \tag{12}$$

For the Miller code, the four rudimentary signals are defined as

$$\begin{aligned} &s_1(t) = -s_4(t) = A; && 0 \leq t \leq T \\ &s_2(t) = -s_3(t) = A; && 0 \leq t \leq T/2 \\ &s_2(t) = -s_3(t) = -A; && T/2 \leq t \leq T \\ &\text{and } E_i = A^2T; && i = 1,2,3,4 \end{aligned} \tag{13}$$

Substituting Eq. (13) into Eq. (12) and putting the results in the form of a matrix,

$$\Gamma = \begin{matrix} 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{matrix} \tag{14}$$

Finally, using Eqs. (10), (11), and (14) in the general PSD result, which is

$$S_m(f) = \frac{1}{T}\sum_{i=1}^{M} p_i|s_i'(f)|^2 + \frac{1}{T^2}\sum_{n=-\infty}^{\infty}\left|\sum_{i=}^{M} p_i s_i\left(\frac{n}{T}\right)\right|^2 \delta\left(f\frac{n}{T}\right) + \frac{2}{T}\mathrm{Re}\left[\sum_{i=1}^{M}\sum_{k=1}^{M} p_i s_i^{*\prime}(f)Pik(e^{-j2\pi fT})\right] \tag{15}$$

where $S_i(f)$ is the Fourier transform of the *i*th elementary signal $s_i(t)$ and

$$p_{ik}(z) \equiv \sum_{n=1}^{\infty} p_{ik}^{(n)} z^n \tag{16}$$

$$s_i'(t) = s_i(t) - \sum_{k=1}^{N} p_k s_k(t) \tag{17}$$

yields the result of Miller code (5)

$$\frac{S_m(f)}{E} = \frac{1}{2\theta^2(17+8\cos 8\theta)}(23 - 2\cos\theta - 22\cos 2\theta - 12\cos 3\theta + 5\cos 4\theta + 12\cos 5\theta + 2\cos 6\theta - 8\cos 7\theta + 2\cos 8\theta) \tag{18}$$

where

$$\theta \equiv \pi fT \tag{19}$$

Spectral properties of the Miller code that make it valuable are as follows:

1. The majority of the signaling energy lies in frequencies less than one-half of the data rate, $R = 1/T$.
2. The spectrum is small, in the vicinity of $f = 0$. This spectral minimum facilitates carrier tracking, which can also be more efficiently achieved than Manchester coding.
3. The Miller coding is insensitive to the 180° phase ambiguity common to NRZ-L and Manchester coding.
4. Bandwidth requirements are approximately one-half those needed by Manchester coding.

When the data pulse stream experiences data asymmetry, distortion of the continuous component of the PSD as well as the presence of a line spectrum in PSD occurs. This problem obviously degrades the error probability of the receiving system. Let us look at the PSD of NRZ and Manchester streams when data assymetry is present.

NRZ Data. Let us assume that $+1$ NRZ symbols are elongated by $\Delta T/2$ (relative to their nominal value of T s) when a negative-going data transition occurs and -1 symbols are shortened by the same amount when a positive-going data transition occurs. During the absence of data transition the symbols maintain their nominal T-s value.

Using generalized M-ary source model, where $M = 4$ with

$$\begin{aligned}
s_1(t) &= A; && -T/2 \le t \le T(1+\Delta)/2 \\
s_1(t) &= 0; && \text{otherwise} \\
s_2(t) &= -A; && -T/2 \le t \le T(1-\Delta)/2 \\
s_2(t) &= 0; && \text{otherwise} \\
s_3(t) &= A; && -T/2 \le t \le T/2 \\
s_3(t) &= 0; && \text{otherwise} \\
s_4(t) &= -A; && -T/2 \le t \le T/2 \\
s_4(t) &= 0; && \text{otherwise}
\end{aligned} \tag{20}$$

The stationary probabilities associated with those four waveforms are

$$\begin{aligned}
p_1 &= pp_t; & p_2 &= (1-p)p_t; \\
p_3 &= p(1-p_t); & p_4 &= (1-p)(1-p_t)
\end{aligned} \tag{21}$$

where p refers to the transition probability, which is related to the priori probability of the $+1$ NRZ symbol, p, by

$$p_t = 2p(1-p) \tag{22}$$

Taking the Fourier transform of Eq. (20) and substituting the results in Eq. (1), with a great detail of simplification we get

$$\begin{aligned}
S_m(f) = {} & A^2T\frac{\sin^2(\pi fT)}{(\pi fT)^2}[A_1(p_t) + A_2(p, p_t, \eta)] \\
& + A^2T\frac{\sin^2(\pi fT\eta)}{(\pi fT)^2}A^3T\frac{\sin^2(\pi fT\eta)}{(\pi fT)^2}A^3(p_t, \eta) \\
& + A^2T\frac{\sin(2\pi fT)}{(\pi fT)^2}[A_4(p, p_t, \eta) - A_5(p, p_t)] \\
& + A^2[2p - (1-\eta p_t)]^2\delta(f) \\
& + \frac{2A^2}{\pi^2}p_t^2\sum_{n=1}^{\infty}\frac{1}{n^2}C(n, p, \eta)\delta\left(f\frac{n}{T}\right)
\end{aligned} \tag{23}$$

where

$$\begin{aligned}
A_1(p_t) &= p_t(1-p_t)[1+2(1-p_t)] - p_t^3 \\
A_2(p, p_t, \eta) &= (3p_t^3 + p_t(1-p_t)[1+2(1-2p)])\cos^2(\pi fT\eta) \\
A_3(p_t, \eta) &= p^t(1+p_t^2-p_t))\cos^2(\pi ft) + p_t^3\cos(2\pi fT\eta) \\
A_4(p, p_t, \eta) &= \Bigg(p_t(1-p_t)(1-2p)\Bigg[\frac{1}{2}\cos(2\pi fT\eta) \\
&\qquad - p\sin(2\pi fT\eta)\Bigg]\Bigg) \\
A_5(p, p_t) &= \frac{1}{2}p_t(1-p_t)(1-2p) \\
C(n, p, \eta) &= \sin^2(n\pi\eta)[\cos^2(n\pi\eta) + (1-2p)^2\sin^2(n\pi\eta)] \\
\eta &= \frac{\Delta}{2}
\end{aligned} \tag{24}$$

Manchester Data. Let us assume that for $+1$ data bit the first half of the Manchester symbol is elongated by $\Delta T/4$ (relative to its nominal value of $T/2$). The same will extend to the -1 symbol—the first half of the Manchester symbol shortened by the same amount. When no data transition occurs, the second half of the Manchester symbol retains its T-s value. In view of the preceding asymmetry model, we can use the generalized M-ary source model, where $M = 4$, with

$$\begin{aligned}
s_1(t) &= A; && -T/2 \le t \le \Delta T/4 \\
s_1(t) &= -A; && \Delta T/4 \le t \le (T/2)(1+\Delta/2) \\
s_1(t) &= 0; && \text{otherwise} \\
s_2(t) &= -A; && -T/2 \le t \le -\Delta T/4 \\
s_1(t) &= A; && \Delta T/4 \le t \le (T/2)(1-\Delta/2) \\
s_2(t) &= 0; && \text{otherwise} \\
s_3(t) &= A; && -T/2 \le t \le \Delta T/4 \\
s_3(t) &= -A; && \Delta T/4 \le t \le (T/2) \\
s_3(t) &= 0; && \text{otherwise} \\
s_4(t) &= -A; && -T/2 \le t \le -\Delta T/4 \\
s_4(t) &= A; && \Delta T/4 \le t \le (T/2) \\
s_4(t) &= 0; && \text{otherwise}
\end{aligned} \tag{25}$$

As before, the stationary probabilities are associated with Eq. (21). Taking Fourier transforms of Eq. (25) and substituting the results in Eq. (1), we get

$$S_m(f) = (S_m(f))_c + (s_m(f))_d \tag{26}$$

where for $p = p_t = 1/2$ the discrete component $(S_m(f))_\mathrm{d}$ is given by

$$\begin{aligned}
(s_m(f))_d = {} & \frac{9}{4}A^2\eta^2\delta(f) + \frac{2A^2}{\pi^2}\sum_{m=1}^{\infty}\frac{1}{m^2}[H_1(m, \eta) + H_2(m, \eta) \\
& + H_3(m, \eta)]\delta\left(f - \frac{m}{T}\right)
\end{aligned} \tag{27}$$

with

$$\begin{aligned}
\eta &= \frac{\Delta}{4} \\
H_1(m, \eta) &= \frac{\sin^2(m\pi\eta)[1+2h_1(m, \eta)]^2}{4} \\
H_2(m, \eta) &= \sin^2(2m\pi\eta) \\
H_3(m, \eta) &= 2\sin^2(m\pi\eta)(\cos m\pi\eta)[1+2h_1(m, \eta)]
\end{aligned} \tag{28}$$

where

$$h_1(m, \eta) = \cos^2\left(\frac{m\pi\eta}{2}\right); \quad \text{for } m \text{ odd} \tag{29}$$

and

$$h_1(m, \eta) = \sin^2\left(\frac{m\pi\eta}{2}\right); \quad \text{for } m \text{ even}$$

Likewise, for $p = p_t = 1/2$, the continuous component of Eq. (26) is given by

$$
\begin{aligned}
(S_m(f))_c = {} & \frac{A^2T^4}{4}\,\frac{\sin^4\left(\frac{\pi fT}{4}\right)}{\left(\frac{\pi fT}{4}\right)^2} \\
& - A^2T[c_1(\eta) + c_2(\eta) + c_3]\frac{\sin^2(\pi fT\eta)}{\left(\frac{\pi fT}{2}\right)^2} \\
& - A^2TC_4(\eta)\frac{\sin^2\left(\frac{\pi fT\eta}{2}\right)}{\left(\frac{\pi fT}{2}\right)^2} \\
& + A^2TC_5(\eta)\frac{\sin^2\left(\frac{\pi fT(1+\eta)}{2}\right)}{\left(\frac{\pi fT}{2}\right)^2} \\
& + A^2TC_6(\eta)\frac{\sin^2\left(\frac{\pi fT(1-\eta)}{2}\right)}{\left(\frac{\pi fT}{2}\right)^2}
\end{aligned}
\tag{30}
$$

where

$$
\begin{aligned}
C_1(\eta) &= \frac{1}{4}\sin^2\left[\frac{\pi fT(1+\eta)}{2}\right]\left\{\sin^2\left[\frac{\pi fT(1-\eta)}{2}\right] + \cos\pi fT\eta\right\} \\
C_2(\eta) &= \frac{1}{8}\cos\pi fT\eta\left\{2\sin^2\left[\frac{\pi fT(1-\eta)}{2}\right] - \sin^2\left[\frac{\pi fT\eta}{2}\right]\right\} \\
C_3(\eta) &= \frac{1}{8}\left[1 - 4\cos\left(\frac{\pi fT}{4}\right)\right] \\
C_4(\eta) &= \frac{\sin\pi fT\eta}{8}\left\{\sin\left(\frac{\pi fT\eta}{2}\right)\right. \\
&\quad \left. + \sin\left(\frac{5\pi fT\eta}{2}\right)[1 - \cos\pi fT\cos\pi fT\eta] - \right\} \\
&\quad - \frac{3}{8}\sin\left(\frac{\pi fT}{4}\right)\sin\left(\frac{\pi fT\eta}{4}\right) \\
&\quad + \frac{1}{8}[2\cos 3\pi fT\eta + \cos 2\pi fT\eta] \\
C_5(\eta) &= \frac{1}{8}\left\{\sin^2\left(\frac{\pi fT(1-\eta)}{2}\right) - \frac{3}{2}\sin^2\left[\frac{\pi fT(1+\eta)}{2}\right]\right\} \\
C_6(\eta) &= \frac{3}{16}\sin^2\left(\frac{\pi fT(1-\eta)}{2}\right) + \frac{1}{4}\sin^2\left(\frac{\pi fT}{2}\right)\cos\pi fT\eta
\end{aligned}
\tag{31}
$$

TELEMETRY SYSTEM DESCRIPTION AND RATIONALE

Packet Telemetry

The traditional way of transmitting scientific applications and engineering data has been the time division multiplexing (TDM) method. Packet telemetry represents an evolutionary step from the traditional TDM method. The packet telemetry process conceptually involves:

1. Encapsulating, at the source, observational data (to which may be added ancillary data to interpret subsequently the observational data), thus forming an autonomous packet of information in real time on the spacecraft
2. Providing a standardized mechanism whereby autonomous packets from multiple data sources on the spacecraft can be inserted into a common frame structure for transfer to another space vehicle or to earth through noisy data channels and delivered to facilities where the packet may be extracted for delivery to the user

The packet telemetry process has the following conceptual attributes:

1. Facilitating the acquisition and transmission of instrument data at a rate appropriate for the phenomenon being observed
2. Defining a logical interface and protocol between an instrument and its associated ground support equipment that remains constant throughout the life cycle of the instrument (bench test, integration, flight, and possible reuse)
3. Simplifying overall system design by allowing microprocessor-based symmetric design of the instrument control and data paths (telecommand packets in, telemetry packets out) compatible with commercially available components and interconnection protocol standards
4. Eliminating the need for mission-dependent hardware and/or software at intermediate points within the distribution networks through which space data flow; in particular, enabling the multimission components of these networks to be designed and operated in highly automated fashion, with consequent cost and performance advantages
5. Facilitating interoperability of spacecraft whose telemetry interfaces conform to CCSDS guidelines (i.e., allowing very simple cross strapping of spacecraft and network capabilities between space agencies)
6. Enabling the delivery of high-quality data products to the user community in a mode that is faster and less expensive than would be possible with conventional telemetry

Figure 2 shows a functional diagram of the telemetry data flow from the creation of a data set by an application process operating within a spacecraft "source" (instrument or subsystem), through to the delivery of the same data to a user "sink" (application process) on the ground. Since many of the elements of this flow are currently mission unique, a primary objective of packet telemetry is to define stable, mission-independent interface standards for the communications path within the flow.

Telemetry Source Packet

A telemetry source packet is a data unit that encapsulates a block of observational data that may include ancillary data and that may be directly interpreted by the receiving end application process. Detailed discussion of the format specification for the telemetry source packet is specified in Ref. 1. The source packet format (Version 1), with the addition of a secondary header and packet error control field, is reproduced in Fig. 3 for the convenience of the reader.

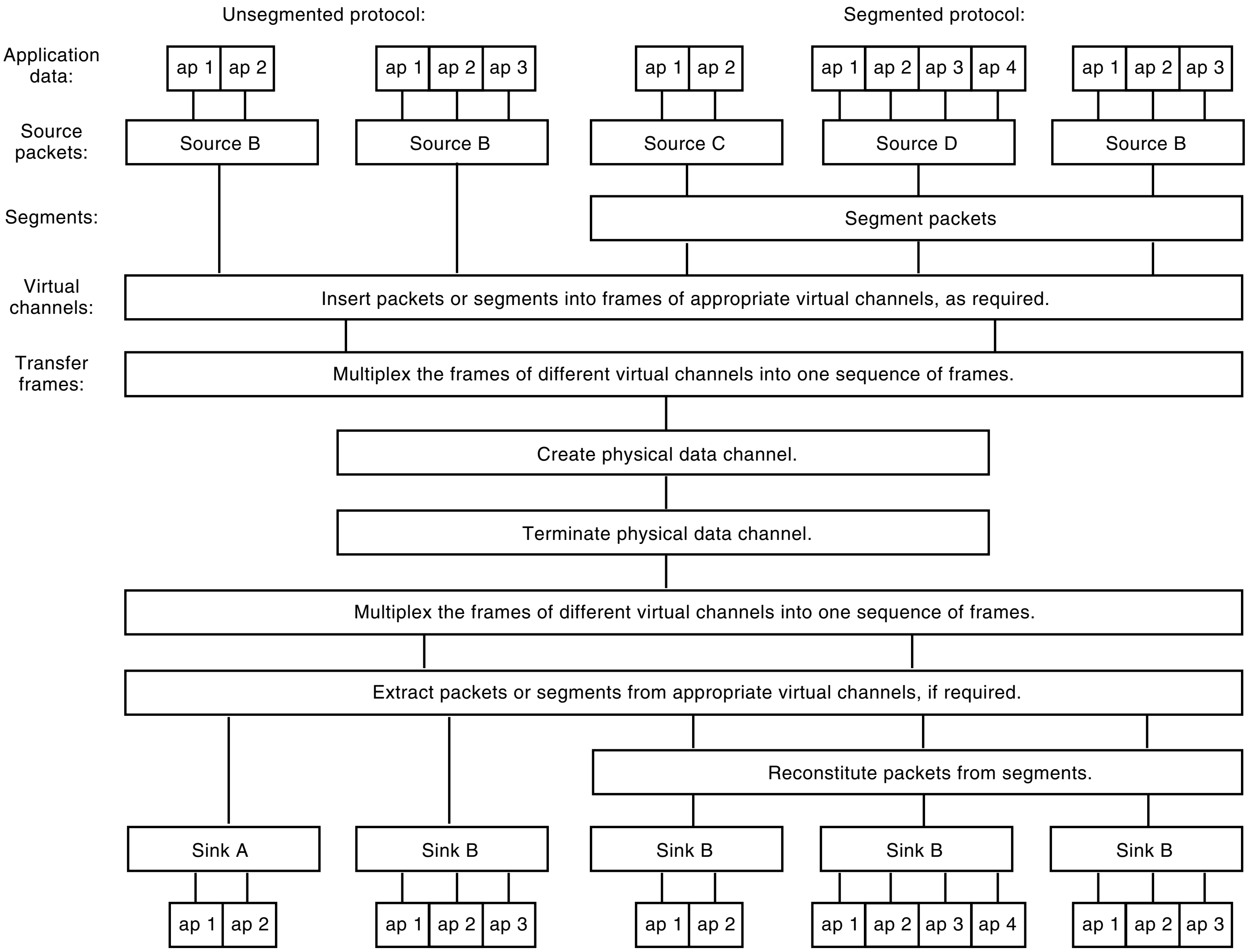


Figure 2. Telemetry data flow.

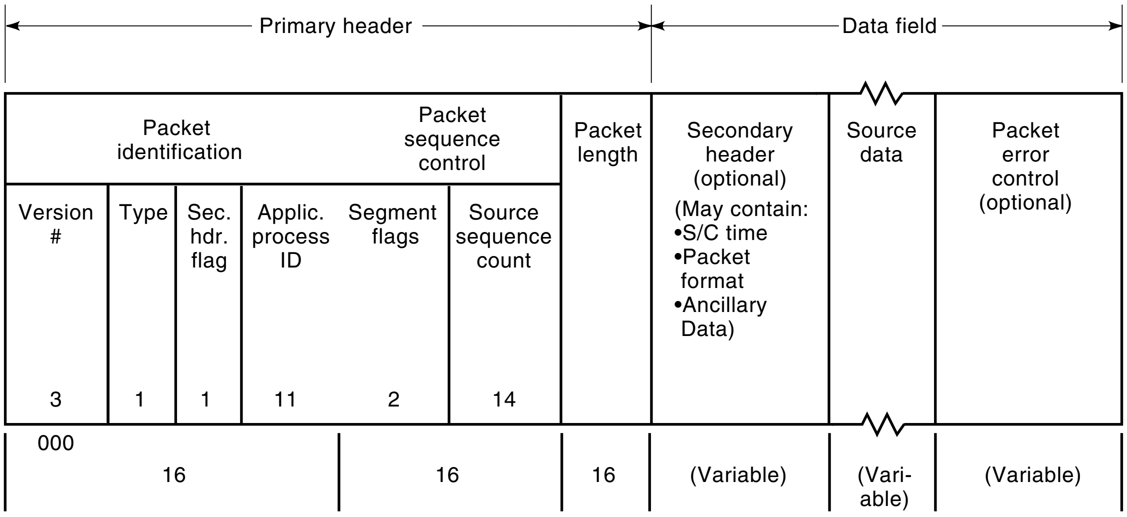


Figure 3. Source packet.

From the viewpoint of data processing efficiency, the CCSDS strongly recommends that all major fields of all telemetry formats should be an even number of octets. This facilitates efficient internal processing within 16 bit or 32 bit computers, which are anticipated to be widely used in application processes.

User application data are encapsulated within a packet by prefacing them with a standard label or primary header, which is used by the data transport system to route the data through the system and to allow the user to reconstruct the original data set. The primary header consists of three main fields: packet identification, packet sequence control, and packet length.

Packet Identification

Version Number. The version number is the first of four subfields of packet identification. This subfield explicitly indicates the version of the formatted packet, and its length of 3 bits allows eight different versions to be identified. While only two versions are currently defined, this arrangement allows a reasonable growth capability to support future needs. However, in the interest of constraining the proliferation of standards, additional versions will be discouraged unless it can be demonstrated that the current versions are truly inadequate.

Type. The second subfield is a 1 bit identifier to signal that this packet is a telemetry packet and not a telecommand packet. It is always set to zero for telemetry packets. (In the first issue of Ref. 1 [May 1984] this field was described as a "reserved spare" and was, by convention, set to zero for telemetry. In Issue 2 [January 1987], the value of the field had not changed, but its function had been established.)

Secondary Header Flag. The third subfield is a 1 bit secondary header flag. The CCSDS recognizes that users may need a means of encapsulating ancillary data (such as time, internal data field format, spacecraft position/attitude), which may be necessary for the interpretation of the information contained within the packet. Therefore, this flag, when set to one, indicates that a secondary header follows the primary header.

Application Process ID. The last subfield in the packet identification field is used to identify the originating source packet application process. In conventional free flyer spacecraft, source data (packets) are traditionally routed to the corresponding user application process on earth; this field could then also be used as a destination ID. (As such, the need for separate destination ID does not seem apparent. However, if users require one or more different destination IDs, these could be placed in the secondary header.) Eleven bits are allocated to the application process ID, permitting identification of up to 2048 separate application processes per spacecraft, sufficient for any envisioned free flyer spacecraft. For positive identification, one can consider this subfield an extension of the spacecraft ID, which is in the transfer frame primary header (see Fig. 5).

Packet Sequence Control

Segmentation Flags. The first subfield of the packet sequence control field is called segmentation flags and provides for a logical representation of four types of segmentation status. These flags identify whether the source data field contains the first, continuing, or last segment of a source packet, or if it contains no segment (meaning it contains a complete set of source application data).

Source Sequence Count. This second subfield provides for each packet to be numbered in a sequential manner, thus providing a method of checking the order of source application data at the receiving end of the system. It is normally used for ground accounting purposes to measure the quantity, continuity, and completeness of the data received from the source. The field provides a straight sequential count to modulo 16,384. Longer-term unambiguous ordering (beyond 16,384 packets) may be accomplished by associating the measurement time code contained within the packet with the source sequence count.

Packet Length. The last major field of the primary header delimits the boundaries of the packet. It is a count of the number of octets in the packet, beginning with the first octet after the 48 bit primary header and ending with the last octet of the packet. The 16 bit field allows packet lengths up to 65,536 octets (not counting the 48 bit primary header). This packet limit was a compromise between the majority of users (who produce medium-size packets) and the few users who may produce exceptionally long packets. Placing a reasonable limit on packet size helps avoid the flow control problems associated with very long packets and eliminates the overhead penalty of a larger-length field for the great majority of packet producers.

Data Field. The remainder of the packet may consist of any data desired, although some suggestions are provided by the Recommendation. The total length of all subsequent data should be an even number of octets (a multiple of 16 bits) for efficiency in computer processing. In addition, Fig. 3 indicates three possible subfields: secondary header, source application data, and a packet error control field.

Secondary Header. A secondary header may be desirable for providing any ancillary data generated by another application process (time, spacecraft position/attitude) or for providing an internal data field format. The CCSDS has not developed a recommendation for the format, but in order to allow for the future standardization of the secondary header, the most significant bit (bit 0) of the first octet of each secondary header shall be set to 0 to signify a non-CCSDS-defined secondary header.

Source Data. Following the secondary header, the source data subfield contains source application data generated by the application process identified in the primary header. For efficiency in computer processing, this subfield should be a multiple of 16 bits.

Packet Error Control. At the discretion of the user, an optional error detection code may be included at the end of the packet to verify that the overall integrity of the message has been preserved during the transport process. The particular implementation of such an error detection code, including the selection of the encoding polynomial and the length of the field, is left to the user or to the local agency.

Flow Control Mechanisms

Space telecommunications systems are usually constrained by the capacity or the bandwidth of the telecommunications channel that connects the spacecraft to the data capture element located in space or on earth. Flow control becomes crucial when multiple users must share the same telecommunications channel. The telemetry system must ensure that all sources have proper access to this common resource frequently enough to ensure timely delivery as well as to control the need to buffer data while other sources are being serviced. Long source packets may present flow control problems if they monopolize the data channel for unacceptable periods of time while forcing other sources to implement unreasonably large local buffering of their data. Several alternative solutions to the problem of flow control are presented in the Recommendation.

Virtual Channelization. One solution to the flow control problem is to assign each source (which generates long packets) its own virtual channel. This is accomplished by inserting these packets into specially identified transfer frames. These dedicated frames form a virtual channel and may be interleaved with other frames containing data from other users.

Source-Internal Segmentation: Source Packet (Version 1). Another solution to the flow control problem is accomplished entirely within the source, whereby it manipulates its own 'segmentation flags when producing packets. That is, if the source is producing a very long message or data unit, it breaks the unit into segments that can fit into working-size Version 1 packets. This way, the spacecraft data system and ground see and handle normal packets whose data fields actually contain segments of a long message whose reassembly by the application can be assured by use of the packet sequence control.

Packet Identification. Except for the secondary header flag, the packet identification fields of each of the source packets created from the original very long message are identical. For example, the secondary header flag may indicate a secondary header present in the first packet of the sequence and not in subsequent packets of the sequence.

Packet Sequence Control. The packet containing the first segment of the original very long message is identified by setting the segmentation flags in the primary header to 0,1. The source sequence count value is incremented by one for each packet of the sequence. The actual value for the first segment depends on the running count at the time the first segment is to appear. Packets containing continuation segments are identified by setting the segmentation flags to 0,0. The sequence of packets is identified by incrementing the source sequence count for each packet. The packet containing the last segment of the original very long message is identified by setting its segmentation flags to 1,0.

Packet Length. Since the packet length field is used to point to the beginning of the next packet for purposes of extraction from the transfer frame, the packet length must always refer to the length of the source packet being handled. The total length of the original very long message can be provided by the user through private, internal message labeling.

Spacecraft Segmentation: Source Packet (Version 1). Instead of source-internal segmentation, another alternative is a more centralized approach to data flow control wherein the spacecraft data system performs the segmentation. Spacecraft segmentation is accomplished by breaking up a completely formed original long source packet and inserting the pieces into newly generated, shorter Version 1 source packets; but in this case the shorter source packets are created by the spacecraft data system instead of the source itself and carry the "S/C data system" application process ID.

Packet Identification. The application process ID in the packet identification field indicates that the spacecraft data system is generating the source packets containing the segments.

Packet Sequence Control. The segmentation flags are set as described in the previous section. The source sequence count subfield contains the count value generated by the spacecraft data system and is incremented for each segment produced. (The original long packet sequence count value remains hidden in the data field of the first packet generated by the spacecraft data system.)

Packet Length. As in the previous section, the packet length field indicates the length of the newly generated packet.

Spacecraft Segmentation: Telemetry Segment (Version 2). The segmentation options discussed previously utilize the source packet (Version 1) format, in which the length is always based on the length, in octets, of the data field (packet or segment) that is transmitted, and the sequence count increments once per packet generated by a given application process. When a long packet (Version 1) requires segmentation, the monotonically increasing nature of the source sequence count, during the source packet generation process, may be disrupted.

For those missions that require the source sequence count for a given application process to increase without any gaps in the sequence, another formatting option exists. Version 2 of the packet format, called a telemetry segment, is a format within which the length field in the data unit defines the length of an original packet that remains to be transmitted, and the sequence count field remains static because it refers to the numbering of the original source packet generated by its application process. The length and sequence count of the data unit being transmitted are, therefore, semantically different between the two versions.

It is assumed that telemetry segments (Version 2) are always generated by an application process other than the original application process. In most cases, such telemetry segments will be generated by the spacecraft data system.

The telemetry segment (Version 2) structure is shown in Fig. 4.

Segment Identification. When a long source packet (Version 1) is segmented using the telemetry segment protocol, the packet ID field is modified only by changing the version number subfield to indicate Version 2. This implies that a separate application process is doing the segmentation and, therefore, the application process ID subfield contains the value of the original application process.

Segment Sequence Control. The protocol for the segmentation flags subfield is the same as for the Version 1 format except that the sequence count subfield indicates the count of the original long packet being segmented and is not incremented for each segment generated. As such, it would seem as though each segment cannot be uniquely identified, but in fact the following fields do provide a mechanism for assigning a serial number to each segment. The serial number may then be used to recombine segments should their natural order be disturbed during transmission or the data handling process.

Segment Length. Instead of indicating the length of the segment, the Version 2 format segment length field is based on the length of data (in octets) from the original long packet (including that contained within the segment) that remains to be transmitted. The length of the segment is a fixed value

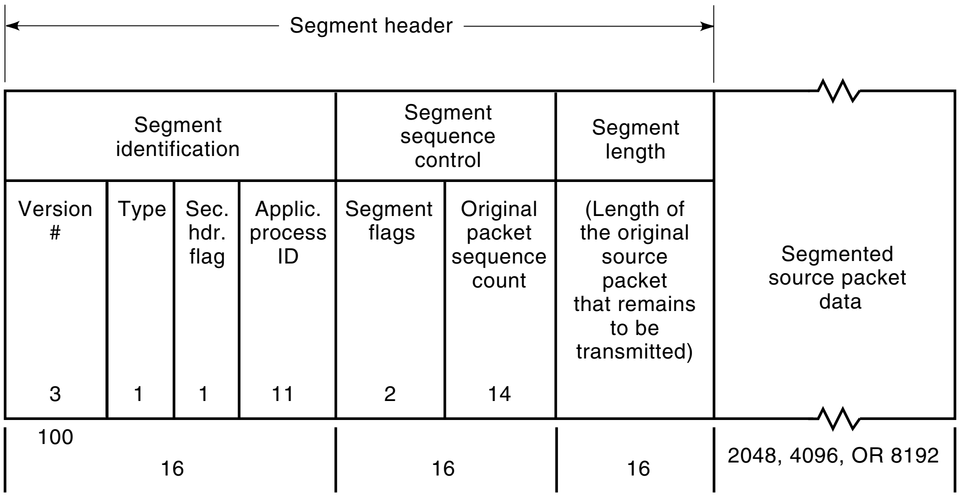


Figure 4. Telemetry segment.

(256, 512, or 1024 octets) for each virtual channel and is specified in the transfer frame header.

Since the fixed segment lengths are defined to be binary values of octets, by utilizing the decrementing length approach, the value of the segment length field will decrease in binary countdown fashion as successive segments are transmitted. This information provides a "serial number" for the segment that may be used to recombine segments should their natural order be disturbed during transmission.

Telemetry Transfer Frame

The source packet data structures described in the previous sections are unsuitable for transmission directly through the communication links that interconnect the spacecraft and data capture element in space or on earth. They must be embedded within a data transfer structure that provides reliable, error-controlled transfer through the media. The CCSDS has developed such a data structure, the telemetry transfer frame, which has a fixed length for a given mission or spacecraft. The attributes of the transfer frame and its supporting rationale will follow during the discussion of the transfer frame format. Figure 5 illustrates the telemetry transfer frame format.

Synchronization Marker. Attached to the beginning of the transfer frame primary header is a 32 bit frame synchronization marker that is used by the receiving network to acquire synchronization with the frame boundaries after transmission through the data channel. A 32 bit synchronization pattern is selected because it provides very good synchronization qualities in a noisy channel environment. The 32 bit pattern is also double-octet compatible with 32 bit computers. The particular bit pattern and its performance characteristics are found in Refs. 1 and 6.

In conjunction with the selection of the 32 bit marker, the recommendations currently require that all transfer frames in a single physical data channel in a given mission be of constant length. When the frame is of fixed length, conventional "flywheeling" techniques may be used to maintain frame synchronization in a noisy environment.

The maximum distance from one attached sync marker to the next when using the maximum-length transfer frame (8920 bits), Reed–Solomon check symbols (1280 bits), and sync marker (32 bits) is 10,232 bits.

Frame Identification. The first major field of the transfer frame primary header is the frame identification field.

Version Number. Only one version of the transfer frame has been defined by the CCSDS, although this 2 bit field allows growth to four. The version refers to the frame structuring principles, which are described in this section. Given the small number of tracking networks, as opposed to the number of end users (packet creators), and the flexibility built into this version to meet future needs, the size of the field is considered adequate.

Spacecraft ID. The spacecraft identification field provides for positive identification of the spacecraft that generated the transfer frame. The 10 bits assigned to spacecraft identification allow up to 1024 separate positive IDs. Spacecraft IDs are assigned per the procedures in Ref. 7 by the CCSDS, and analysis (8) has shown that under those procedures 1024 is an adequate number for future needs.

Virtual Channel ID. This 3 bit subfield allows up to eight virtual channels to be run concurrently on a particular physical data channel. Frames from different virtual channels are multiplexed together on the telecommunications channel, and, with this identifier in each frame, can be easily split apart after receipt at the ground. Virtual channels can be used for a variety of purposes, such as flow control to prevent long packets from "hogging" the channel; and selecting different types of data for stream splitting at the ground. Eight virtual channels are considered sufficient to provide adequate flexibility for envisioned future free flyer spacecraft.

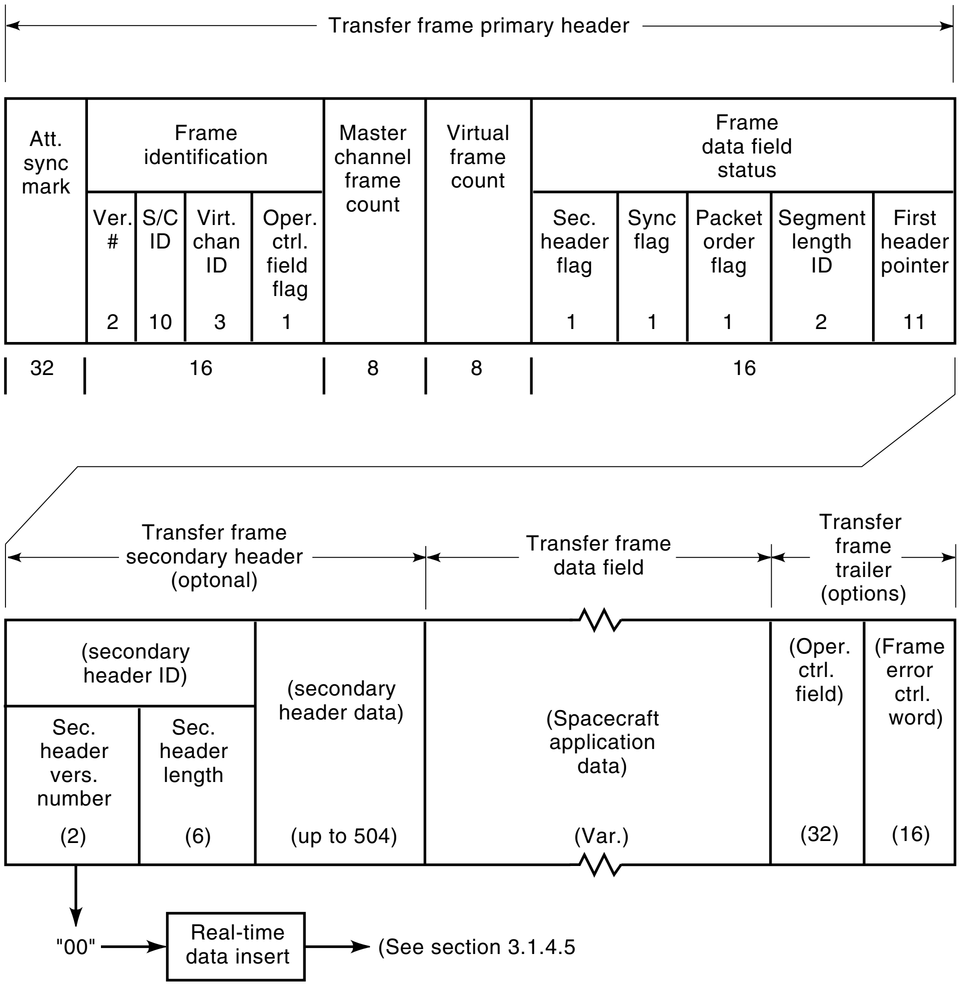


Figure 5. Telemetry transfer frame format.

Operational Control Field Flag. The last bit of the frame identification field, when set to one, signals the presence of the 32 bit operational control field, which is contained within the frame trailer. The information in this field is defined to provide a standardized spacecraft reporting mechanism for spacecraft telecommanding.

Master Channel and Virtual Channel Frame Count. The next two fields provide a running count of the number of frames transmitted. These counters provide a degree of data accountability (for short-duration data outages); the ambiguity level is defined by the field lengths.

Master Channel Frame Count. This 8 bit field provides a sequential count (modulo 256) of the number of frames transmitted by a single physical spacecraft data channel. The counter is long enough to provide a reasonable probability of detecting a discontinuity, in a sequence of frames, when the physical channel is briefly interrupted. If such a discontinuity does occur, the virtual channel accounting process can provide a greater probability of detecting the number of missing frames.

Virtual Channel Frame Count. This 8 bit field provides accountability for each of the eight independent virtual channels. This field is used with the virtual channel ID subfield to provide accountability via a sequential count (modulo 256). The rationale for the counter ambiguity level is the same as for the master channel frame counter. If only one virtual channel is incorporated for a given mission, both the virtual channel frame counter and the master channel frame counter must increment once per generated transfer frame (i.e., the two fields should not be concatenated into a master frame counter). This is because the ground facilities would normally be designed to handle the general case of spacecraft with multiple virtual channels.

Frame Data Field Status. The frame data field status field provides control information that allows the receiving end to extract and reconstitute packets and/or segments.

Secondary Header Flag. The first subfield indicates the presence or absence of the optional secondary header. If its presence is so indicated, the secondary header must appear in every frame transmitted through a physical data channel, and its length must also be fixed. Rationale for this requirement is provided later in the discussion about the secondary header.

Synchronization Flag. This flag indicates whether the packet or segment data units are inserted into the transfer frame data field on octet boundaries. If they are, then they are said to be synchronously inserted (packet octet boundaries align with frame octet boundaries) and the extraction technique (pointing to specific octet) is valid. If the flag indicates asynchronous data insertion (i.e., unstructured [nonpacketized] data contents or packets are inserted without regard to octet boundaries), then the transfer frame layer at the receiving end will not be able to reconstitute the original data sets without additional knowledge.

Packet Order Flag. This flag indicates whether the sequence count order of the contained packet or segment is increasing (forward) or decreasing (reverse). This has important implications when tape recorded data are played back opposite to their recorded direction. When this is the case, the spacecraft electronics rejustify the bit direction of each packet/segment so each packet or segment individually flows in the forward direction and its header can be read to allow proper packet extraction from the transfer frame. Even though the playback packets appear individually to flow the same as the rest of the data, the sequence of packets will be running backward in time, as indicated by the decreasing sequence counter.

Segment Length ID. The segment length identifier subfield identifies which of three fixed segment lengths are contained within the data field of the standard Version 2 telemetry segment. The lengths are fixed in order to provide a method of serializing each telemetry segment, as explained in Section 4.3.1 in Ref. 1. The 2 bit flag allows for indication of three different lengths (2048, 4096, or 8192 bits) or an indication that the Version 2 telemetry segment is not being used on this virtual channel. Three lengths provide efficient flow control for the types of data and missions envisioned. Shorter lengths are not considered because the overhead becomes unacceptably large, while higher values are not considered because virtual channelization becomes a more effective flow control method.

First Header Pointer. The first header pointer subfield points directly to the location of the starting octet of the first packet or segment header structure within the frame data field. It counts from the end of the primary header (secondary header if present) and effectively delimits the beginning of the first packet/segment. The packet/segment length field, in turn, delimits the beginning of the next packet/segment, and so on.

Since the pointer counts octets, this feature works only when the headers are aligned with octet boundaries (i.e., when the packet/segment data are synchronously inserted [data field synchronization flag set to zero]). The 11 bits allocated to the pointer allow for a count to 2048 octets, which exceeds the count required to point to an octet at the end of the data field. Special pointer values are used to denote the following:

1. No packet/segment header is contained in this frame, but there is valid data.
2. No valid data are contained in this frame ("idle channel").

Frame Secondary Header (Optional). An optional secondary header is provided for users who desire a means for deterministically inserting real-time data (e.g., time division multiplexed data) that may be required for spacecraft monitoring and control applications.

When the secondary header presence is indicated by the secondary header flag, its length must be of a fixed value and must appear in every frame transmitted through a physical channel. Given the requirement for fixed transfer frame length, a fixed secondary header length simplifies data processing and packet extraction at the receiving end.

Secondary Header ID. The first part of the secondary header has two subfields. The first is the secondary header version number, a 2 bit field allowing four versions (or structuring rules). Only one version is currently defined by the CCSDS. This provides for a reasonable future growth capability.

The second subfield, secondary header length, indicates what length has been selected for the secondary header. This 6 bit subfield provides a binary count of the total number of octets contained within the entire transfer frame secondary header (including the ID field itself, which is one octet in length). This limits the total secondary header length to 64 octets (512 bits), which is considered adequate for currently understood applications.

Secondary Header Data. This subfield contains up to 504 bits of user-specified data.

Transfer Frame Data Field. The transfer frame data field contains an integral number of octets of data (e.g., source packets and/or telemetry segments) to be transmitted from the spacecraft to the receiving element. The maximum length of this field depends on which optional fields are implemented. As discussed in Ref. 2, if frame lengths shorter than the 8920 bit maximum are implemented and the frame is encoded using the recommended Reed–Solomon algorithm, then the length of the frame data field must be selected, bearing in mind the constraint that virtual fill (see the glossary at the end of this article) must occur in fixed increments. This is necessary to simplify data processing at the receiving end. This field may also accommodate an unstructured bit stream (not necessarily packetized) as its data contents. In such a case, standard data extraction services would not be provided.

Transfer Frame Trailer (Optional). An optional transfer frame trailer is provided and is divided into two main fields, each of which is optional.

Operational Control Field. The presence or absence of the operational control field is indicated by a flag located in the frame identification field of the primary header. When present, this field facilitates closed-loop reporting of standardized real-time functions. The first bit (bit 0) of this field indicates the type of report and is currently set to zero. This signifies that this field contains a command link control word, which is used for acceptance reporting of spacecraft command activity and certain other front-end telecommunication status. This reporting mechanism is fundamental to the automated telecommand system, which is summarized in Ref. 9. The standardized internal format of the command link control word is fully defined in Ref. 10.

Frame Error Control Word. When present, this field occupies the two trailing octets of the transfer frame. Its presence or absence is implicitly defined from the spacecraft identifier and thus must or must not appear in all frames of a given spacecraft ID. It provides the capability for detecting errors that may have been introduced into the frame during the data handling processes. Its presence is mandatory if the transfer frame is *not* Reed–Solomon encoded but is optional if the frame is synchronously contained within the data space of a Reed–Solomon codeblock.

A cyclic redundancy code (CRC) has been selected for this purpose because of its effectiveness and simplicity and is defined and specified in Ref. 1, Section 5.5.2. Parity is generated over the entire transfer frame (less the final 16 bits), and the 16 bits of parity checks are then appended to complete the frame. To maintain compatibility with already built systems, it was necessary to allow for two options over which the CRC is applied: that is, it may include the sync marker or it may exclude it. Since the marker pattern is always known, the preferred choice is to omit the marker when encoding. This is explained in Ref. 1, Section 5.5.2.

TELEMETRY CHANNEL CODING

Channel coding is a method by which data can be sent from a source to a destination by processing data so that distinct messages are easily distinguishable from one another. This allows reconstruction of the data with low error probability. In spacecraft, the data source is usually digital, with the data represented as a string of zeroes and ones. A channel encoder is a device that takes this string of binary data and produces a modulating waveform as output. If the channel code is chosen correctly for the particular channel in question, then a properly designed decoder will be able to reconstruct the original binary data even if the waveforms have been corrupted by channel noise. If the characteristics of the channel are well understood and an appropriate coding scheme is chosen, then channel coding provides higher overall data throughput at the same overall quality (bit error rate) as uncoded transmission—but with less energy expended per information bit. Equivalently, channel coding allows a lower overall bit error rate than the uncoded system using the same energy per information bit.

There are other benefits that may be expected from coding. First, the resulting "clean" channel can benefit the transmission of compressed data. The purpose of data compression schemes is to map a large amount of data into a smaller number of bits. Adaptive compressors will continually send information to direct a ground decompressor in how to treat the data that follow. An error in these bits could result in improper handling of subsequent data. Consequently, compressed data are generally far more sensitive to communication errors than uncompressed data. The combination of efficient low error rate channel coding and sophisticated adaptive data compression can result in significant improvement in overall performance (11–13,15).

Second, a low bit error rate is also required when adaptive telemetry is used. Adaptive telemetry is much like adaptive data compression in that information on how various ground processors should treat the transmitted data is included as part of the data. An error in these instructions could cause improper handling of subsequent data and the possible loss of much information.

Third, low error probability telemetry may allow a certain amount of unattended mission operations. This is principally because the operations systems will know that any anomalies detected in the downlink data are extremely likely to be real and not caused by channel errors. Thus, operators may not be required to try to distinguish erroneous data from genuine spacecraft anomalies.

In a typical space channel, the principal signal degradations are due to the loss of signal energy with distance and to the thermal noise in the receiving system. The codes described in Ref. 2 can usually provide good communication over this channel. An additional degradation, caused by interference from earth-based pulse radars, may occur for users of the Tracking and Data Relay Satellite System (TDRSS). Such users may consider adding periodic convolutional interleaving (PCI) to their coding system; in this case, they should carefully analyze the effects of the PCI on their systems.

If interagency cross support requires one agency to decode the telemetry of another, then the codes recommended in Ref. 2 should be used. A block diagram of the recommended coding system appears in Fig. 6.

The relative performance of the various codes in a Gaussian channel is shown in Fig. 7. Here, the input is constrained to be chosen from between two levels, because biphase modulation is assumed throughout this Recommendation. These performance data were obtained by software simulation and assume that there are no synchronization losses. The channel symbol errors were assumed to be independent. This is a good assumption for the deep space channel. Also, infinite interleaving was assumed in the Reed–Solomon code. The use of the outer Reed–Solomon code results in an additonal 2.0 dB of coding gain. Note that Fig. 7 does not necessarily represent the performance of the TDRSS channel.

These codes are included in the CCSDS Recommendation because they represent state-of-the-art coding technology and provide substantial coding gain over an uncoded system. They have already been incorporated, or are planned to be incorporated, into missions of member agencies of the CCSDS.

The next three sections explain the choice of the codes and the parameters of each code in more detail.

Convolutional Code

A rate 1/2, constraint length 7 convolutional code with Viterbi (maximum likelihood) decoding is already a standard for both NASA and the European Space Administration (ESA). It has been used in several missions and has demonstrated the expected coding gain.

The encoder for this code is extremely simple. It consists of a shift register of length six and some exclusive OR gates that implement the two parity checks. The two checks are then multiplexed into one line. This means that the encoder can be made small and that it dissipates very little power. These are good attributes for spacecraft hardware. It has been customary to invert one or the other parity check in the encoder. This is to ensure that there are sufficient transitions in the channel stream for the symbol synchronizer to work in the case of a steady-state (all zeroes or all ones) input to the encoder.

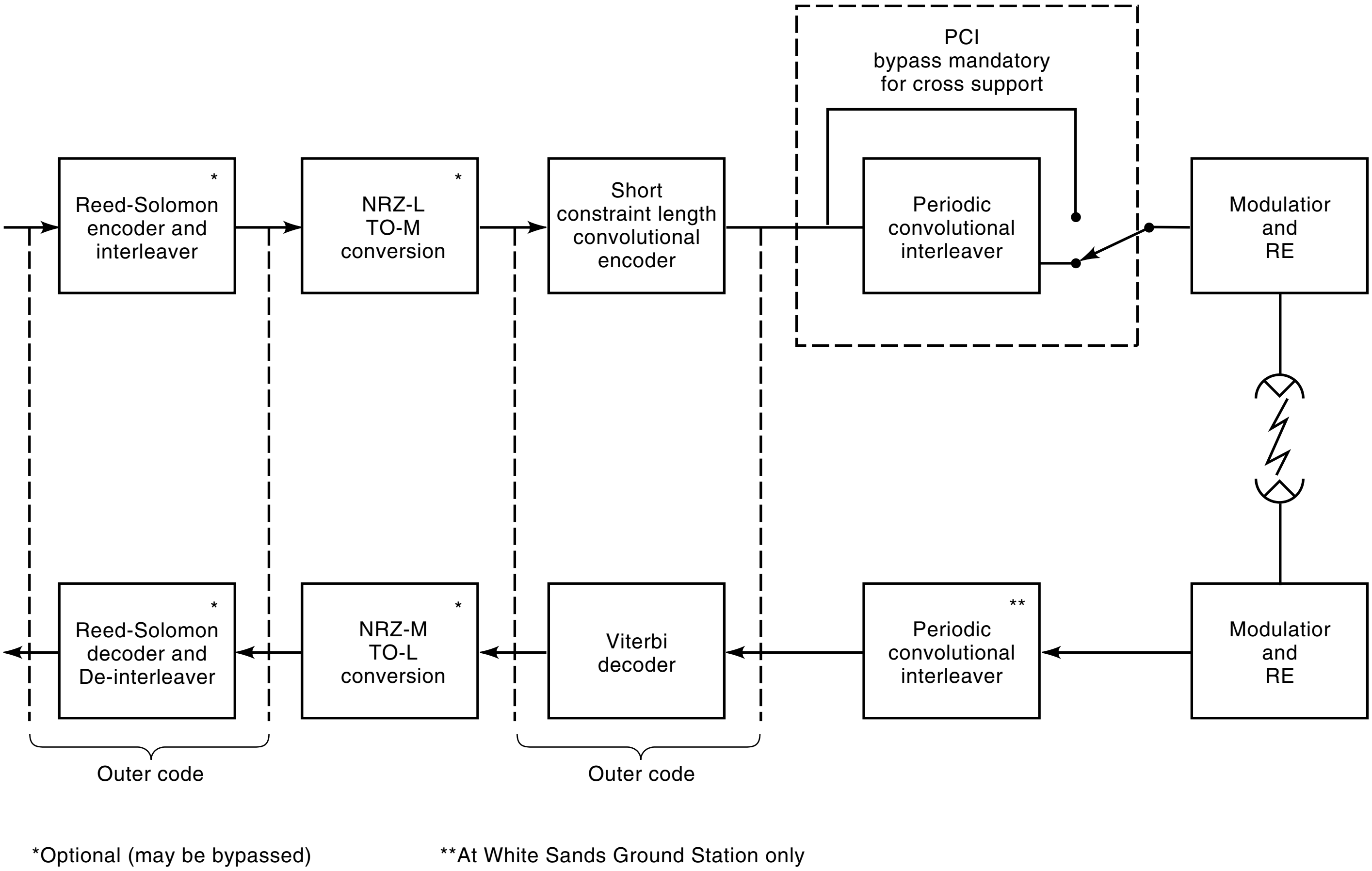


Figure 6. Block diagram of recommended coding system.

Historically, ESA, NASA-Goddard Space Flight Center (GSFC), and NASA-Jet Propulsion Laboratory (JPL) have each used a different ordering of the two parity checks or has inverted a different parity check. Performance is not affected by these minor differences. While interim cross support of these different conventions may require minor differences in ground station equipment, all agencies are encouraged to adopt for all facilities the single convention described in Ref. 2, which is the NASA-GSFC convention.

Periodic Convolutional Interleaving

Low earth-orbiting spacecraft sending telemetry to the ground using the services of the TDRSS S-band single-access

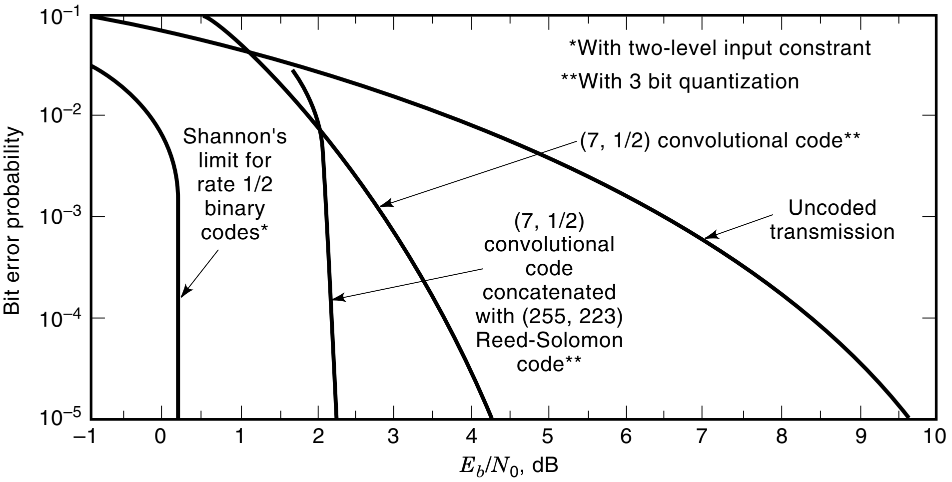


Figure 7. Performance of various codes in a Gaussian channel.

(SSA) channel when symbol rates exceed 300 kilosymbols/second (ks/s) may experience pulsed radio interference, which is expected to degrade the link performance severely during certain portions of the user orbit. To be able to maintain specified performance on this link at all times, the user satellite must employ an interleaving technique in conjunction with the convolutional coding and must increase the effective isotropic radiated power (EIRP). These techniques will ensure that no more than one of the dependent symbol errors due to a single radio frequency interference (RFI) pulse is within the path memory length of the decoder at any given time, and that the signal energy has been increased sufficiently to offset the increased symbol error probability (14). The interleaving parameters have been selected to achieve this goal for a particular worst-case pulse interference signature and the maximum symbol rate (6 Ms/s) of the SSA channel. Deinterleaving must take place before convolutional decoding and therefore is accomplished at the White Sands Ground Terminal.

Reed–Solomon Code

Due to the nature of Viterbi decoding, the decoded bit errors of the (7, 1/2) convolutional code tend to clump together in bursts. For this reason, in a concatenated coding system that uses a convolutional inner code, the outer code should be tailored to a burst error environment. The code that is recommended as the outer code is a (255, 223) Reed–Solomon code. This is a nonbinary code. Each member of its coding alphabet is one of 256 elements of a finite field rather than zero or one. A string of 8 bits is used to represent elements in the field so that the output of the encoder still looks like binary data.

Reed–Solomon codes are block codes. This means that a fixed block of input data is processed into a fixed block of output data. In the case of the (255, 223) code, 223 Reed–Solomon input symbols (each 8 bits long) are encoded into 255 output symbols. The Reed–Solomon code in the Recommendation is systematic. This means that some portion of the code word contains the input data in unalterable form. In this case, the first 223 symbols are the input data. The Reed–Solomon decoder almost always knows when there are too many errors to correct a word. In the event that this happens, the decoder can inform the user of this fact.

A Reed–Solomon symbol size of 8 bits was chosen because the decoders for larger symbol sizes would be difficult to implement with current technology. This choice forces the longest code word length to be 255 symbols. A 16 Reed–Solomon symbol error correction capability was chosen as this was shown to have the best performance when concatenated with the (7, 1/2) convolutional inner code (11,15,16). Since two check symbols are required for each error to be corrected, this results in a total of 32 check symbols and 223 information symbols per code word.

The (255, 223) Reed–Solomon code is capable of correcting up to 16 Reed–Solomon symbol errors in each code word. Since each symbol is actually 8 bits, this means that the code can correct up to 16 short bursts of error due to the inner convolutional decoder.

In addition, the Reed–Solomon code words can be interleaved on a symbol basis before being convolutionally encoded. Since this separates the symbols in a code word, it becomes less likely that a burst from the Viterbi decoder disturbs more than one Reed–Solomon symbol in any one code word. This improves the performance of the Reed–Solomon code. An interleaving depth of five was chosen for two reasons (15): A depth of five results in performance that is virtually indistinguishable from a depth of infinity. Also, a depth of five results in a frame length (a set of five code words that, together with the check symbol field, constitutes a codeblock) that is a good compromise considering ease of handling, data outages (quality, quantity, and continuity), and frame synchronization rate.

The same encoding and decoding hardware can implement a shortened $(n, n - 32)$ Reed–Solomon code, where $n = 33, 34, \ldots, 254$. This is accomplished by assuming that the remaining symbols are fixed: In the case of the Recommendation, they are assumed to be all zero. This virtual zero fill allows the frame length to be tailored, if necessary, to suit a particular mission or situation.

The method currently recommended for synchronizing the codeblock is by synchronization of the transfer frame, which contains a frame synchronization marker of 32 bits. However, advanced approaches being studied (e.g., self-synchronizing Reed–Solomon codes) may enable these two functions to be separately synchronized in the future.

The Reed–Solomon code, like the convolutional code, is a transparent code. This means that if the channel symbols have been inverted somewhere along the line, the decoders will still operate. The result will be the complement of the original data. However, the Reed–Solomon code loses its transparency if virtual zero fill is used. For this reason it is mandatory that the sense of the data (i.e., true or complemented) be resolved before Reed–Solomon decoding.

The two polynomials that define the Reed–Solomon code [Sections 4.2(4) and 4.2(5) in Ref. 2, and Ref. 17) were chosen to minimize the encoder hardware. The code generator polynomial is a palindrome (self-reciprocal polynomial) so that only half as many multipliers are required in the encoder circuit. The particular primitive element "a" (and hence the field generator polynomial) was chosen to make these multipliers as simple as possible. An encoder using the "dual basis" representation requires for implementation only a small number of integrated circuits or a single VLSI chip.

GLOSSARY

Block encoding. A one-to-one transformation of sequences of length k of elements of a source alphabet to sequences of length n of elements of a code alphabet $n\>k$.

Channel symbol. The unit of output of the innermost encoder that is a serial representation of bits, or binary digits, that have been encoded to protect against transmission-induced errors.

Clean data (bits). Data (bits) that are error free within the error detection and optional error correction capabilities of the TM system.

Codeblock. A codeblock of an (n, k) block code is a sequence of n channel symbols that were produced as a unit by encoding a sequence of k information symbols and will be decoded as a unit.

Code rate. The average ratio of the number of binary digits at the input of an encoder to the number binary digits at its output.

Code word. In a block code, one of the sequences in the range of the one-to-one transformation (see *block encoding*).

Command link control word. The telecommand system transfer layer protocol data unit for telecommand reporting via the TM transfer frame operational control field.

Concatenation. The use of two or more codes to process data sequentially with the output of one encoder used as the input of the next.

Constraint length. In convolutional coding, the number of consecutive input bits that are needed to determine the value of the output symbols at any time.

Convolutional code. As used in this document, a code in which a number of output symbols are produced for each input information bit. Each output symbol is a linear combination of the current input bit as well as some or all of the previous $k - 1$ bits, where k is the constraint length of the code.

Fill bit(s). Additional bit(s) appended to enable a data entity to fit exactly an integer number of octets or symbols.

Inner code. In a concatenated coding system, the last encoding algorithm that is applied to the data stream. The data stream here consists of the code words generated by the outer decoder.

Modulating waveform. A way of representing data bits (1 and 0) by a particular waveform.

NRZ-L. A modulating waveform in which a data one is represented by one of two levels, and a data zero is represented by the other level.

NRZ-M. A modulating waveform in which a data one is represented by a change in level and a data zero is represented by no change in level.

Octet. An 8 bit word consisting of eight contiguous bits.

Outer code. In a concatenated coding system, the first encoding algorithm that is applied to the data stream.

Packet. An efficient application-oriented protocol data unit that facilitates the transfer of source data to users located in space or on earth.

Protocol. A set of procedures and their enabling format conventions that define the orderly exchange of information between entities within a given layer of the TM system.

Reed-Solomon (R-S) symbol. A set of J bits that represents an element in the Galois field $GF(**2J)$, the code alphabet of a J-bit Reed–Solomon code.

Reliable. Meets the quality, quantity, continuity, and completeness criteria that are specified by the TM system.

Segment. A protocol data unit that facilitates telemetry flow control through the breaking of long source packets into communication-oriented data structures.

Systematic code. A code in which the input information sequence appears in unaltered form as part of the output codeword.

Telemetry system. The end-to-end system of layered data handling services that exist to enable a spacecraft to send measurement information, in an error-controlled environment, to receiving elements (application processes) in space or on earth.

Transfer frame. A communication-oriented protocol data unit that facilitates the transfer of application-oriented protocol data units through the space-to-ground link.

Transparent. The invisible and seemingly direct (virtual) transfer of measurement information from the spacecraft source application process to the user (receiving application process).

Transparent code. A code that has the property that complementing the input of the encoder or decoder results in complementing the output.

User. A human or machine-intelligent process that directs and analyzes the progress of a space mission.

Virtual channel. A given sequence of transfer frames, that are assigned a common identification code (in the transfer frame header), enabling all transfer frames who are members of that sequence to be uniquely identified. This allows a technique for multiple source application processes to share the finite capacity of the physical link (i.e., through multiplexing).

Virtual fill. In a systematic block code, a code word can be divided into an information part and a parity (check) part. Suppose that the information part is N symbols long (*symbol* is defined here to be an element of the code's alphabet) and that the parity part is M symbols long. A "shortened" code is created by taking only S ($S < N$) information symbols as input, appending a fixed string of length $N - S$ and then encoding in the normal way. This fixed string is called fill. Since the fill is a predetermined sequence of symbols, it need not be transmitted over the channel. Instead, the decoder appends the same fill sequence before decoding. In this case, the fill is called virtual fill.

BIBLIOGRAPHY

1. Packet Telemetry, Recommendation CCSDS 102.0-B-2, Issue 2, Blue Book, Consultative Committee for Space Data Systems, January 1987 or later issue.
2. Telemetry Channel Coding, Recommendations CCSDS 101.0-B-2, Issue 2, Blue Book, Consultative Committee for Space Data Systems, January 1987 or later issue.
3. Reference Model of Open Systems Interconnection, International Organization for Standardization, Draft International Standard DIS-7498, February 1982 or later issue.
4. R. F. Rice and E. Hilbert, U.S. Patent No. 3,988,677, October 26, 1976.
5. M. K. Simon, S. M. Hinedi, and W. C. Lindsey, *Digital Communication Techniques: Signal Design and Detection,* Upper Saddle River, NJ: Prentice-Hall, 1995.
6. J. C. Morakis, Discussion of synchronization words, NASA Tech. Memo. 86222, NASA-Goddard Space Flight Center, Greenbelt, Maryland, May 15, 1985.
7. Procedures Manual for the Consultative Committee for Space Data Systems, Issue 1, Consultative Committee for Space Data Systems, August 1985 or later issue.
8. R. Cager, Spacecraft Identification Requirements Analysis, CCSDS Panel 1-C Telecommand Action Item 6.26, June 3–7, 1985.
9. Telecommand: Summary of Concept and Service, Report CCSDS 200.0-G-6, Issue 6, Green Book, Consultative Committee for Space Data Systems, January 1987 or later issue.
10. Telecommand, Part 2: Data Routing Service, Architectural Specification, Recommendation CCSDS 202.0-B-1, Issue 1, Blue Book, Consultative Committee for Space Data Systems, January 1987 or later issue.

11. R. F. Rice, Channel coding and data compression system considerations for efficient communication of planetary imaging data, Tech. Memo. 33-695, NASA-Jet Propulsion Laboratory, Pasadena, CA, June 15, 1974.
12. R. F. Rice, End-to-end imaging rate advantages of various alternative communication systems, JPL Publication 82-61, NASA-Jet Propulsion Laboratory, Pasadena, CA, September 1, 1982.
13. R. F. Rice, Mission science value/cost savings from the advanced imaging communications system, JPL Publication 84-33, NASA-Jet Propulsion Laboratory, CA, July 15, 1984.
14. Tracking and Data Relay Satellite System (TDRSS) Users' Guide, STDN 101.2, Rev. 5, NASA-Goddard Space Flight Center, Greenbelt, MD, September 1984.
15. J. P. Odenwalder, Concatenated Reed-Solomon/Viterbi channel coding for advanced planetary missions, Final Rep., Contract 953866, December 1, 1974.
16. J. P. Odenwalder et al., Hybrid coding systems study, Final Rep., NASA-Ames Research Center Contract NAS2-6722, Linkabit Corporation, San Diego, CA, September 1972.
17. M. Perlman and J. J. Lee, Reed-Solomon encoders—Conventional vs Berlekamp's architecture, JPL Publication 82-71, NASA-Jet Propulsion Laboratory, Pasadena, CA, December 1, 1982.

VICTOR BOYADZHYAN
Jet Propulsion Laboratory

TELEMETRY IN MEDICINE. See BIOMEDICAL TELEMETRY (BIOTELEMETRY).

TELEMETRY, RADIO. See RADIOTELEMETRY.

TELEPHONE EXCHANGES. See TELECOMMUNICATION EXCHANGES.