POLITECNICO DI TORINO

Facoltà di Ingegneria
Corso di Laurea in Ingegneria delle telecomunicazioni

Tesi di Laurea

Sistema di telecomando e telemetria per aerei stratosferici senza pilota: definizione delle specifiche e validazione tramite simulazione

Relatori:
Prof. Mario PENT
Prof. Marina MONDIN
Ing. Fabio DOVIS

Candidati:
Francesco BAIETTO
Maurizio FODRINI
Maurizio SISTO

Dicembre 2000

Contributi dei candidati

Questa tesi è stata realizzata mediante una stretta collaborazione ed una costante cooperazione tra i candidati. Tutte le problematiche affrontate sono state esaminate singolarmente, suddividendole opportunamente per consentire un'analisi parallela dei relativi aspetti da studiare. I risultati così ottenuti sono poi stati analizzati e confrontati per decidere la strategia più adatta a risolvere tali problematiche, in modo da disporre di una base di partenza con cui impostare il problema successivo. In particolare, il contributo fornito da ciascun tesista per lo studio dei vari argomenti è stato il seguente:

- *Baietto*: analisi del modello di telemetria proposto dal CCSDS e del flusso dati in down-link, realizzazione del modello di canale, teoria della modulazione OFDM, realizzazione e simulazione dello schema DOFDM, teoria e simulazione dei codici a blocco, valutazione delle prestazioni del codice di Reed-Solomon, scansione elettronica delle schiere, antenne ad elica, valutazione del margine disponibile.

- *Fodrini*: analisi dei servizi di telemetria proposti dal CCSDS, trattazione teorica del modello del canale, teoria e simulazione della modulazione GMSK, realizzazione dello schema DOFDM, teoria e simulazione dei codici convoluzionali, valutazione dei ritardi complessivi del sistema, schiera ottenuta mediante sintesi FTM e antenna a dipoli incrociati, valutazione del link-budget per le diverse situazioni di volo.

- *Sisto*: analisi del modello di telecomando proposto dal CCSDS, analisi del flusso dati in up-link, teoria e simulazione delle modulazioni DQPSK e $\pi/4$ DQPSK, teoria e valutazione delle prestazioni fornite dal protocollo ARQ, analisi della correlazione tra i bit, schiere di tipo lineare con distribuzione uniforme, a coseno e a coseno quadrato, studio dei parametri riguardanti il calcolo del link-budget.

Ringraziamenti

Esprimiamo i nostri ringraziamenti al prof. Mario PENT e alla prof.ssa Marina MONDIN per il supporto fornitoci durante lo svolgimento di questa tesi e, in modo particolare, all'ing. Fabio DOVIS per la sua assistenza e costante disponibilità presentata nei nostri confronti.

Siamo grati, inoltre, all'ing Gabriella POVERO per la sua consulenza in merito al simulatore utilizzato, nonché ai docenti dei Dipartimenti di ingegneria Aeronautica e Spaziale, di Energetica e di Elettrica Industriale per le informazioni che gentilmente ci hanno fornito, relativamente al flusso dati.

Desideriamo anche ringraziare il tesista Roberto FANTINI per il suo contributo apportato nello studio del canale di propagazione.

Porgiamo, poi, la più sentita riconoscenza in modo particolare ai nostri rispettivi genitori, Lorenzo e Cesira BAIETTO, Italo e Maria Rosa FODRINI, Ugo e Michela SISTO, per la loro costante presenza ed il loro continuo supporto manifestatoci durante tutti questi anni.

Esprimiamo infine i nostri ringraziamenti a tutti gli amici e a quelle persone che hanno creduto nelle nostre capacità.

Indice

Capitolo 2

Schema proposto dal CCSDS per il telecomando e la telemetria

2.1 Introduzione

In questo capitolo verrà presentata la struttura del telecomando e della telemetria secondo i documenti approvati dal CCSDS (*Consultative Committee for Space Data Systems*). Questo ente è costituito da un'organizzazione internazionale di agenzie spaziali che cooperano per lo sviluppo di uno standard necessario a promuovere uno scambio di informazioni; in modo particolare, però, il CCSDS si occupa di missioni di tipo satellitare. Le specifiche fornite, quindi, dovranno essere considerate soltanto come uno schema da seguire a grandi linee; probabilmente risulterà necessario apportare alcune modifiche per adattarlo al caso della piattaforma stratosferica Heliplat. Quest'ultima, infatti, rispetto ai sistemi satellitari presenta alcune sostanziali differenze, quali, ad esempio, il problema del decollo e dell'atterraggio, la distanza da terra notevolmente inferiore, la possibilità che il velivolo sia soggetto a forti raffiche di vento, ecc., ma un'importante analogia, cioè la presenza di un sistema di telecomando e telemetria dei dati per una piattaforma spaziale senza pilota.

Il sistema di telecomando si occupa del trasporto affidabile e trasparente di informazioni di controllo da una sorgente (utente umano) ad un dispositivo remoto (strumentazione di bordo), mentre la telemetria considera il flusso di dati, in down-link, degli strumenti di bordo.

Il CCSDS ha definito un'architettura stratificata (modello di riferimento ISO-OSI, *International Organization for Standardization-Open System Interconnection*) applicabile ad un'ampia gamma di missioni; l'utilizzo della stratificazione permette alle complesse procedure di comando di essere suddivise in funzioni più semplici. La relazione tra il sistema di telemetria e il telecomando è rappresentata nella figura 2.1. Si può notare come i due sistemi cooperino per assicurare il trasferimento delle direttive d'utente dal trasmettitore al ricevitore.

2.2 Telecomando

L'architettura del Telecomando, si veda [1], prevede una suddivisione in sette strati che possono essere raggruppati nei seguenti tre tipi di servizi:

- Data Management Service
  - Application Process Layer
  - System Management Layer
  - Packetization Layer
- Data Routing Service
  - Segmentation Layer
  - Transfer Layer
- Channel Service
  - Coding Layer
  - Physical Layer

Questa struttura è sintetizzata nella figura 2.2. In essa sono anche indicati i tipi di pacchetti che vengono scambiati tra i vari strati la cui struttura è rappresentata in figura 2.3. La prima struttura è il *TC Packet*, costituito da

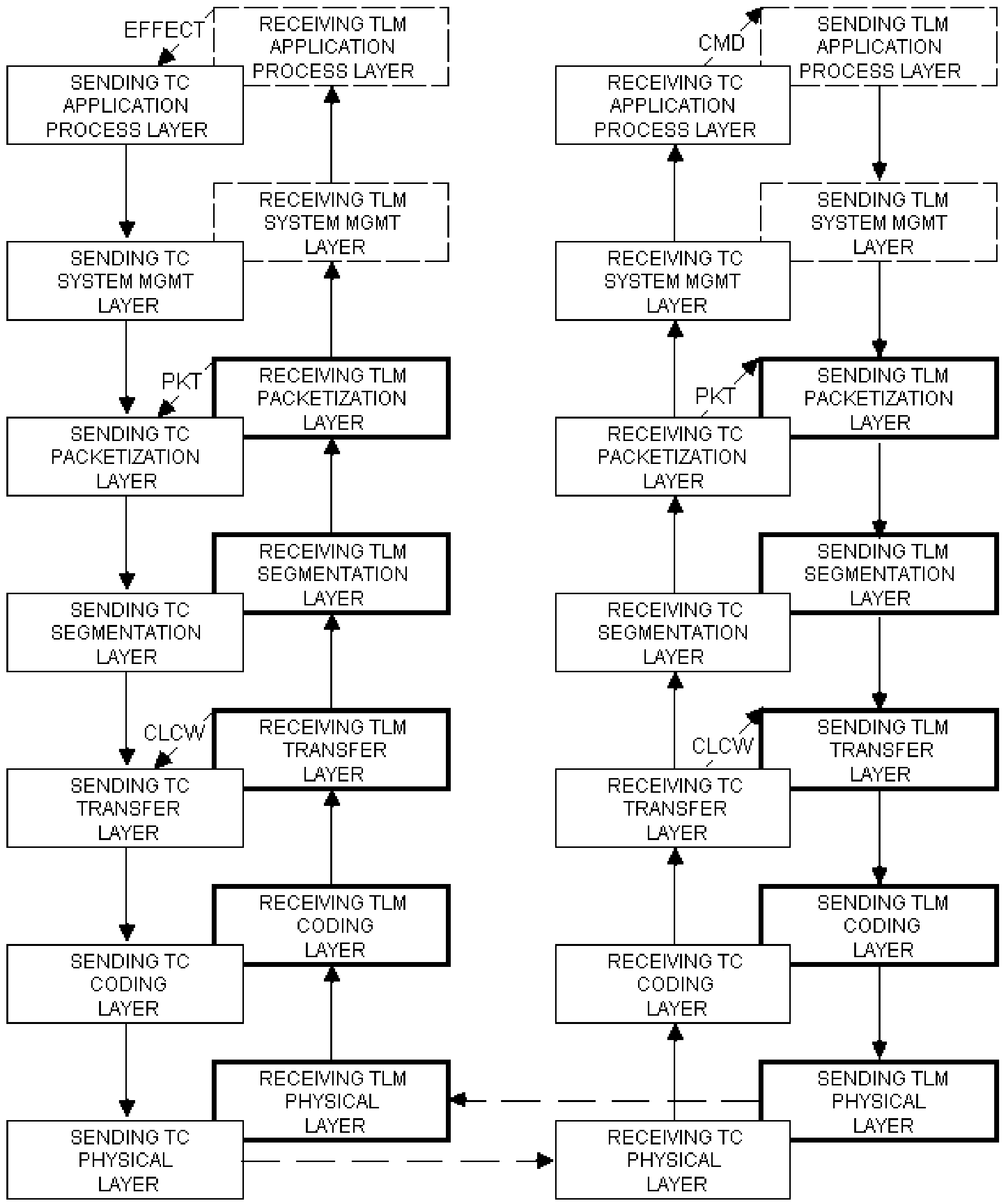

Figura 2.1. Relazione tra i sistemi di telemetria e di telecontrollo

una parte di intestazione e una di dati, quest'ultima di dimensione variabile; a sua volta il TC Packet è suddiviso in più *TC Segment*, anch'essi suddivisi in intestazione e campo dati. Ciascun TC Segment viene poi incapsulato in un

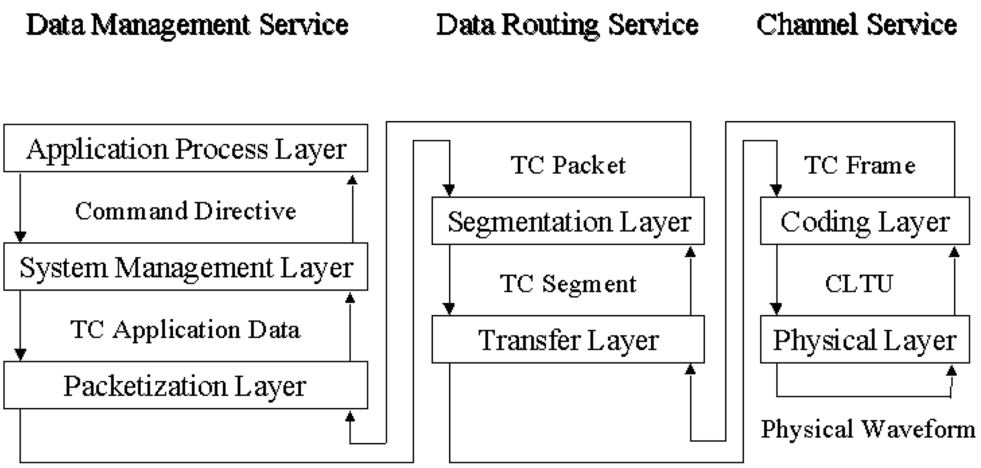

Figura 2.2. Architettura stratificata del telecomando

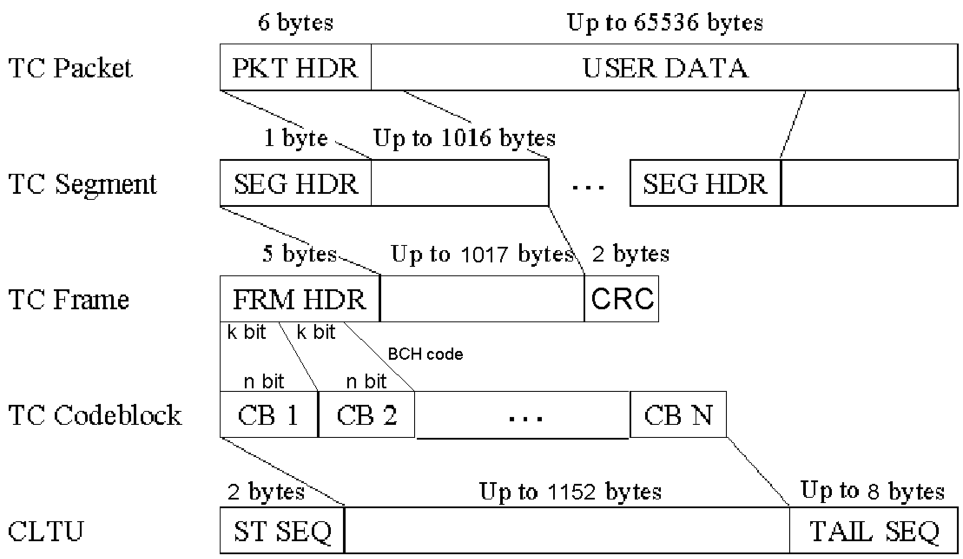

Figura 2.3. Struttura dati del telecomando

TC Frame con l'aggiunta di un'intestazione; questi dati vengono suddivisi in gruppi di k bit, poi codificati in n bit mediante un codice a blocco. L'insieme dei bit così risultanti è chiamato *TC Codeblock*, al quale vengono aggiunti una sequenza d'inizio (*start*) ed una di coda (*tail*).

Di seguito verrà analizzata più in dettaglio l'architettura del Telecomando.

2.3 Telecomando: flusso dati al trasmettitore

2.3.1 Application Process Layer

Il trasmettitore di questo strato, si veda [5], contiene diversi processi applicativi; questi vengono inseriti dagli utenti tramite *workstation* e forniscono l'interfaccia uomo-macchina permettendo la comunicazione con altri processi. Essi sono costituiti da una serie di direttive di comando specificate in un linguaggio ad alto livello e da richieste per il controllo della ricezione dei comandi. La gestione della consegna end-to-end dei comandi è fornita da un apposito processo applicativo che interfaccia lo strato in esame con quelli sottostanti, offrendo un servizio di traduzione delle sintassi tra i due livelli. Le strutture dati devono essere in grado di portare informazioni in modo bidirezionale attraverso l'interfaccia tra lo stato dei processi applicativi e quello sottostante di gestione del sistema. L'Application Process Layer riceve dall'utente informazioni sulla programmazione della missione e manda al ricevitore di bordo, tramite i livelli sottostanti, istruzioni di controllo necessarie per l'esecuzione dei comandi. Inoltre invia direttive di comando multiutente richieste per lo stato delle consegne e mantiene un data base sullo stato della missione, indicando la disponibilità delle risorse di sistema. Ulteriori funzioni riguardano la possibilità di sospendere, riprendere o abortire il trasferimento dei dati con priorità sull'esecuzione.

2.3.2 System Management Layer

Questo strato, si veda [5], si occupa di tradurre i comandi espressi mediante linguaggi di alto livello in un linguaggio di controllo utilizzato dallo strato sottostante ed analizza le direttive di comando predisponendole al trasporto. I dati delle applicazioni di comando vengono formattati in modo da poter essere inseriti direttamente nel campo dati del TC Packet, la cui struttura dettagliata sarà analizzata nel § 2.3.3. Il System Management Layer segnala agli strati inferiori la tipologia dei dati delle applicazioni, cioè se devono essere utilizzati come messaggi di comandi individuali o composti. Inoltre specifica la classe di servizio e tiene traccia dell'insieme di comandi traducendo i nomi utente in identificatori che possono essere inseriti nelle unità dati sottostanti. Un ulteriore compito è quello di specificare i tempi delle finestre e le procedure di recupero utilizzate in caso di interruzioni del servizio durante la trasmissione. Questo strato si occupa anche di inviare richieste sullo strato del trasporto e sulle misure di sicurezza e crea apposite istruzioni di recupero dopo aver analizzato i rapporti degli strati inferiori.

2.3.3 Packetization Layer

Il suo compito, si veda [5], è quello di creare il TC Packet incapsulando i dati provenienti dallo strato superiore mediante l'inserimento delle informazioni di controllo. La struttura del pacchetto è costituita da un'intestazione primaria, una secondaria opzionale e dal campo dati, come illustrato in figura 2.4.

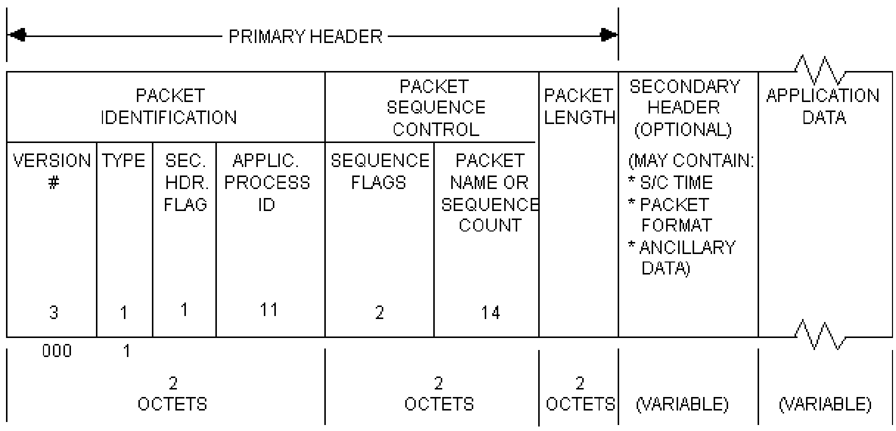

Figura 2.4. Formato del TC Packet

I campi indicati hanno il seguente significato:

1. *Primary Header*: è un campo costituito da sei ottetti suddivisi nei seguenti sottocampi:

(a) *Packet Identification*: contiene il numero della versione, il tipo (settato a '0' se è un pacchetto per la telemetria, '1' se è per il telecomando), il flag di intestazione secondaria, che ne indica la presenza nel caso sia settato, e l'identificatore del processo applicativo. Quest'ultimo (11 bit) indica il processo applicativo ricevente in un particolare *"spacecraft"*.

(b) *Packet Sequence Control*: contiene una coppia di bit che serve ad identificare la posizione del pacchetto all'interno del flusso dati ('01'⇒ primo pacchetto del flusso; '00'⇒ pacchetto di continuazione del flusso; '10'⇒ ultimo pacchetto del flusso; '11' ⇒ pacchetto singolo).

Inoltre contiene un contatore di sequenze (14 bit) che permette di identificare un pacchetto rispetto agli altri all'interno di una sessione di telecomando.

(c) *Packet Length* : contiene la lunghezza del pacchetto, tra il primo bit dell'intestazione secondaria e l'ultimo del pacchetto, espressa come (*numero di ottetti*)−1; quella massima è 65536. Possono essere utilizzati pacchetti molto lunghi, ma questi possono presentare particolari problemi in termini di monopolio del collegamento ed aggiungere una maggiore complessità ai processi a terra.

2. *Secondary Header*: è un campo opzionale, di lunghezza variabile, costituito da un numero intero di ottetti, con lo scopo di codificare i dati ausiliari, come ad esempio l'indirizzo del trasmettitore, il formato del campo dati, ecc., che possono essere necessari per interpretare i dati contenuti nel pacchetto. Se questo campo è presente occorre settare a '1' il corrispondente flag nell'intestazione primaria.

3. *Application Data*: contiene le informazioni del System Management Layer. Gli utenti possono utilizzare il formato più conveniente purchè la lunghezza totale sia un numero intero di ottetti minore o uguale alla massima lunghezza del campo definita nel Packet Length (punto 1c). A discrezione dell'utente, in questo campo è possibile includere un codice per la rivelazione degli errori, in modo da verificare la totale integrità

del pacchetto.

Il Packetization Layer si occupa anche delle misure di sicurezza, definendo il concetto di protezione dei dati del telecomando. Questa può essere fornita secondo due modalità:

- *fisica*: in questo caso solo un personale ristretto ha la possibilità di accedere ai dati ed è impraticabile nelle trasmissioni spaziali;
- *logica*: i comandi vengono trasformati in modo da renderne difficile l'interpretazione; per le trasmissioni spaziali si utilizza questo tipo.

Per la protezione logica possono essere utilizzate due tecniche:

- *autenticazione criptata*: riconosce il comando della sorgente autorizzata in modo che un utente non autorizzato non possa modificare le informazioni. Il trasmettitore genera un'unica parola di autenticazione che accompagna ogni blocco trasmesso in chiaro: il ricevitore la riconosce e ritrasmette a terra un messaggio in chiaro per la conferma di riconoscimento;
- *criptaggio dati*: un utente non autorizzato non può interpretare il significato dei comandi; il trasmettitore cripta i dati applicando uno speciale algoritmo che sarà decriptato da un processo complementare al ricevitore.

2.3.4 Segmentation Layer

Questo strato, si veda [3], riceve dallo strato sovrastante i TC Packet di lunghezza variabile e li scompone creando i TC Segment che dovranno essere compatibili con l'inserzione nel campo dati del TC Frame. Questo strato è provvisto di *Multiplexer Access Points* (MAPs) a cui vengono assegnati pacchetti individuali; ognuno viene suddiviso in segmenti compatibili con il campo dati del frame. In ciascun segmento viene inserita l'intestazione di controllo della sequenza e l'identificatore di MAP; i pacchetti vengono organizzati in modo che possano condividere un canale virtuale (VC, *Virtual Channel*). Viene inoltre effettuato un controllo di flusso e mantenuto lo stato di disponibilità di un particolare VC.

Il formato del TC Segment è costituito da un'intestazione e da un campo dati, come illustrato in figura 2.5.

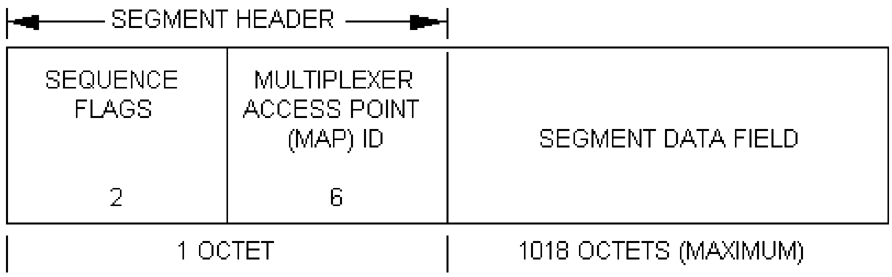

Figura 2.5. Formato del TC Segment

In particolare si ha:

1. *Segment Header*: contiene i seguenti due campi:

   (a) *Sequence Flags*: è un campo di due bit che indica la posizione sequenziale del TC Segment ('01'⇒ primo segmento del TC Packet su un MAP; '00'⇒ segmento di continuazione del TC Packet su un MAP; '10'⇒ ultimo segmento del TC Packet su un MAP; '11'⇒ assenza di segmentazione).

   (b) *MAP Identifier*: questo campo è di 6 bit, quindi è possibile distinguere 64 MAP; non è richiesto che i MAP siano numerati consecutivamente e se non vengono utilizzati su un particolare VC questi 6 bit devono essere settati ad un valore costante.

2. *Segment Data Field*: questo campo ha lunghezza variabile con valore massimo pari alla massima lunghezza del campo dati del TC Frame meno un ottetto.

2.3.5 Transfer Layer

Questo strato, si veda [3], utilizza un protocollo di telecomando ad anello chiuso mediante la tecnica di tipo *Go-Back-N*, la quale si basa su di un insieme di procedure sincronizzate, denominate *Command Operation Procedure* (COP), costituite dalla coppia:

- *Frame Operation Procedure* (FOP): viene eseguita al trasmettitore e permette la supervisione e il trasferimento dei TC Frame;
- *Frame Acceptance and Reporting Mechanism* (FARM): viene eseguita al ricevitore e permette di ricevere i TC Frame, trasmettendo le relative conferme (si veda quanto esposto nel § 2.4.3).

Attualmente è stata definita soltanto la prima versione, la quale viene indicata come COP-1: essa è in grado di fornire i seguenti due tipi di servizi:

- *Sequence-Controlled Service*: viene utilizzato per frame di tipo "AD" e "BC" (si vedano i successivi punti 1b e 1c);
- *Expedited Service*: viene utilizzato solo per frame di tipo "BD" (si vedano i successivi punti 1b e 1c).

All'inizio della missione è possibile impostare la dimensione delle finestre del protocollo Go-Back-N. Occorre notare che non si tratta di un Go-Back-N classico, in quanto la dimensione della finestra di trasmissione non deve essere necessariamente pari ad uno, ma può essere impostata fino ad un massimo pari a PW_{RX} e comunque minore di 256 (si veda la figura 2.14). Il formato del frame è costituito da un'intestazione, da un campo dati e da uno di controllo di errore, come illustrato in figura 2.6. Essi hanno il seguente significato:

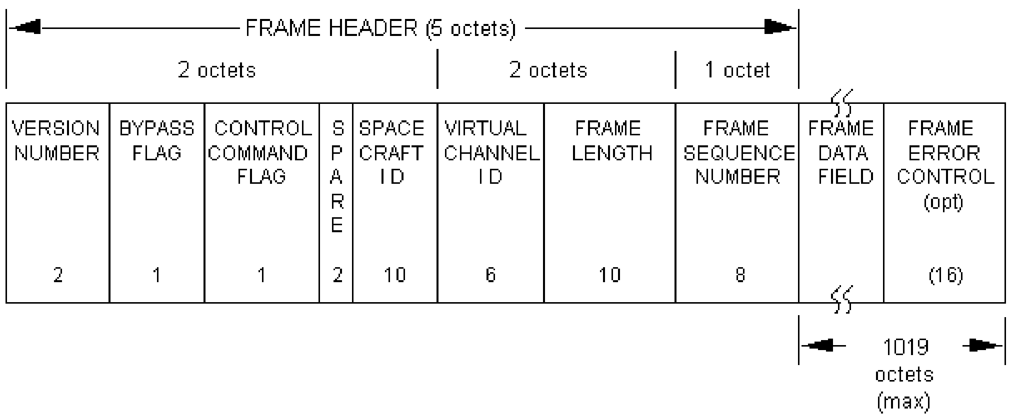

Figura 2.6. Formato del TC Frame

1. *Frame Header*: questo campo è costituito da cinque ottetti suddivisi nei seguenti sottocampi:

    (a) *Version Number*: al momento risulta definita soltanto la versione 1, indicata con '00'.

    (b) *Bypass Flag* : indica il tipo di frame che viene utilizzato. Se posto a '0', indica che si tratta di un TC Frame di tipo A, cioè che è soggetto al normale controllo di accettabilità da parte del FARM; viceversa, se posto a '1', indica un TC Frame di tipo B, cioè il controllo viene evitato. Ovviamente è possibile trasmettere sullo stesso VC entrambi i tipi di frame.

    (c) *Control Command Flag* : indica il contenuto dei frame che vengono trasmessi. Se posto a '0', indica che si tratta di un frame di dati (D); viceversa, se posto a '1', indica un frame di comando, cioè nel campo dati del frame sono contenute informazioni di controllo utilizzate per settare i parametri del FARM presente a bordo. Attualmente i frame di comando possono essere solo di tipo B, mentre quelli relativi ai dati sia di tipo A che B.

    (d) *Reserved Spares*: spazio riservato per applicazioni future.

    (e) *Spacecraft Identifier*: contiene un codice di identificazione dello *"spacecraft"*.

    (f) *Virtual Channel Identifier*: permette di identificare fino a 64 VC, ciascuno con la propria conferma di ricezione del frame. Ogni VC può essere poi connesso a 64 MAP, presenti nello strato di segmentazione.

    (g) *Frame Length*: indica la lunghezza del frame, espressa in numero di ottetti meno uno, inclusa l'intestazione e l'eventuale campo di controllo di errore.

    (h) *Frame Sequence Number "N(S)"*: rappresenta un contatore binario, crescente, modulo 256, indicato con N(S); esso viene utilizzato solo nei frame di tipo A, altrimenti viene lasciato a zero. Il numero dei contatori è pari al numero dei VC.

2. *Frame Data Field*: è un campo di lunghezza variabile, ma costituito da un numero intero di ottetti. La lunghezza massima è pari a 1019 ottetti (1017 se è presente il campo opzionale sul controllo d'errore). Questo campo può contenere:
    - *unità dati*: pacchetti, segmenti, parti o aggregati di essi;
    - *comandi di controllo*: sono definiti due soli tipi di comandi:
        - *UNLOCK*: sblocca la situazione di stallo che può verificarsi al ricevitore ed è costituito da un singolo ottetto di tutti zeri;
        - *SET V(R)*: viene utilizzato per impostare sul ricevitore il valore iniziale del contatore V(R) di bordo[1] ed è costituito da tre ottetti aventi il seguente formato:

          10000010 00000000 xxxxxxxx

          dove le "x" corrispondono al valore di V(R).

3. *Frame Error Control Field* : è un campo opzionale, costituito da 16 bit. Viene utilizzato per la sola rivelazione degli errori, qualora si vogliano ottenere elevate prestazioni. Il codice utilizzato è un codice a blocco, binario e sistematico del tipo (n,n-16), con polinomio generatore:

   $$g(x) = x^{16} + x^{12} + x^5 + 1 \tag{2.1}$$

   Gli schemi del codificatore e decodificatore sono illustrati in figura 2.7. Nello schema relativo al codificatore, i registri a scorrimento vengono inizializzati a '1'; entrano poi (n-16) bit con gli interruttori A e B chiusi e C aperto. Successivamente avvengono altri 16 colpi di clock con A='0', B aperto e C chiuso.

   Nello schema del decodificatore, i registri vengono inizializzati a '1'; entrano poi n bit, di cui (n-16) con B aperto e 16 con B chiuso.

Si effettuerà ora una breve trattazione sulle prestazioni, si veda [1], che vengono richieste a livello del frame. Si fa l'ipotesi di utilizzare un canale binario simmetrico (BSC: *Binary Symmetric Channel*) con rumore gaussiano bianco

[1] V(R) è il contatore del FARM: contiene il valore del numero di sequenza che sarà presente nell'intestazione del prossimo frame di tipo A che deve essere ricevuto.

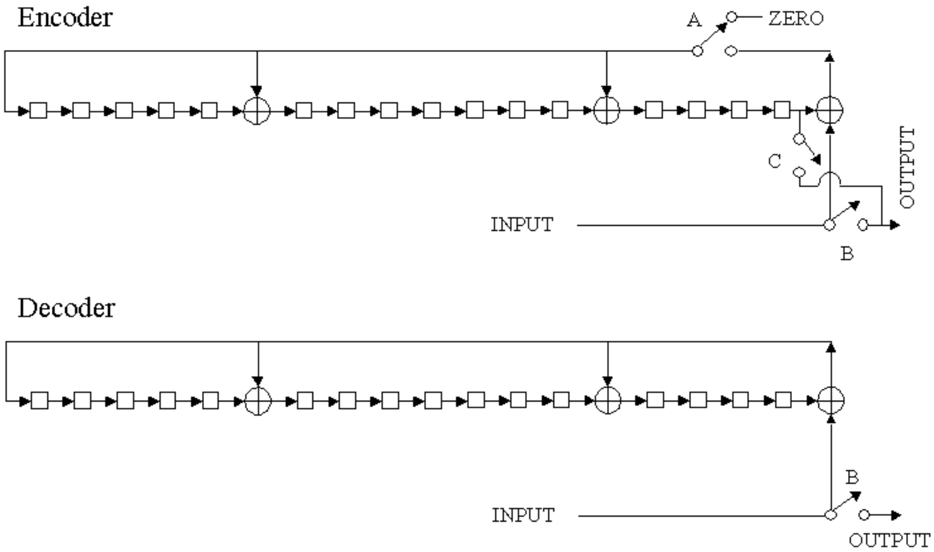

Figura 2.7. Schemi dei circuiti di codifica e decodifica per il controllo d'errore nel TC Frame

additivo (AWGN: *Additive White Gaussian Noise*); si definiscono inoltre:
- p: probabilità di errore sul bit del canale BSC;
- n: numero di bit in un TC Codeblock (questa struttura verrà illustrata nel § 2.3.6, figura 2.10;
- N: numero di TC Codeblock presenti in un TC Frame.

Vengono analizzati due casi:

- *Prestazioni sul rifiuto dei frame*: viene richiesta una probabilità di rifiuto del frame di 10^{-3}.

  Possono essere utilizzate due modalità : rivelazione di errori tripli, detta modo TED (*Triple Error Detection*), e correzione di errori singoli, detta modo SEC (*Single Error Correction*).

  La probabilità di rifiuto del frame nel modo TED vale:

$$P_{F,TED} = P_{S,TED} + (1 - P_{S,TED})P_{C,TED} \tag{2.2}$$

  dove:

$$P_{S,TED} = 1 - (1 - p)^{16} \tag{2.3}$$

è la probabilità di rifiuto della sequenza di inizio del CLTU (*Command Link Transmission Unit*), che verrà trattato nel § 2.3.6, figura 2.9, mentre

$$P_{C,TED} = 1 - [(1-p)^n]^N \tag{2.4}$$

è la probabilità che uno o più TC Codeblock siano errati.

La probabilità di rifiuto del frame nel modo SEC vale:

$$P_{F,SEC} = P_{S,SEC} + (1 - P_{S,SEC})P_{C,SEC} \tag{2.5}$$

dove:

$$P_{S,SEC} = 1 - [(1-p)^{16} + 16p(1-p)^{15}] \tag{2.6}$$

e

$$P_{C,SEC} = 1 - [(1-p)^n + np(1-p)^{n-1}]^N \tag{2.7}$$

hanno lo stesso significato di prima.

- *Prestazioni sui frame errati non rivelati*: viene richiesta una probabilità di non rivelare frame errati pari a 10^{-9}.

  Anche in questo caso esistono i due modi, TED e SEC. I risultati sono illustrati nel grafico di figura 2.8. Come si può osservare, le prestazioni richieste per gli errori non rivelati sono soddisfatte sia dal modo TED che da quello SEC utilizzando un canale con $p = 10^{-5}$ senza ulteriori precauzioni; se si vogliono ottenere prestazioni migliori, si può aggiungere ad ogni frame un codice ciclico come descritto al punto 3 di questo paragrafo. Aggiungendo infatti tale codice, sempre con $p = 10^{-5}$, la probabilità di non rivelare un frame errato diventa di 10^{-9} utilizzando il modo SEC, ed ancora meglio con il modo TED.

Occorre notare che spesso i metodi utilizzati per migliorare la probabilità di rifiuto del frame, come il modo SEC, causano una riduzione delle prestazioni per gli errori non rivelati e viceversa: sarà quindi necessario determinare un compromesso tra i due modi.

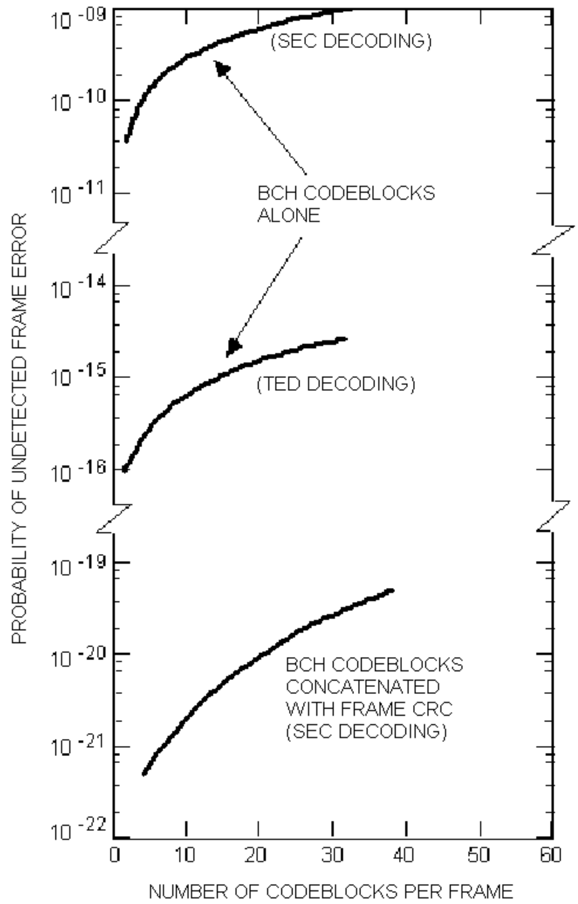

Figura 2.8. Grafici relativi alle prestazioni sui TC Frame errati non rivelati

2.3.6 Coding Layer

Questo strato, si veda [2], codifica i dati per ridurre gli effetti del rumore, ricevendo dallo strato superiore il TC Frame, il quale verrà incluso in un *Command Link Transmission Unit* (CLTU); manda poi allo strato inferiore i CLTU realizzati.

Il formato del CLTU è costituito da una sequenza di inizio, una serie di codeblock e da una sequenza di fine, come illustrato in figura 2.9.
I rispettivi campi hanno il seguente significato:

1. *Start Sequence*: segnala l'inizio dei TC Codeblock; è costituita da una sequenza di 16 bit la cui funzione di autocorrelazione possiede bassi

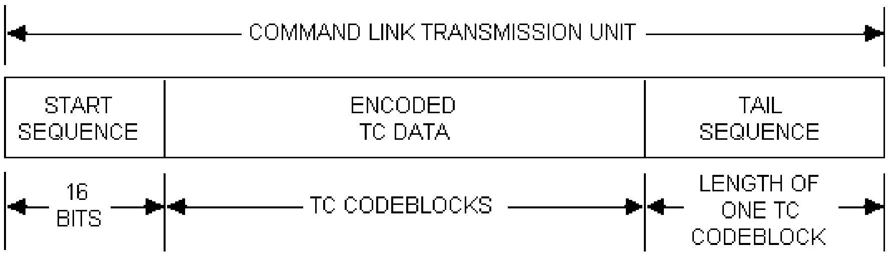

Figura 2.9. Formato del CLTU di telecomando

lobi laterali. La sequenza utilizzata è:

(bit 0) 11101011 10010000 (bit 15)

2. *Encoded TC Data*: contiene una serie di TC Codeblock.

3. *Tail Sequence*: è costruita appositamente per essere una sequenza non correggibile che delimita la fine del CLTU. Tale sequenza deve avere la stessa lunghezza dei TC Codeblock (da 40 a 64 bit, come si vedrà in seguito); è costituita da n ottetti, di cui i primi $(n-1)$ sono costituiti dalla sequenza 11000101 che si ripete, mentre l'ultimo ottetto è sempre la sequenza 01111001.

Viene ora analizzata la struttura del TC Codeblock, illustrata in figura 2.10, costituita da un campo di informazione e da uno di controllo d'errore.

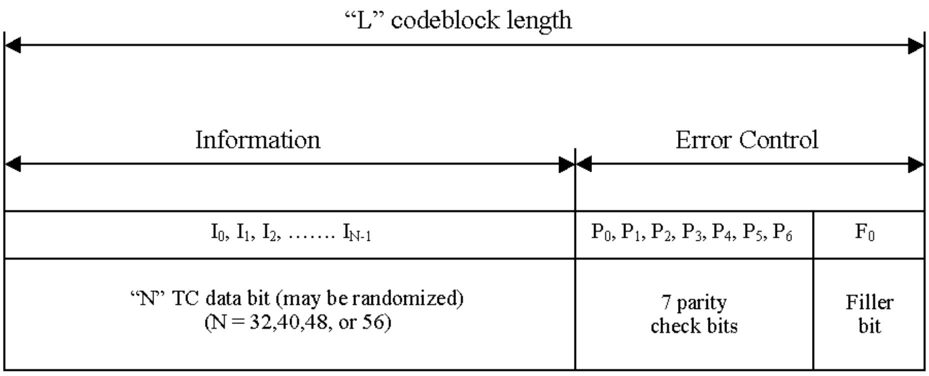

Figura 2.10. Formato del TC Codeblock

1. *Information*: è costituito da k bit di informazione, dove k è fissato e può essere scelto tra 32, 40, 48 e 56 bit. La lunghezza del TC Codeblock consigliata è pari a n=64 bit, cioè k=56 bit. Qualora sia necessario, è possibile casualizzare i bit di informazione, secondo la modalità descritta di seguito.

2. *Parity Check Bits*: contiene i bit per il controllo di parità; la procedura di codifica utilizzata viene descritta successivamente.

3. *Appended Filler Bit*: è un unico bit che viene utilizzato con il solo scopo di ottenere dei TC Codeblock con un numero intero di ottetti; questo bit deve essere sempre settato a zero.

Vengono ora analizzate dapprima la procedura di casualizzazione e, successivamente, quella di codifica.

1. *Procedura di casualizzazione*: per mantenere la sincronizzazione dei bit (o simboli), il segnale ricevuto deve avere una minima densità di transizione dei bit; se questa non è assicurata viene richiesta la casualizzazione, altrimenti il suo utilizzo è opzionale. Come illustrato in figura 2.11, i dati casualizzati si ottengono in uscita da una porta EX-OR, con in ingresso i dati da casualizzare e la sequenza casuale; al ricevitore, i dati casualizzati ricevuti vengono mandati alla porta EX-OR con la stessa sequenza casuale, ottenendo perciò i dati originali. Tale casualizzatore viene anche chiamato BTG (*Bit Transition Generator*): esso utilizza una sequenza casuale generata dal polinomio:

$$h(x) = x^8 + x^6 + x^4 + x^3 + x^2 + x + 1 \tag{2.8}$$

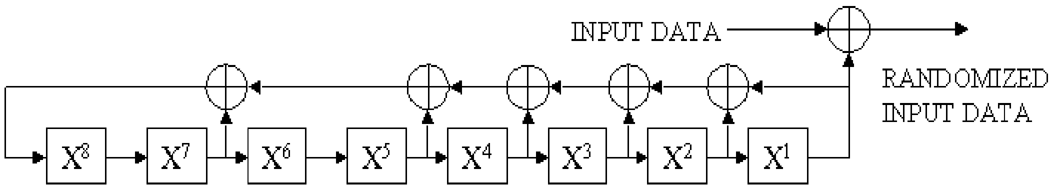

Figura 2.11. Schema del casualizzatore per il telecomando

che si ripete dopo 255 bit. Occorre inoltre osservare che il BTG è preselezionato con tutti '1' e poi viene messo in EX-OR bit per bit.

2. *Procedura di codifica*: questa procedura genera sempre 7 bit di parità utilizzando in ingresso sempre 56 bit di informazione; se vengono utilizzati un numero inferiore di bit in ingresso, cioè 32, 40 o 48 bit invece di 56 bit, può essere utilizzato lo stesso algoritmo aggiungendo un "riempimento virtuale" (dei bit a '0') che non vengono emessi dal codificatore e non vengono trasmessi.

   Il codice utilizzato è un (63,56) *Bose-Chaudhuri-Hocquenghem* (BCH) modificato, con polinomio generatore:

$$g(x) = x^7 + x^6 + x^2 + 1 \tag{2.9}$$

   Lo schema del codificatore è riportato in figura 2.12. I regisri a scor-

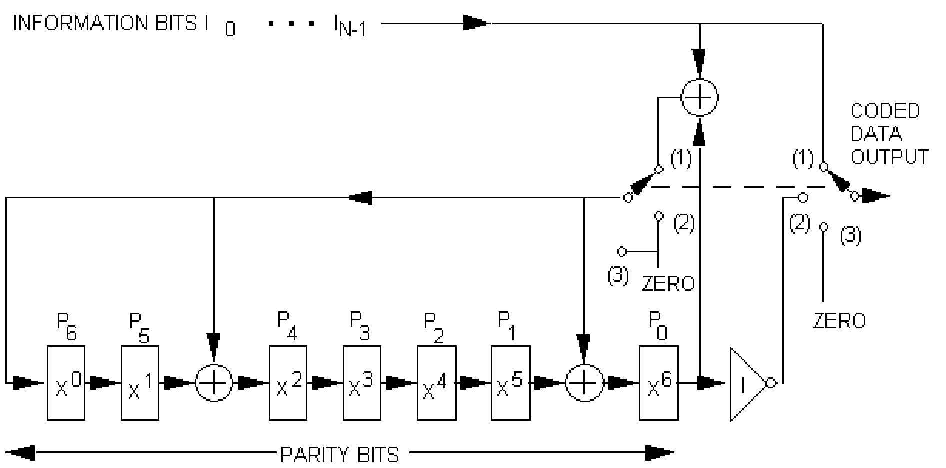

Figura 2.12. Schema del codificatore BCH modificato

rimento devono essere inizializzati tutti a zero; gli interruttori restano in posizione (1) durante la trasmissione dei k bit di dati, poi passano in posizione (2) per i sette bit di parità, ed infine in posizione (3) per gli eventuali "bit di riempimento". Essi differiscono dal "riempimento virtuale" visto in precedenza, in quanto sono costituiti da una sequenza di zeri e uno alternati che inizia con '0': questi bit vengono posti nell'ultimo ottetto del campo *Information* del TC Codeblock (punto 1 di questo paragrafo), qualora i dati di ingresso non riempano esattamente un numero intero di TC Codeblock.

2.3.7 Physical Layer

Questo strato, si veda [2], riceve dallo strato superiore un buffer di bit corrispondente ad un CLTU e trasmette al ricevitore le forme d'onda modulate a radiofrequenza. Inoltre stabilisce il cammino fisico con lo *"spacecraft"* secondo apposite modalità , chiamate CMM (*Carrier Modulation Modes*), ed irradia serialmente un buffer di bit di dati in accordo con specifiche procedure denominate PLOP (*Physical Layer Operation Procedures*).

Le strutture dati standard che vengono utilizzate in questo strato sono:

1. *Acquisition Sequence*: fornisce la sincronizzazione di simbolo iniziale. La lunghezza di tale sequenza deve essere scelta in base alle prestazioni richieste dalla missione, ma viene tuttavia consigliato l'uso di una lunghezza minima pari a 16 ottetti (non è necessario che tale sequenza corrisponda ad una lunghezza pari ad un numero intero di ottetti). La sequenza di acquisizione è costituita da un flusso di '0' e '1' alternati.

2. *CLTU*: è la struttura dati ricevuta dallo strato superiore, che contiene i dati veri e propri che devono essere trasmessi. Ogni blocco del CLTU ha almeno due transizioni dati per ciascun TC Codeblock, ma se la sincronizzazione dei simboli dello *"spacecraft"* necessita di transizioni più frequenti, allora può essere adottatta la procedura di casualizzazione vista nel § 2.3.6 punto 1, oppure lo strato fisico deve adottare una tecnica per garantire transizioni sufficientemente frequenti per una sincronizzazione di simboli adeguata.

3. *Idle Sequence*: serve a mantenere la sincronizzazione di simbolo in assenza di CLTU. Essa è costituita da una sequenza alternata di '0' e '1', la cui lunghezza non è fissata.

Vengono ora illustrate le procedure standard utilizzate da questo strato; esse si suddividono in due tipi, CMM e PLOP. Le prime corrispondono a diversi stati di modulazione sulla portante a radiofrequenza che crea il canale fisico; si suddividono in:

- *CMM-1*: solo la portante non modulata;
- *CMM-2*: portante modulata con la sequenza di acquisizione;
- *CMM-3*: portante modulata con i dati;
- *CMM-4*: portante modulata con la sequenza di idle.

Durante una sessione di telecomando, vengono trasmessi allo *"spacecraft"* una serie di CLTU; la sessione inizia con l'applicazione della portante a radiofrequenza e termina con la sua rimozione. Il cammino a radiofrequenza viene attivato o disattivato mediante la selezione di appropriate PLOP.

Una PLOP consiste nell'applicazione sequenziale di vari CMM; ne esistono di due tipi:

- *PLOP-1*: viene utilizzata per la trasmissione di CLTU individuali, per i quali il decodificatore dello *"spacecraft"* è sempre forzato nello stato di inattività, disattivando il canale dopo la fine di ogni CLTU;
- *PLOP-2*: il canale non viene disattivato dopo ogni trasmissione di un CLTU. La fine di un CLTU viene segnalata dalla sequenza di fine, che posiziona il decodificatore nello stato di ricerca dopo ogni CLTU; il decodificatore viene forzato nello stato di inattività, disattivando il canale soltanto alla fine di una serie di CLTU. Quando si utilizza questa procedura, si consiglia di inserire sistematicamente una sequenza di idle minima (pari ad un ottetto) tra ogni CLTU.

Il tipo di PLOP utilizzato deve essere scelto all'inizio della missione.

2.4 Telecomando: flusso dati al ricevitore

Si è visto che il flusso dati al trasmettitore provoca, come ultimo effetto, la trasmissione di forme d'onda irradiate nell'etere a radiofrequenza. Queste ultime raggiungono così il ricevitore e ripercorrono, in senso inverso rispetto a prima, i sette strati precedentemente analizzati. Il ricevitore si preoccuperà di ricevere, interpretare ed eseguire i comandi trasmessi da terra e, quando richiesto, fornirà un rapporto di conferma al trasmettitore sfruttando il flusso dati della telemetria. Di seguito viene analizzato il percorso del flusso dati che giunge al ricevitore.

2.4.1 Physical Layer

Questo strato riceve dal trasmettitore le forme d'onda, modulate a radiofrequenza, che vengono irradiate: in pratica riceve e rileva la portante modulata. Successivamente esegue la demodulazione e la sincronizzazione dei simboli; a questo punto, determina lo stato del canale fisico (stato di attività o di inattività), esegue la rivelazione dei simboli e li manda allo strato superiore.

2.4.2 Coding Layer

Questo strato riceve da quello inferiore un flusso di simboli sincronizzato, ma con possibili errori. Inizialmente viene rilevata la sequenza di inizio: i codeblock che seguono risultano automaticamente sincronizzati essendo contigui.

Segnala, inoltre, l'inizio della validità dei codeblock, valutando la possibile presenza di errori nei blocchi, eventualmente correggendoli. Se viene rivelato un blocco errato non correggibile, il ricevitore lascia lo stato di decodifica ed entra in quello di ricerca di una nuova sequenza d'inizio, cessando il trasferimento dei blocchi allo strato superiore: segnala perciò la fine della validità dei dati. Qualora il segnale venga perso, viene abbandonato lo stato di decodifica entrando così in quello di inattività, cessando il trasferimanto dei blocchi e segnalando la fine della validità dei dati.

Allo strato superiore vengono passati i dati, cioè i blocchi che hanno già subito l'operazione di decodifica, "puliti", decodificati ed eventualmente decasualizzati (se richiesto): l'operazione di decasualizzazione viene applicata ai blocchi che hanno già subito la decodifica.

La procedura di decodifica utilizzata al ricevitore si basa sul codice (63,56) BCH modificato illustrato nel § 2.3.6 punto 2 e può essere utilizzata secondo due modalità:

- *Error-detecting mode (TED)*: permette di rivelare uno, due, o tre bit errati in ciascun codeblock (escluso il bit di riempimento);

- *Error-correcting mode (SEC)*: permette di correggere un bit errato e di rivelarne due.

2.4.3 Transfer Layer

Questo strato riceve dal trasmettitore i TC Frame con i comandi di controllo per poter elaborare gli ottetti puliti, cioè i frame di dati, provenienti dal Coding Layer; un segnale di "inizio dati" e uno di "fine dati" indica l'inizio o la fine della validità degli ottetti. Sui frame ricevuti, il Transfer Layer esegue la procedura di controllo della validità (sia sul tipo A che su quello B) e il controllo di accettabilità, con relativo meccanismo di rapporto di consegna, solo per i frame di tipo A. Questo strato, inoltre, elabora i frame che sono stati ritrasmessi e crea i *Command Link Control Word* (CLCW), cioè i rapporti di conferma, descriventi lo stato di accettabilità dei frame. Infine estrae il campo dati del TC Frame, cioè il TC Segment, e lo passa allo strato superiore.

Il CLCW è l'unica struttura dati del telecomando che viene ritrasmessa a terra, tramite il flusso dati della telemetria: essa fornisce un rapporto che descrive lo stato di accettabilità e riassemblamento del TC Frame. Sebbene non sia necessario che la velocità di riporto dei dati della telemetria uguagli quella di trasferimento dei dati del telecomando, è necessaria una minima velocità di campionamento del CLCW per le tipiche operazioni del COP. La struttura del CLCW è illustrata in figura 2.13.
Nel seguito è riportata la relativa descrizione dettagliata:

1. *Control Word Type*: questo bit viene sempre posto a zero in un CLCW.

2. *CLCW Version Number*: attualmente è definita solo la versione 1, indicata con '00'.

3. *Status Field*: è un campo lasciato di libero uso alle varie agenzie spaziali.

4. *COP In Effect*: questo campo indica la versione di COP utilizzato: attualmente è definita soltanto la versione COP-1.

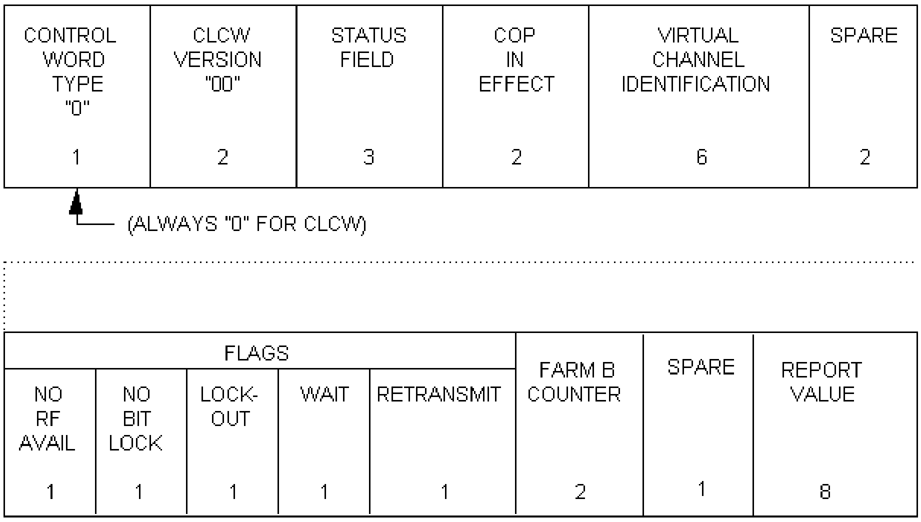

Figura 2.13. Formato del CLCW di telecomando

5. *Virtual Channel Identifier*: contiene l'identificatore del VC al quale il CLCW di rapporto è associato.

6. *Reserved Spares*: è un campo riservato ad applicazioni future: attualmente è settato a zero.

7. *Flags*:

   (a) *No RF Available*: se posto a '1' indica lo stato di pronto degli elementi a radiofrequenza, altrimenti il canale fisico non risulta disponibile.

   (b) *No Bit Lock*: è un indicatore della qualità delle prestazioni; indica, cioè, che gli strati di servizio di canale non stanno lavorando normalmente, ossia senza abbastanza energia di segnale per ottenere la sincronizzazione dei bit ed evitare un'elevata velocità di rifiuto del frame.

   (c) *Lockout*: viene settato a '1' ogni volta che viene ricevuto un frame di tipo A che viola i controlli di accettabilità del frame. Quando il FARM del ricevitore si trova nello stato di lockout rifiuta tutte le sotto-sequenze del frame che riceve, finché questa condizione non viene sbloccata.

(d) *Wait*: indica l'impossibilità di passare i dati dal Transfer Layer al Segmentation Layer in seguito a mancanza di memoria e/o di risorse di elaborazione al ricevitore. Tutti i frame di tipo A, presenti su un certo VC, vengono rifiutati dal FARM finché non termina la condizione di congestione.

(e) *Retransmit*: indica al trasmettitore che uno o più frame di tipo A sono mancanti o sono stati rifiutati dal FARM ed è perciò richiesta la ritrasmissione.

8. *FARM-B Counter*: contiene il valore contenuto nel contatore presente all'interno del FARM e viene incrementato ogni volta che viene accettato un frame di tipo B.

9. *Reserved Spare*: spazio riservato per applicazioni future.

10. *Report Value "N(R)"*: contiene il valore N(R) corrispondente al valore del contatore V(R) presente nel FARM . Indica il prossimo numero di sequenza del frame di tipo A atteso su di un particolare VC.

Viene ora brevemente illustrato il principio di funzionamento della finestra di ricezione, si veda [4], utilizzata nel protocollo *Go-Back-N* accennato nel § 2.3.5.

La finestra di ricezione viene schematicamente illustrata in figura 2.14, dove:

- W_{TX}: ampiezza della finestra di trasmissione;
- W_{RX}: ampiezza della finestra di ricezione, che deve essere sempre un numero intero pari;
- PW_{RX} *e* NW_{RX}: ampiezze delle finestre di ricezione positiva e negativa;
- *N(S)*: numero di sequenza del frame trasmesso;

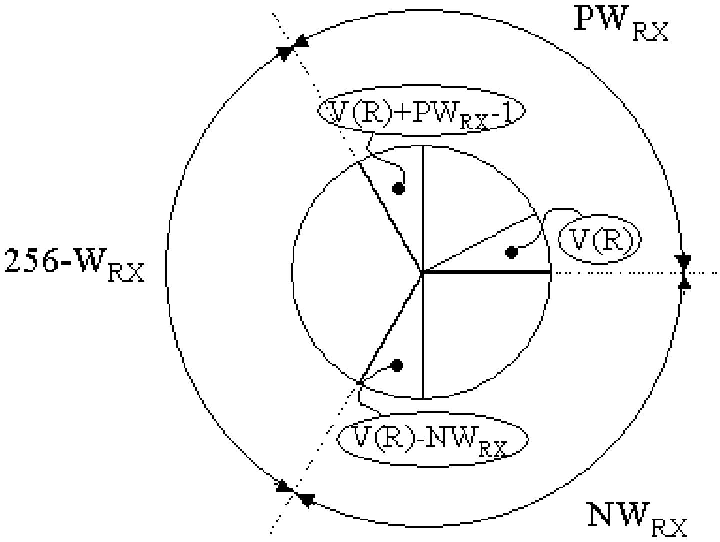

Figura 2.14. Finestra di ricezione del protocollo Go-Back-N

- *V(R)*: contatore del FARM che corrisponde al valore di N(S) del prossimo frame atteso.

Valgono inoltre le seguenti relazioni:

$$1 \leq W_{TX} \leq PW_{RX} \qquad e \qquad W_{TX} < 256 \tag{2.10}$$

$$2 \leq W_{RX} \leq 254 \tag{2.11}$$

$$PW_{RX} = NW_{RX} = \frac{W_{RX}}{2} \tag{2.12}$$

Il funzionamento della finestra di ricezione prevede quattro differenti condizioni di funzionamento:

1. $(V(R) + PW_{RX} - 1) < N(S) < (V(R) - NW_{RX})$

   Il frame ricevuto viene scartato ed il FARM va nello stato di lockout settando la rispettiva flag.

2. $N(S) = V(R)$

   Il frame ricevuto è quello atteso e viene accettato.

3. $V(R) < N(S) \leq (V(R) + PW_{RX} - 1)$

   Il frame ricevuto è fuori sequenza, ma non era ancora stato ricevuto; viene scartato e viene settata ad uno la flag di ritrasmissione.

4. $(V(R) - NW_{RX}) \leq N(S) < V(R)$

   Il frame ricevuto è fuori sequenza ed è già stato ricevuto; viene scartato, ma non viene fatta alcuna altra azione.

2.4.4 Segmentation Layer

Questo strato riceve da quello inferiore il campo dati del TC Frame (cioè il TC Segment), completo, in sequenza, senza parti mancanti o duplicate. Ordina i segmenti associati ai vari VC in accordo con gli identificatori di MAP e riassembla i segmenti associati ad un particolare MAP per ricostruire il pacchetto. Determina, inoltre, quando tutti i segmenti associati ad un particolare pacchetto sono stati ricevuti correttamente. Infine ricrea il pacchetto di telecomando e lo invia allo strato superiore.

2.4.5 Packetization Layer

Questo strato riceve da quello inferiore i pacchetti di telecomando ed informazioni sullo stato di instradamento dei dati; inoltre il trasmettitore gli fornisce le istruzioni di controllo sulle modalità per trasferire i dati allo strato superiore incluse le misure di sicurezza.

Estrae poi le informazioni contenute nel campo dati del pacchetto riassemblandole nello stesso ordine con cui sono state trasmesse. Analizza, inoltre, le istruzioni di controllo e le misure di sicurezza, e passa le applicazioni dati allo strato superiore quando risultano verificate appropriate condizioni di sistema. Il Packetization Layer, infine, manda allo strato superiore il contenuto dei pacchetti ed informazioni di stato relative alla correttezza, completezza e sequenzialità dei dati ricevuti; manda, poi, al trasmettitore i rapporti sullo stato di ricezione e riassemblamento dei pacchetti.

2.4.6 System Management Layer

Questo strato riceve da quello inferiore i dati, le istruzioni di controllo sulle condizioni di consegna e le informazioni sulla capacità di accettare ulteriori dati. Determina poi quando i dati sono pronti per la consegna e traduce i comandi ricevuti dal Packetization Layer in una sintassi interpretabile dall'Application Process Layer. Il System Management Layer formula poi i rapporti sullo stato di consegna, per il trasmettitore, e sulla possibilità di accettare ulteriori dati, per lo strato inferiore.

2.4.7 Application Process Layer

Al ricevitore di questo strato arrivano dallo *"spacecraft"* le condizioni sullo stato di quest'ultimo e su quello delle risorse; riceve inoltre dallo strato inferiore l'insieme di comandi da eseguire. Analizza poi le istruzioni di controllo ed esegue i comandi quando le loro condizioni vengono soddisfatte. Questo strato fornisce all'utente la possibilità di sospendere, riprendere ed abortire i comandi con maggiore priorità rispetto alla loro esecuzione; notifica inoltre, allo strato inferiore, l'impossibilità di accettare ulteriori consegne di comandi.

2.5 Telemetria

Lo scopo del sistema di telemetria, si veda [6], è quello di convogliare le informazioni di misura in modo trasparente ed affidabile da una sorgente di dati remota ad un utente situato a terra o nello spazio. Anche il sistema di telemetria, come quello di telecomando, è definito mediante lo stile stabilito dall'OSI (*Open System Interconnection*). Infatti, i servizi di comunicazione sono forniti da strati di protocolli e ogni strato fornisce un'interfaccia all'utente del servizio nello strato superiore. Un'interfaccia di servizio è definita in termini di primitive indipendenti dalla specifica implementazione, cioè si utilizza un modello astratto per lo scambio delle strutture dati e per il controllo dell'informazione.

L'architettura della telemetria prevede una suddivisione in quattro strati identificati dalle lettere A (strato più basso), B, C, D (strato più alto). Il progetto degli strati è comunque indipendente dal tipo di servizio che deve

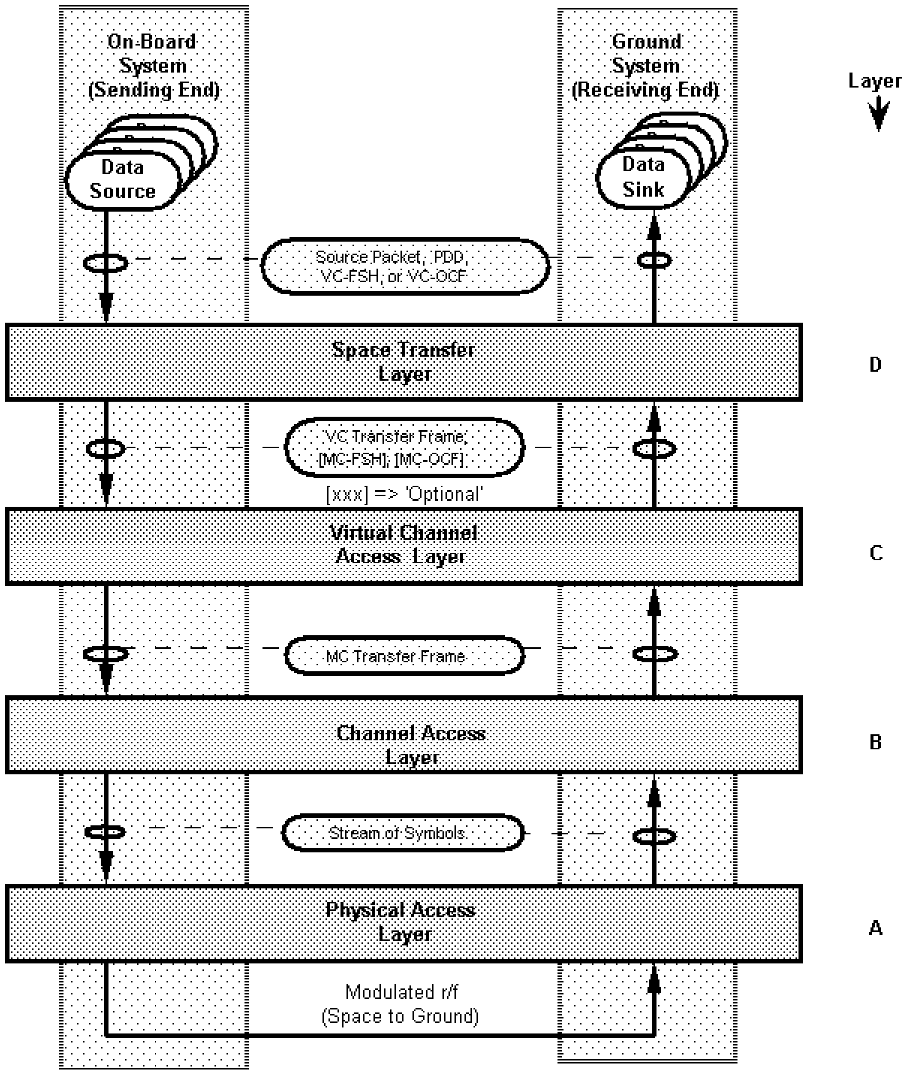

Figura 2.15. Architettura stratificata della telemetria

essere fornito: la struttura generale è rappresentata in figura 2.15. I servizi specificati nella telemetria presentano le seguenti caratteristiche:

- sono unidirezionali, in quanto lo *"spacecraft"* può spedire, ma non ricevere, mentre la stazione di terra può ricevere, ma non spedire;

- sono non confermati, cioè lo *"spacecraft"* non riceve la conferma che i dati da esso spediti siano stati ricevuti, così si evitano i ritardi implicati con un servizio confermato;

- possono essere implementati come servizi asimmetrici, cioè gli strati di un sottosistema nello spazio e a terra possono essere strutturati differentemente;
- necessitano di primitive di servizio.

I servizi, inoltre, possono essere di due tipi:

- *servizi accodati*: ogni unità dati spedita è posta in una coda FIFO (*First In First Out*); la caratteristica principale è che tutte le unità dati sono trasferite solo una volta, anche se vi sono errori;
- *servizi "bufferizzati"*: ogni unità dati spedita è posta in un buffer che può contenerne una sola; la caratteristica principale di questa struttura è che è a divisione di tempo. Il contenuto del buffer viene spedito in un istante di tempo (costante o aperiodico) determinato dal fornitore del servizio e non dall'utente. A differenza del precedente tipo, qui una particolare unità dati potrebbe essere spedita più volte o non essere spedita.

Le risorse trasmissive della telemetria sono condivise, cioè più utenti possono accedere contemporaneamente al canale dati in down-link e possono essere maneggiati tipi di dati differenti; il CCSDS fornisce, per controllare il flusso di dati, un metodo di canalizzazione virtuale. Una possibile architettura del sistema potrebbe prevedere, ad esempio, l'assegnazione dei pacchetti di *payload* su un canale virtuale e quelli di controllo e telecomando multiplati su di un secondo canale virtuale; i VC possono anche essere utilizzati per separare il flusso di pacchetti in tempo reale dagli altri.

Un esempio del flusso dati della telemetria è rappresentato in figura 2.16. Di seguito vengono analizzati i vari strati della telemetria, da quello di più alto livello a quello più basso.

2.5.1 Layer D: Space Transfer Layer

Questo strato, si veda [9], fornisce l'accesso a quattro sevizi di trasferimento, di tipo spazio→terra o spazio→spazio, utilizzando un canale virtuale VC. I

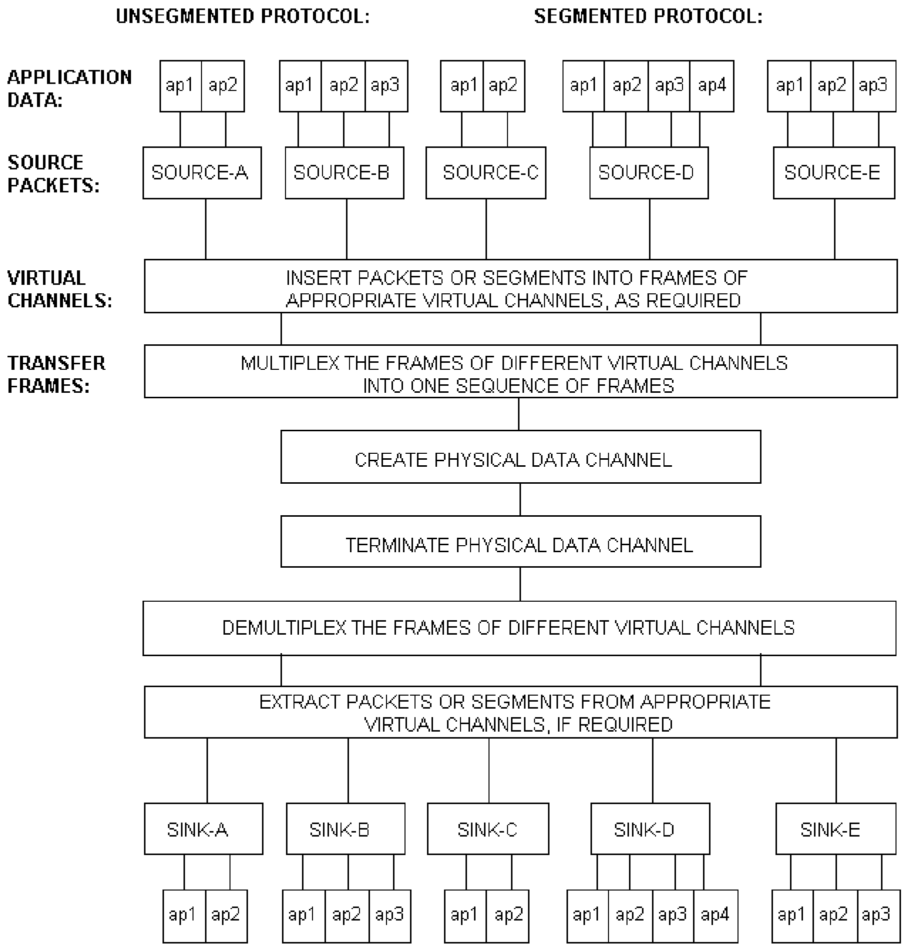

Figura 2.16. Esempio del flusso dei dati della telemetria

suddetti servizi sono i seguenti:

1. *Service Packet Transfer Service (SP-XFR)*: esso trasferisce una sequenza di pacchetti di lunghezza variabile da una sorgente nello spazio ad una o più applicazioni sulla terra (si ricorda che questi servizi sono unidirezionali). Si tratta di un servizio che non garantisce completezza, tuttavia può rilevare delle mancanze nella sequenza di unità dati spedite verso una stazione ricevente. Si tratta di un servizio modellizzato come una coda che collega il punto d'accesso al servizio SAP (*Service Acces Point*) per un dato canale virtuale dell'SP-XFR a bordo con il corrispondente SP-XFR SAP a terra, come mostrato in figura 2.17; esiste

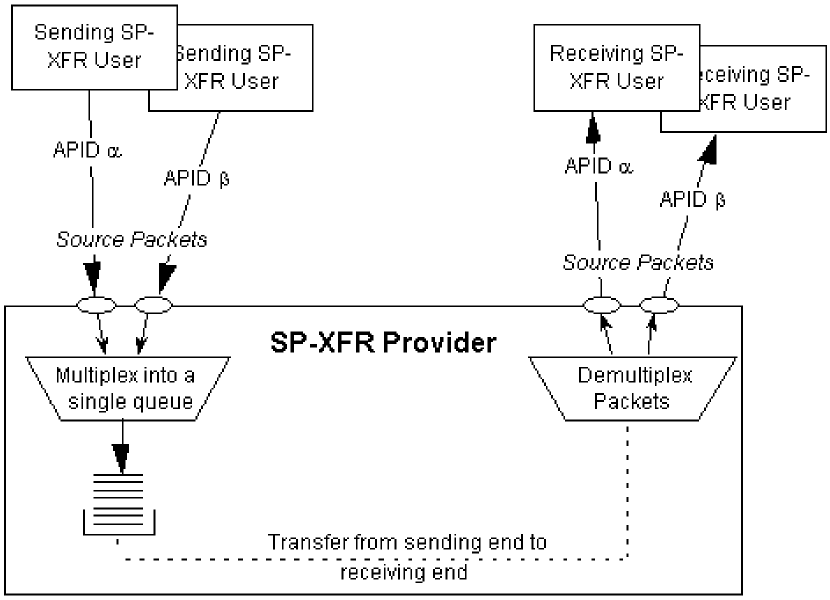

Figura 2.17. Esemplificazione del servizio di accodamento in un fornitore di servizi SP-XFR

una coda separata per ogni canale virtuale. I pacchetti sorgente spediti vengono, inoltre, trasferiti in ordine, mentre non vi sono specifiche in merito alle relazioni di questi servizi tra due canali virtuali.

Le primitive utilizzate da questo servizio sono:

- *SP-XFR.request*: per richiedere la multiplazione in uno specifico canale virtuale dell'unità dati di servizio SP-XFR_SDU (*SP-XFR Service Data Unit*);
- *SP-XFR.indication*: per spedire una SP-XFR_SDU.

2. *Privately Defined Data Transfer Service (PDD-XFR)*: esso trasferisce PDD di lunghezza fissa da bordo alla stazione di terra: non garantisce la completezza, ma rileva mancanze nella sequenza. Può essere modellizzato con un servizio accodato per ogni VC collegando il PDD-XFR SAP di bordo con il corrispondente PDD-XFR SAP a terra.

   Questo servizio utilizza due primitive:

- *PDD-XFR.request*: per richiedere che venga trasferita una PDD-XFR_SDU;
- *PDD-XFR.indication*: per spedire una PDD-XFR_SDU.

3. *VC Frame Secondary Header Service (VC_FSH)*: esso fornisce il trasferimento sincronizzato di FSH di lunghezza fissa su ciascun frame di un VC. Il servizio è unidirezionale, mantiene la sequenzialità, ma non garantisce la completezza rilevando, tuttavia, mancanze nella sequenza.

   Questo servizio può essere modellizzato con un buffer, il cui contenuto viene trasferito dal VC_FSH SAP di bordo a quello di terra, come mostrato in figura 2.18; esiste una coda separata per ogni canale virtuale. I VC_FSH vengono trasmessi in modo sequenziale: in particolare viene

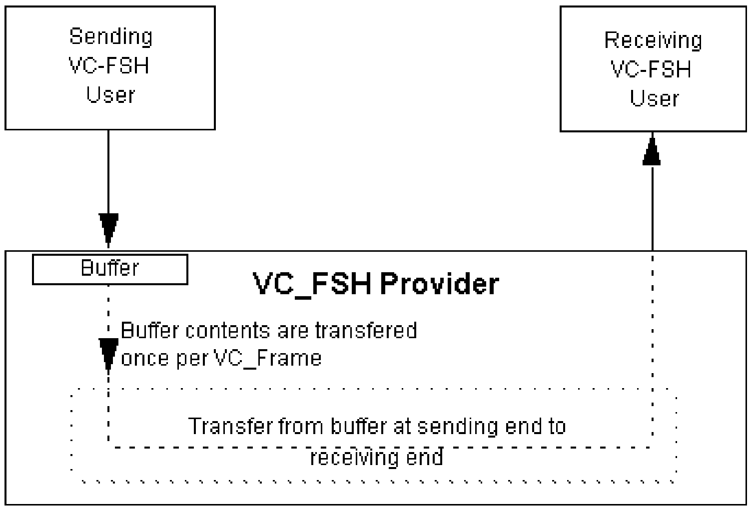

Figura 2.18. Esemplificazione del servizio di accodamento in un fornitore di servizi VC_FSH

   mandata solo una VC_FSH SDU per ogni VC_Frame. Questo servizio utilizza due primitive:

   - *VC_FSH.request*: per richiedere che una VC_FSH SDU venga posizionata in un buffer FSH;
   - *VC_FSH.indication*: per spedire una VC_FSH SDU.

4. *VC Operational Control Field Service (VC_OCF)*: esso fornisce il trasferimento sincronizzato di OCF di lunghezza fissa su ciascun frame di un VC: è di tipo unidirezionale. Questo servizio può essere modellizzato con un buffer, il cui contenuto viene trasferito dal VC_OCF SAP di bordo a quello di terra.

   I VC_FSH vengono trasmessi in modo sequenziale. In particolare gli OCF vengono trasferiti solo quando viene rilasciato un VC_Frame: alcuni OCF possono essere sovrascritti prima di essere spediti, mentre altri possono essere trasmessi più di una volta.

   Questo servizio utilizza due primitive:

   - *VC_OCF.request*: per richiedere che una VC_OCF SDU venga posizionata in un buffer OCF;
   - *VC_OCF.indication*: per spedire una VC_FSH SDU.

Occorre inoltre osservare che, qualunque sia il tipo di servizio utilizzato, le primitive di *request* vengono mandate dall'utente del servizio a specifici SAP e la loro ricezione fa si che lo Space Transfer Layer (Layer D) trasmetta le relative SDU. Le primitive di *indication* vengono passate da questo strato allo specifico utente e l'effetto della loro ricezione da parte dell'utente non viene definito. I quattro servizi analizzati richiedono il VC_Frame Service dallo strato inferiore (Layer C).

Viene ora analizzata la struttura dati, si veda [8], che viene passata dallo Space Transfer Layer (Layer D) al Virtual Channel Layer (Layer C): tale struttura viene denominata *Source Packet*. Essa può essere generata ad intervalli di tempo fissi o variabili e può essere di lunghezza fissa o variabile. Il Source Packet è sotto il completo controllo dei processi applicativi: ogni dato di sorgente possiede così un'organizzazione dei dati indipendente da quella degli altri.

Se un Source Packet contiene nel suo campo dati dei dati idle, viene chiamato *Idle Packet*: esso può essere generato quando è necessario mantenere la sincronizzazione dei dati da trasportare e dei processi di estrazione dei pacchetti. Una serie di Source Packet, generati consecutivamente da un singolo processo applicativo, viene denominato *Group of Source Packets*.

Un Source Packet incapsula una parte dei dati delle applicazioni che vengono trasmessi dai processi applicativi nello spazio a uno o più processi a terra. La sua struttura è identica a quella illustrata nel § 2.3.3, figura 2.4.

2.5.2 Layer C: Virtual Channel Access Layer

Esso, si veda [9], fornisce tre servizi allo strato superiore utilizzando il *Master Channel Transfer Frames*:

1. *VC_Frame Service (VCF)*: esso fornisce il trasferimento dei VC_Frame (VCF) da ognuno degli otto VC su di un Master Channel (MC) e può contenere o non contenere gli FSH o gli OCF. Questo servizio mantiene la sequenzialità, ma non garantisce la completezza. E' un servizio unidirezionale che può essere modellizzato come una coda che collega il VCF SAP a bordo con quello di terra, come mostrato in figura 2.19; esiste una coda separata per ogni VC.

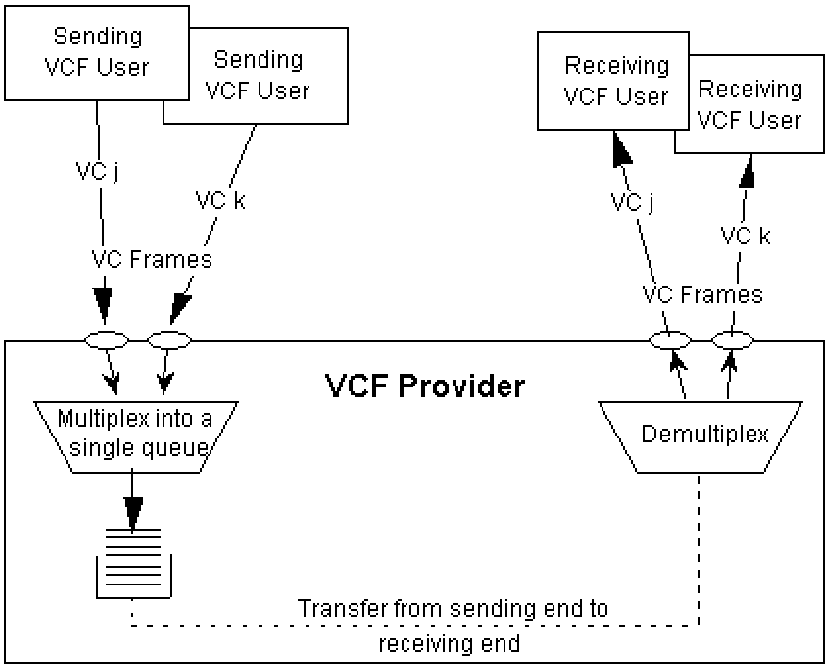

Figura 2.19. Esemplificazione del servizio di accodamento in un fornitore di servizi VCF

I VC_Frame vengono trasmessi in modo sequenziale. Questo servizio utilizza due primitive:

- *VCF.request*: per richiedere che una VCF SDU venga multiplata in uno specifico MC;
- *VCF.indication*: per spedire una VCF SDU.

La ricezione di una *VCF.request* provoca il trasferimento di una VCF SDU da parte del Virtual Channel Access Layer (Layer C), mentre la ricezione di una *VCF.indication* realizza le funzioni di sblocco, demultiplazione dei pacchetti dal campo dati del frame, rimozione dei pacchetti di idle, generazione di indicatori di qualità.

2. *MC Frame Secondary Header Service (MC_FSH)*: esso fornisce il trasferimento di FSH di lunghezza fissa in ogni frame su di un MC; il trasferimento è sincronizzato con il rilascio del MC_Frame. Questo servizio mantiene la sequenzialità, ma non garantisce la completezza. E' un servizio unidirezionale che può essere modellizzato come un buffer che collega il MC_FSH SAP a bordo con quello di terra. Il servizio MC_FSH (Layer C) e quello VC_FSH (Layer D) sono mutuamente esclusivi.

   Gli MC_FSH vengono trasmessi in modo sequenziale: in particolare viene spedito un MC_FSH per ogni MC_Frame. Questo servizio utilizza due primitive:

   - *MC_FSH.request*: per richiedere che una MC_FSH SDU venga posizionata in un buffer per essere trasmessa;
   - *VCF.indication*: per spedire una MC_FSH SDU.

   La ricezione di una *MC_FSH.request* provoca la sostituzione del contenuto del buffer da parte del Virtual Channel Access Layer (Layer C), mentre l'effetto della ricezione di una *MC_FSH.indication* da parte di un utente non viene definito.

3. *MC Operational Control Field Service (MC_OCF)*: esso fornisce il trasferimento di OCF di lunghezza fissa in ogni frame su di un MC; il

trasferimento è sincronizzato con il rilascio del MC_Frame. Questo servizio mantiene la sequenzialità, ma non garantisce la completezza. E' un servizio unidirezionale che può essere modellizzato con un buffer che collega il MC_OCF SAP a bordo con quello di terra. Il servizio MC_OCF (Layer C) e quello VC_OCF (Layer D) sono mutuamente esclusivi.

Gli MC_OCF vengono trasmessi in modo sequenziale. Questo servizio utilizza due primitive:

- *MC_OCF.request*: per richiedere che una MC_OCF SDU venga inserita in un Transfer Frame di uno specifico MC;
- *MC_OCF.indication*: per spedire una MC_OCF SDU.

Occorre inoltre osservare che, qualunque sia il tipo di servizio utilizzato, le primitive di *request* vengono passate dallo strato superiore a specifici SAP. Le primitive di *indication* vengono passate dal Virtual Channel Access Layer (Layer C) allo specifico utente. Tutti i servizi analizzati richiedono il Channel Access Service dallo strato inferiore (Layer B). Uno *"spacecraft"* può produrre più di un MC; questo strato possiede inoltre delle funzioni addizionali relative al MC:

- multiplare, al trasmettitore, due o più MC in un flusso di Transfer Frame a velocità costante;
- demultiplare, al ricevitore, due o più MC in un flusso di Transfer Frame a velocità costante.

Viene ora analizzata la struttura dati, si veda [8], che viene passata dal Virtual Channel Layer (Layer C) al Channel Access Layer (Layer B): tale struttura viene denominata *Transfer Frame*. Essa viene utilizzata per la trasmissione di Source Packet, dati di idle e di tipo privato. La lunghezza del Transfer Frame deve essere mantenuta costante per tutta la durata della missione e non deve comunque superare gli 8920 bit.

Tutti i Transfer Frame con lo stesso identificatore del pacchetto sullo stesso canale fisico costituiscono un *Master Channel*, che può avere da uno ad otto VC.

La struttura del Transfer Frame è illustrata in figura 2.20 ed è descritta di seguito.

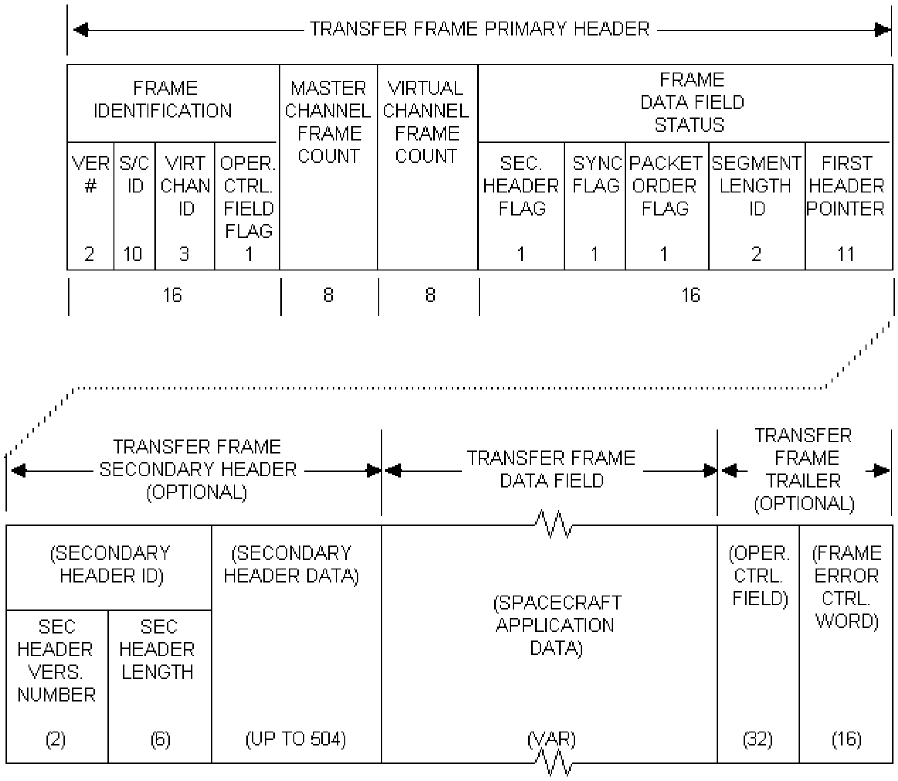

Figura 2.20. Formato del Transfer Frame della telemetria

1. *Frame Identification*: è costituita da due ottetti suddivisi nei seguenti sottocampi:

    (a) *Version Number*: al momento risulta definita soltanto la versione 1, indicata con '00'.

    (b) *Spacecraft Identifier*: contiene un codice di identificazione dello *"spacecraft"*.

    (c) *Virtual Channel Identifier*: fornisce l'identificatore del VC utilizzato.

    (d) *Operational Control Field Flag*: se settato a '1' indica la presenza dell' Operational Control Field.

2. *Master Channel Frame Count*: contiene un contatore sequenziale binario, modulo 256, il quale conta ogni Transfer Frame trasmesso su di un particolare Master Channel.

3. *Virtual Channel Frame Count*: contiene un contatore sequenziale binario, modulo 256, il quale conta ogni Transfer Frame trasmesso attraverso un particolare VC di un Master Channel.

4. *Frame Data Field Status*: è costituito da due ottetti suddivisi nei seguenti sottocampi:

   (a) *Secondary Header Flag*: viene settato a '1' se è presente la Secondary Header.

   (b) *Syncronization Flag*: segnala il tipo di dati che viene trasportato. Se posto a '0' indica che vengono trasportati Source Packet o dati idle (separati tra loro da un otteto di protezione), mentre viene posto a '1' se si trasportano dati privati (senza l'uso dell'ottetto di protezione).

   (c) *Packet Order Flag*: attualmente non è utilizzato.

   (d) *Segment Length Identifier*: viene posto ad '11' quando il Syncronization Flag vale '0', come richiesto dalle precedenti versioni del CCSDS.

   (e) *First Header Pointer*: contiene la rappresentazione binaria della locazione del primo ottetto dell'intestazione primaria relativa al primo pacchetto.

5. *Secondary Header Identification*: è costituito da un ottetto suddiviso in due sottocampi:

   (a) *Secondary Header Version Number*: attualmente è definita soltanto la versione '00'.

   (b) *Secondary Header Length*: è espressa in numero di ottetti meno uno.

6. *Secondary Header Data*: contiene i dati dell'intestazione secondaria.

7. *Spacecraft Application Data*: può contenere tre tipi di dati (Source Packet, dati idle, dati privati), ma i Source Packet non possono essere mescolati con i dati privati.

8. *Operational Control Field*: se presente, deve essere mandato in ogni Transfer Frame trasmesso attraverso uno specifico VC. Se il primo bit vale '0', questo campo contiene il *Command Link Control Word* (CLCW), denominato Type-1 Report; se il primo bit vale '1', contiene invece un Type-2 Report (non ancora definito).

9. *Frame Error Control Word*: è un campo obbligatorio se il Transfer Frame non viene codificato con Reed-Solomon. La procedura di codifica e decodifica utilizzata è identica a quella descritta nel § 2.3.5 punto 3.

2.5.3 Layer B: Channel Access Layer

Esso, si veda [9], fornisce il *Channel Access* (CA) Service, che realizza un trasferimento di una sequenza di Transfer Frame, di lunghezza fissa, a velocità costante. E' un servizio unidirezionale con un'opzionale rivelazione o correzione di errore. Viene modellizzato con una coda al CA SAP del trasmettitore, il cui contenuto viene trasferito, periodicamente ed in modo sincrono, al corrispondente CA SAP a terra, come mostrato in figura 2.21. A differenza degli altri servizi accodati visti negli strati superiori, questo richiede delle relazioni di tempo tra il CA dal lato dell'utente e quello dello strato. Il primo fornisce i Transfer Frame ad una velocità media che permetta al fornitore del CA di mantenere un *simbol rate* costante allo strato Physical Access sottostante. I Transfer Frame vengono trasferiti in modo sequenziale. Questo servizio utilizza due primitive:

- *CA.request*: per richiedere che una CA SDU venga trasferita; la sua ricezione causa il trasferimento di un CA SDU da parte del Channel Access Layer;

- *CA.indication*: viene passata dallo strato CA del ricevitore per trasmettere una CA SDU; la sua ricezione esegue alcune funzioni come la demultiplazione dei VC_Frame, il controllo del *Frame Error Control Field* e la decodifica.

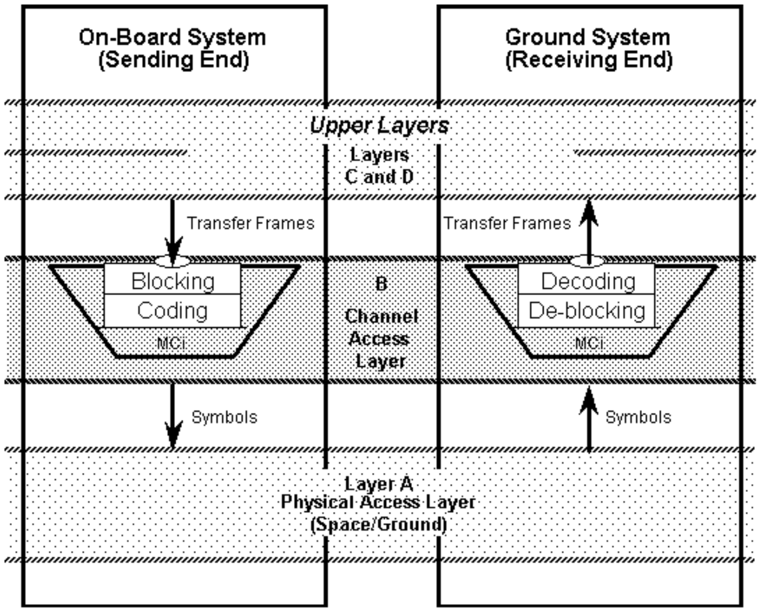

Figura 2.21. Struttura del Channel Access Service

2.5.4 Layer A: Physical Access Layer

Questo strato, si veda [7], si occupa di quella che viene chiamata *Telemetry Channel Coding*: essa è un metodo di elaborazione dei dati che vengono trasmessi dalla sorgente alla destinazione. In questo modo è possibile ricostruire i dati con bassa probabilità d'errore, migliorando così le prestazioni del canale.

Per un canale di telecomunicazioni che possiede una larghezza di banda fissata e non può tollerare l'incremento richiesto dai codici convoluzionali, il codice di Reed-Solomon risulta avere il vantaggio di una piccola espansione della larghezza di banda e la capacità di indicare la presenza di errori non correggibili.

Vengono di seguito illustrati i vari tipi di codifica che possono essere utilizzati.

1. *Codifica convoluzionale*

   Le caratteristiche di base di questo tipo di codifica sono:

   - Rate: 1/2

- Constraint-length: 7
- Vettori di connessione: G1=1111001; G2=1011011

Il decodificatore convoluzionale è del tipo a massima verosimiglianza (decodificatore di Viterbi). Se viene superata la capacità di correzione del decodificatore, possono verificarsi in uscita dei *burst* di errori. Lo schema a blocchi del codificatore è illustrato in figura 2.22.

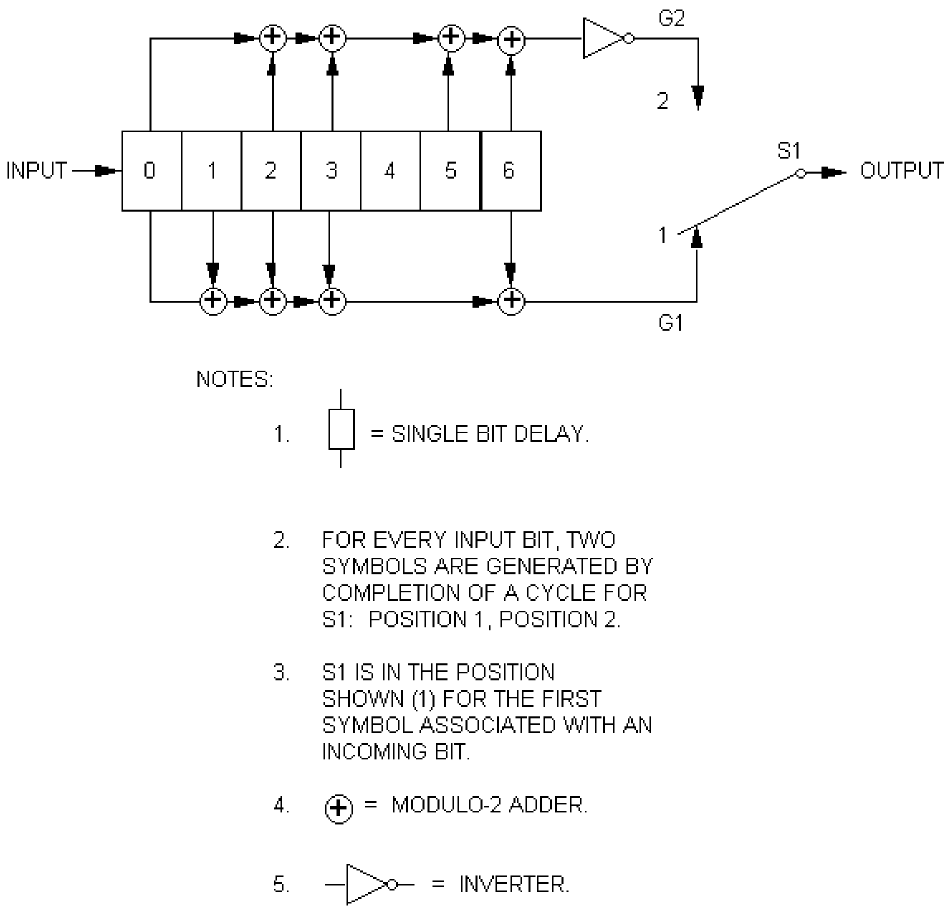

Figura 2.22. Schema a blocchi del codificatore convoluzionale

2. *Codifica con Reed-Solomon*

Il codice di Reed-Solomon è un potente correttore di errori a burst che fornisce un'eccellente capacità di correzione di tipo FEC (*Forward*

Error Correction) in un canale con rumore a burst. Il codice di Reed-Solomon utilizzato è il (255,223), capace di correggere fino a 16 simboli di Reed-Solomon errati in ogni parola di codice: per raggiungere questa affidabilità è necessaria una sincronizzazione dei codeblock. Il codice di Reed-Solomon, poichè non fornisce un sufficiente guadagno di codice, viene spesso concatenato con un codice convoluzionale: il codice Reed-Solomon viene utilizzato esternamente (*outer code*), mentre quello convoluzionale viene utilizzato internamente (*inner code*).

Per il codice Reed-Solomon occorre definire:

- J: bit per simbolo di Reed-Solomon (per questo codice J=8);
- E: capacità di correzione di un simbolo di Reed-Solomon errato in una parola di codice (per questo codice E=16);
- $n = 2^J - 1$: numero di simboli in una parola di codice (per questo codice n=255);
- $2E$: numero di simboli di Reed-Solomon che rappresentano il controllo di parità;
- $k = n-2E$: numero di simboli di Reed-Solomon che rappresentano informazione.

Questo codice è di tipo sistematico, cioè la sequenza d'ingresso appare come la prima parte della parola di codice in uscita.

La codifica con Reed-Solomon permette di utilizzare un interlacciatore di simboli: i valori della profondità dell'interlacciatore, indicati con i, possono variare da 1 a 5. Tale profondità viene normalmente fissata in ogni missione; lo schema dell'interlacciatore Reed-Solomon è illustrato in figura 2.23. I dati entrano nella porta chiamata IN; gli interruttori S1 e S2 sono sincronizzati insieme e commutano tra i vari codificatori seguendo la sequenza in modo crescente (dal primo all'i-esimo). Un codeblock sarà formato da $(223{\cdot}i)$ simboli di Reed-Solomon che entrano in ingresso. L'azione sincronizzata di S2 riassembla i simboli alla porta di uscita, chiamata OUT, nello stesso modo dell'ingresso. Procedendo in questo modo, ogni codificatore manda in uscita $255 - 223 = 32$

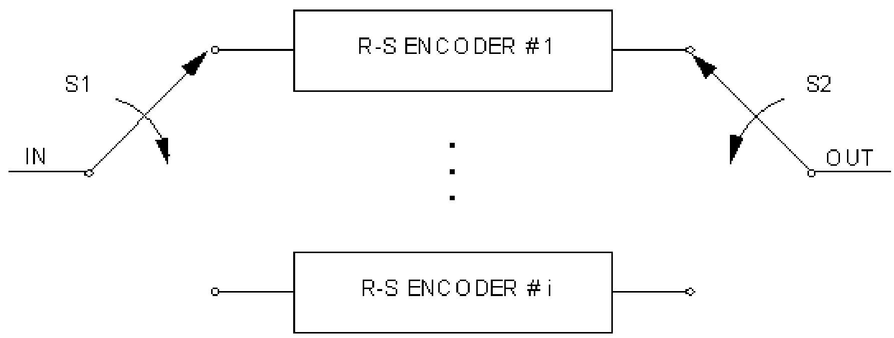

Figura 2.23. Schema a blocchi dell'interlacciatore Reed-Solomon

simboli di controllo d'errore. Mediante il metodo dell'interlacciatore, i $(k \cdot i)$ simboli di informazione, originalmente consecutivi, che entrano nel codificatore, non risultano scambiati alla sua uscita, con l'aggiunta, inoltre, di $(2 \cdot E \cdot i)$ simboli di controllo d'errore. La massima lunghezza del codeblock risulta perciò di $L_{max} = n \cdot i = (s^J - 1) \cdot i = 255 \cdot i$ simboli.

Il codice di Reed-Solomon può essere utilizzato anche in forma accorciata: qualora infatti i primi Q bit siano pari a zero, è possibile evitare di trasmetterli costituendo il cosidetto "riempimento virtuale". In questo modo è possibile trasmettere un numero maggiore di codeblock per unità di tempo. Il decodificatore deve operare comunque sull'intero blocco, aggiungendo quindi i bit di zero che non sono stati trasmessi. Quando il decodificatore riceve i $(k \cdot i - Q)$ simboli di informazione, il calcolo dei $(2 \cdot E \cdot i)$ simboli di controllo di errore viene comunque fatto su $(k \cdot i)$ simboli, aggiungendo gli zeri mancanti. La partizione e il riempimento virtuale dei codeblock, realizzati con il codice Reed-Solomon accorciato, sono illustrati in figura 2.24.

3. *Codifica con "Turbo codice"*

   I turbo codici sono codici binari con codeblock molto grandi (circa 1000 bit). Essi sono di tipo sistematico e non trasparente, cioè viene sconsigliata la codifica differenziale dopo il turbo codificatore: l'ambiguità di fase deve quindi essere rilevata e risolta mediante la sincronizzazione dei frame, descritta più avanti.

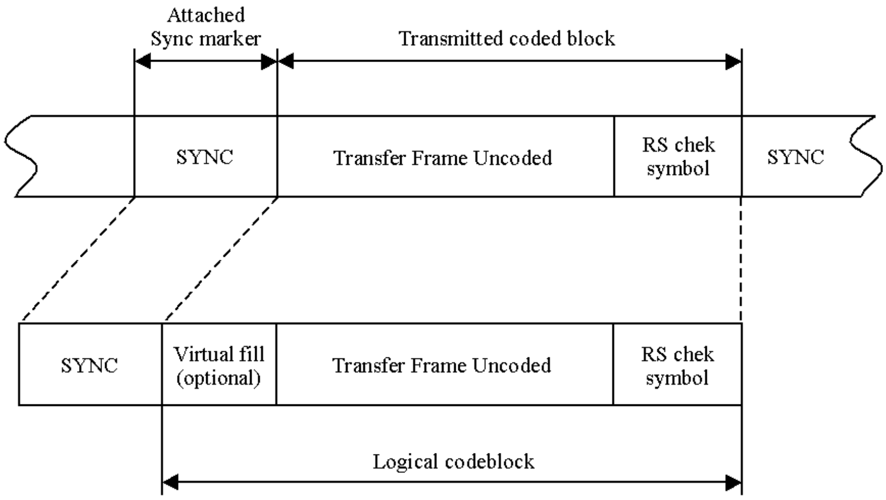

Figura 2.24. Partizione e riempimento virtuale del codice di Reed-Solomon accorciato

Il turbo codice può essere utilizzato per ottenere un elevato guadagno di codifica: esso è costituito da una combinazione di due semplici codificatori concatenati, paralleli e sistematici, ciascuno utilizzante un codice convoluzionale recursivo. Il numero di stati di ognuno dei due semplici codificatori è pari a 16, mentre il rate di codifica nominale, senza la terminazione del traliccio, può essere scelto tra:

$$r = \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{6}$$

La lunghezza della parte di informazione relativa ai blocchi è pari a $k = (223 \cdot 8 \cdot i)$ e sono scelte per avere la compatibiltà con la corrispondente profondità dell' interlacciatore di Reed-Solomon; la lunghezza del codeblock risulta pari a $n = (k + 4)/r$ bit. L'interlacciatore per turbo codici esegue una permutazione bit per bit dell'intero blocco dati, seguendo un certo algoritmo.

Lo schema a blocchi del turbo codificatore, illustrato in figura 2.25, prevede l'uso di un vettore di connessione verso l'ingresso (G0=10011) e di tre verso l'uscita (G1=11011, G2=10101, G3=11111). Ogni frame, costituito da k bit di informazione, giunge dapprima all'interlacciatore

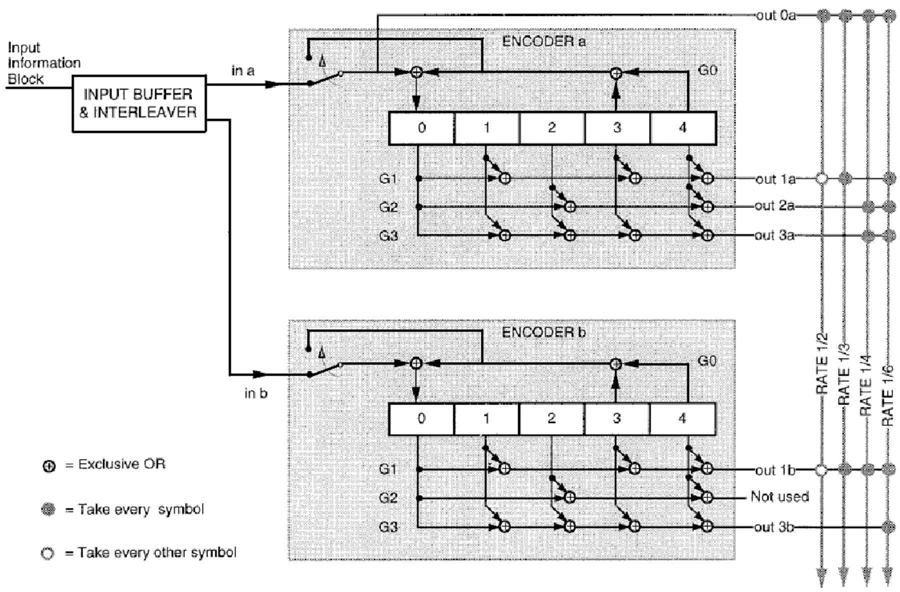

Figura 2.25. Schema a blocchi del turbo codificatore

dove viene mantenuto in un apposito buffer; l'algoritmo di permutazione consente di ottenere in uscita due buffer: nel primo è presente il frame nell'ordine in cui è arrivato e nel secondo lo stesso risulta permutato. I due buffer di uscita sono poi connessi ai codificatori "a" e "b". Il primo opera sui bit non permutati, mentre il secondo riceve gli stessi bit permutati dall'interlacciatore: entrambi sono inizializzati a zero. Inizialmente essi ricevono in ingresso $(k + 4)$ bit, producendo un codeblock di uscita di $(k + 4)/r$ bit. In particolare, per i primi k bit, gli interruttori sono in posizione bassa, mentre per gli ultimi 4 bit gli interruttori vengono spostati verso l'alto. I simboli codificati vengono multiplati dall'alto verso il basso attraverso la linea di uscita selezionata per il rate desiderato. Ad esempio per i diversi tipi di rate si ha:

- r=$\frac{1}{2}$ $\Rightarrow$ 0a, 1a, 0a, 1b, 0a, 1a, ...
- r=$\frac{1}{3}$ $\Rightarrow$ 0a, 1a, 1b, 0a, 1a, 1b, ...
- r=$\frac{1}{4}$ $\Rightarrow$ 0a, 2a, 3a, 1b, 0a, 2a, ...

- r=$\frac{1}{6}$ $\Rightarrow$ 0a, 1a, 2a, 3a, 1b, 3b, ...

Vengono di seguito trattati altri due aspetti importanti presenti in questo strato:

- *Frame Synchronization*: la sincronizzazione tra i frame o i codeblock è necessaria per la decodifica mediante Reed-Solomon e Turbo Codice. Essa viene raggiunta utilizzando flussi di codeblock di lunghezza fissata, inserendo tra essi un ASM (*Attached Sync Marker*): la sincronizzazione viene acquisita riconoscendo la specifica sequenza di bit.

  Dal lato del codificatore, se il canale della telemetria non è codificato, o è codificato solo con Reed-Solomon o con Turbo Codice, l'ASM viene aggiunto direttamente all'uscita del codificatore; se viene utilizzato un codificatore interno insieme al codificatore di Reed-Solomon esterno, l'ASM viene codificato da quello interno, ma non da quello esterno.

  Dal lato del decodificatore, se viene utilizzato un Reed-Solomon insieme al convoluzionale, l'ASM può essere acquisito sui simboli di canale o sui bit già decodificati dal codificatore interno; se invece viene utilizzato un sistema con Turbo Codice, l'ASM deve essere acquisito soltanto nei simboli di canale.

  L'ASM, per dati che non vengono turbo-codificati, è costituito da 32 bit, mentre per un Turbo Codice la lunghezza è pari a $32/r$ bit.

- *Pseudo Randomizer*: per mantenere la sincronizzazione dei bit è richiesto che il segnale abbia una minima densità di transizione dei bit. Per assicurare questo, ogni bit del Transfer Frame o del codeblock viene mandato in una EX-OR con una sequenza pseudocasuale standard. Il casualizzatore viene posizionato dopo la turbo codifica o la codifica di Reed-Solomon, ma prima del codificatore convoluzionale. L'ASM è già configurato in modo ottimale per la sincronizzazione. Al ricevitore il codeblock originale viene ricostruito utilizzando la stessa sequenza pseudo-casuale; il polinomio generatore utilizzato vale:

$$h(x) = x^8 + x^7 + x^5 + x^3 + 1 \tag{2.13}$$

che si ripete dopo 255 bit. Lo schema è illustrato in figura 2.26, dove i registri devono essere tutti inizializzati ad '1' durante ogni ASM.

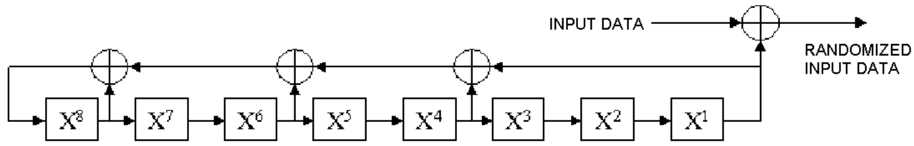

Figura 2.26. Schema a blocchi del casualizzatore per la telemetria

2.6 Riepilogo dello standard CCSDS

Per ottenere una migliore comprensione del funzionamento del complesso schema CCSDS, in questo paragrafo verrà brevemente riassunta l'intera struttura del telecomando e della telemetria. Per ottenere una conoscenza più approfondita dei singoli strati e delle singole strutture dati, verranno inseriti dei riferimenti ai paragrafi precedentemente illustrati.

2.6.1 Telecomando secondo il CCSDS

Il CCSDS prevede per il telecomando una suddivisione in sette strati, che possono essere raggruppati in tre categorie.

La prima, illustrata in figura 2.27, riguarda i processi applicativi, cioè gli strati di più alto livello (essi non vengono descritti in dettaglio dal CCSDS). Lo strato più alto è l'*Application Process Layer* (§ 2.3.1 e § 2.4.7) che fornisce l'interfaccia uomo-macchina. Esso genera le direttive di comando (*Command Directive*) da passare allo strato sottostante. Quest'ultimo viene chiamato *System Management Layer* (§ 2.3.2 e § 2.4.6) e svolge il compito di tradurre le direttive di comando, espresse mediante linguaggi di alto livello,

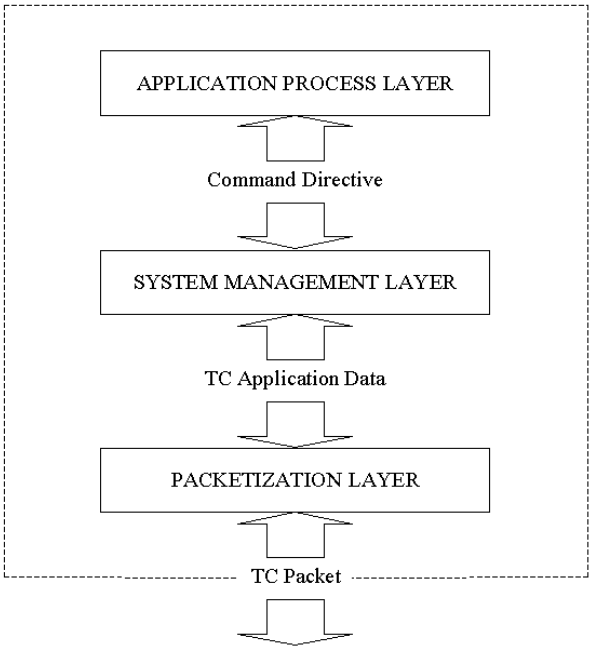

Figura 2.27. Strati del telecomando relativi ai processi applicativi

in un linguaggio di controllo (*TC Application Data*) utilizzato dallo strato inferiore. L'ultimo strato dei processi applicativi viene detto *Packetization Layer* (§ 2.3.3 e § 2.4.5) e si occupa di creare una struttura dati, chiamata TC Packet, costituita da un'intestazione di 6 ottetti e un campo dati avente lunghezza massima pari a 65536 ottetti; tale struttura viene poi passata al blocco che si occupa dell'aspetto retistico del sistema. In questo strato, inoltre, vengono inserite le varie misure di sicurezza necessarie per ottenere una protezione dei dati trasmessi.

La seconda categoria degli strati del telecomando, illustrata in figura 2.28, si occupa della sezione retistica del sistema, cioè tratta gli aspetti relativi ai protocolli di trasmissione con particolare riferimento alla tecnica ARQ. Il primo di questi strati, denominato *Segmentation Layer* (§ 2.3.4 e § 2.4.4), riceve il TC Packet precedentemente generato e lo scompone in una struttura dati più piccola detta TC Segment. Questo strato, inoltre, è in grado di "muliplexare" fino ad un massimo di 64 flussi dati differenti, cioè di TC Packet

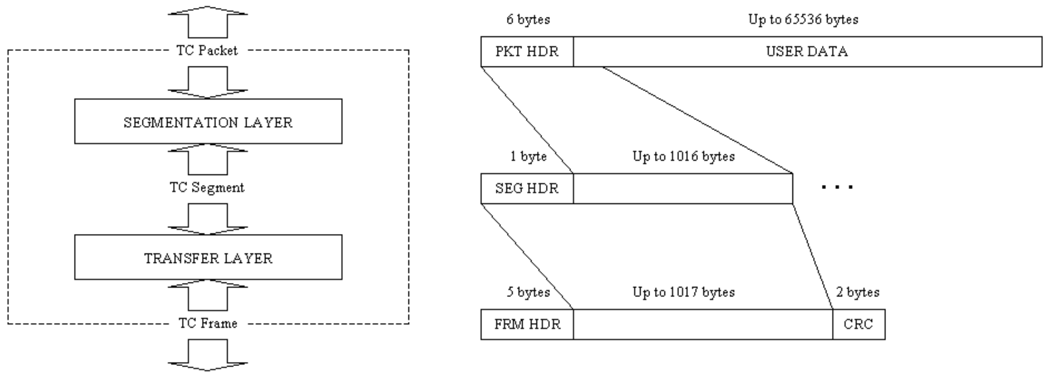

Figura 2.28. Strati e strutture dati della sezione retistica relativa al telecomando

che possono provenire da diversi processi applicativi. Il TC Segment prodotto viene poi passato allo strato inferiore, chiamato *Transfer Layer* (§ 2.3.5 e § 2.4.3), che si occupa di gestire l'intero protocollo ARQ, il quale adotta una tecnica di tipo Go-Back-N. Questo strato, inoltre, incapsula il TC Segment in una nuova struttura dati, detta TC Frame, tramite l'aggiunta di un apposito campo di intestazione e di uno opzionale relativo al controllo d'errore (CRC). Tale campo permette di etichettare ulteriormente i dati che quindi possono essere suddivisi in 64 canali virtuali con differente priorità di consegna. Infine il TC Frame prodotto viene passato allo strato sottostante.

L'ultima categoria di strati del telecomando, mostrata in figura 2.29, si occupa della parte trasmissiva, cioè delle operazioni di codifica e di trasmissione dei segnali sul canale. Il primo di questi strati, denominato *Coding Layer* (§ 2.3.6 e § 2.4.2), riceve il TC Frame dallo strato superiore e, prendendo k bit alla volta, applica su di essi un codice BCH (63,56) modificato, che produce in uscita n bit: i valori consigliati dal CCSDS sono pari a 64 per n e 56 per k. I blocchi codificati prodotti vengono incapsulati in una struttura, detta CLTU, tramite l'aggiunta di una sequenza di "start", che identifica l'inizio della struttura dati, e di una sequenza di "tail", che identifica la fine del CLTU. Quest'ultimo viene poi passato allo strato sottostante, denominato *Physical Layer* (§ 2.3.7 e § 2.4.1), che modula i bit e trasmette sul canale i segnali così prodotti. Tale strato applica, inoltre, particolari procedure che

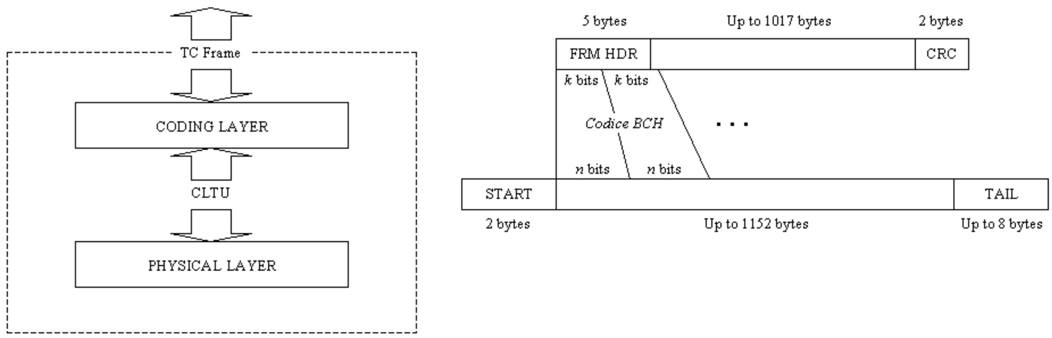

Figura 2.29. Strati e strutture dati della sezione trasmissiva relativa al telecomando

prevedono la trasmissione di una *sequenza di acquisizione* in grado di fornire il sincronismo di simbolo iniziale.

La sequenza delle operazioni descritta è stata effettuata considerando il flusso dati al trasmettitore di terra relativo al telecomando; le stesse procedure verranno applicate al ricevitore di bordo, ma in ordine inverso. Per il flusso dati del telecomando occorre osservare che la presenza della tecnica ARQ necessita di un controllo ad anello chiuso mediante delle conferme; queste ultime verranno inglobate nel flusso dati della telemetria, come descritto nel paragrafo seguente.

2.6.2 Telemetria secondo il CCSDS

Il CCSDS prevede per la telemetria una suddivisione in quattro strati, secondo quanto illustrato in figura 2.30. Lo strato più alto, detto *Space Transfer Layer* (§ 2.5.1) fornisce l'interfaccia uomo-macchina. Esso raccoglie i dati forniti dai sensori presenti a bordo e li incapsula in una struttura dati denominata Source Packet, tramite l'aggiunta di un campo di intestazione; occorre osservare che la struttura di questo pacchetto è identica a quella del TC Packet utilizzato nel telecomando. Il Source Packet viene poi passato allo strato sottostante, detto *Virtual Channel Access Layer* (§ 2.5.2). Esso inserisce parte del Source Packet nel campo dati di una struttura, chiamata Transfer Frame, aggiungendo, in testa, un'intestazione primaria più una

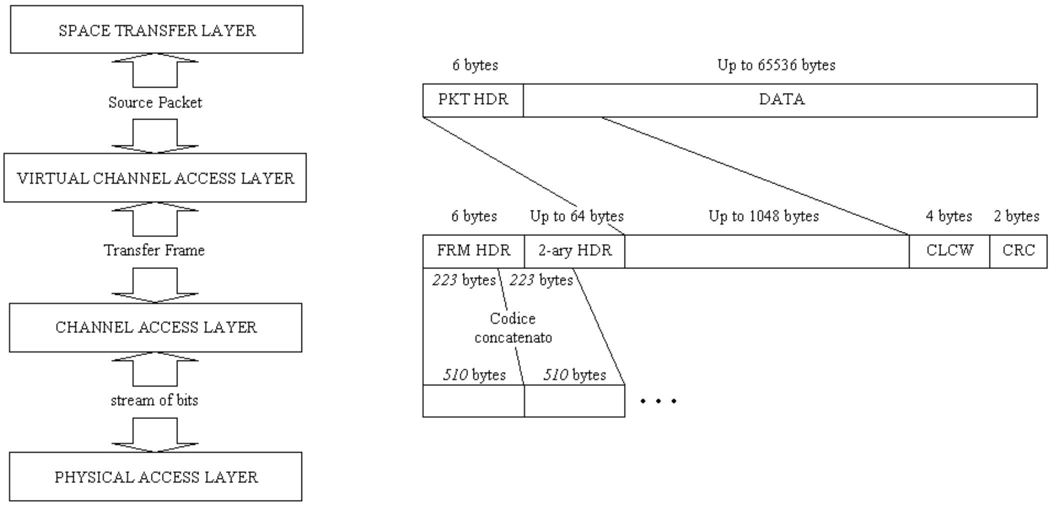

Figura 2.30. Strati e strutture dati della telemetria

secondaria opzionale e, in coda, la conferma (CLCW) fornita dal protocollo ARQ per i dati del telecomando più un campo opzionale per il controllo d'errore. Questo strato, inoltre, è in grado di "muliplexare" il flusso dati utilizzando un numero massimo di canali virtuali differenti pari a nove. Il Transfer Frame realizzato viene poi passato allo strato inferiore, denominato *Channel Access Layer* (§ 2.5.3), che si occupa della codifica. Esso utilizza un codificatore concatenato (convoluzionale e Reed-Solomon) che richiede 223 byte in ingresso e fornisce 510 byte in uscita. Il flusso di bit in uscita da questo strato, infine, viene inviato all'ultimo strato, detto *Physical Access Layer* (§ 2.5.4), che modula i bit e trasmette sul canale i segnali così prodotti.

La sequenza delle operazioni descritta è stata effettuata considerando il flusso dati al trasmettitore della telemetria (a bordo); le stesse procedure verranno applicate al ricevitore (a terra), ma in ordine inverso.

2.6.3 Applicabilità dello schema CCSDS ad Heliplat

Viene ora effettuata un'analisi sulla possibilità di applicare la struttura descritta dal CCSDS alla piattaforma stratosferica Heliplat. La trattazione svolta in questa tesi prende in considerazione soltanto i livelli più bassi, in particolar modo quelli legati alla parte trasmissiva. Sia per la tratta in up-link che per quella in down-link, lo strato fisico proposto dal CCSDS non specifica alcun tipo di modulazione: uno degli obiettivi di questa tesi sarà quindi quello di effettuare uno studio per la scelta dello schema di modulazione ottimale.

Per quanto riguarda lo strato di codifica del telecomando, il CCSDS propone di utilizzare un codice BCH (63,56) al quale applicare successivamente una tecnica ARQ di tipo Go-Back-N. Nello studio che verrà effettuato si manterrà lo stesso schema di codifica valutando, però, l'opportunità o meno di utilizzare lo stesso codice BCH. Facendo riferimento alla figura 2.29, si è scelto di utilizzare il campo CRC del TC Frame (previsto opzionale dal CCSDS) in modo da garantire delle probabilità d'errore adeguatamente basse; come conseguenza di questa scelta, le sequenze di "start" e "tail" risultano superflue e, per non appesantire ulteriormente la struttura, sono state eliminate. Per quanto riguarda la tecnica ARQ si è effettuato uno studio senza scendere nei dettagli implementativi di tipo retistico, quali ad esempio le temporizzazioni e il dimensionamento delle finestre di trasmissione e ricezione. Gli strati trasmissivi utilizzati, con le relative strutture dati adattate al sistema Heliplat, vengono riportati in figura 2.31; le dimensioni del campo dati del TC Frame verranno analizzate nel capitolo 8, dove si terrà conto dei ritardi da esso introdotti.

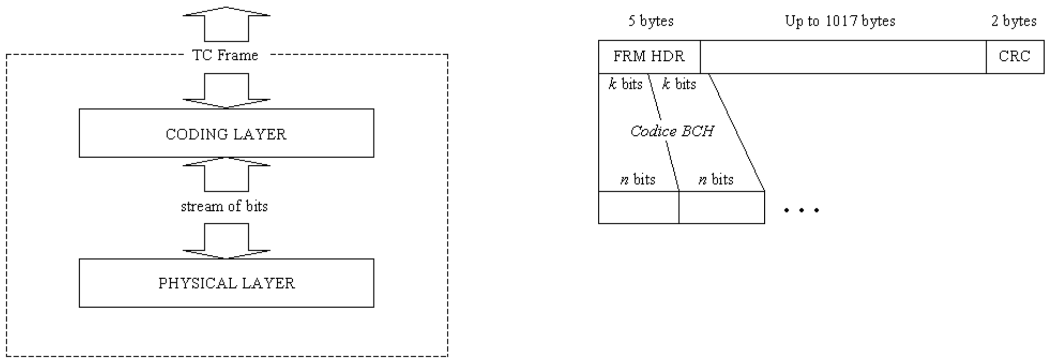

Figura 2.31. Strati e strutture dati della sezione trasmissiva relativa al telecomando, applicate al caso del sistema Heliplat

Anche per lo strato di codifica della telemetria si è scelto di seguire lo stesso schema proposto dal CCSDS, cioè l'utilizzo di un codificatore concatenato costituito da un codice convoluzionale interno (2,1,7), un interlacciatore e un codice di Reed-Solomon esterno (255,223). Come per il caso del telecomando verrà effettuato uno studio per valutare l'opportunità o meno di utilizzare gli stessi parametri di tali codici. Facendo riferimento al Transfer Frame della figura 2.30, si è scelto di eliminare i campi Secondary Header e CRC previsti opzionali dal CCSDS; questa scelta è dettata dal fatto che il primo non essendo obbligatorio tenderebbe soltanto ad appesantire la struttura e ad aumentare i ritardi del sistema, mentre il secondo non è necessario qualora si utilizzi il codice di Reed-Solomon. Uno schema riassuntivo degli strati utilizzati e delle strutture dati adattate al sistema Heliplat, viene riportato in figura 2.32; anche in questo caso le dimensioni del campo dati del Transfer Frame verranno analizzate nel capitolo 8.

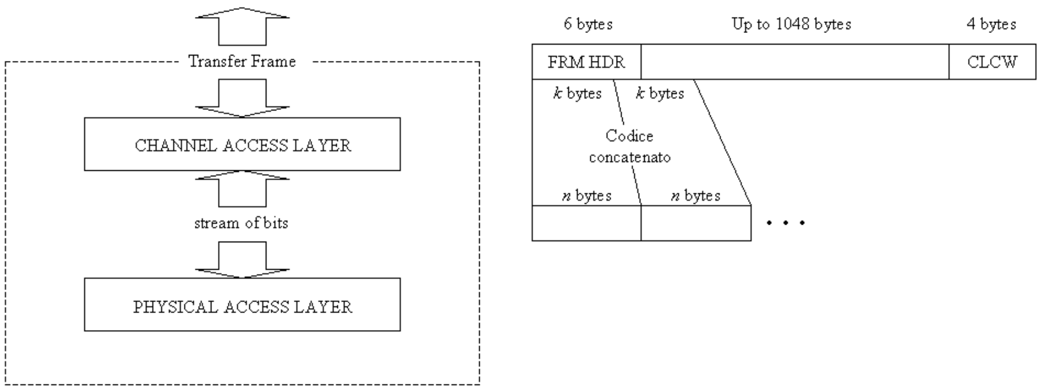

Figura 2.32. Strati e strutture dati della telemetria, applicate al caso del sistema Heliplat

Lo studio dell'applicabilità degli altri strati, invece, non viene preso in considerazione in quanto esula dagli obiettivi di questa tesi.

Bibliografia

[1] *Telecommand: Summary of Concept and Rationale*, Report CCSDS 200.0-G-6, Green Book, Issue 6, Washington D.C., January, 1987, http://www.ccsds.org/all_books.html#telecommand

[2] *Telecommand, Part 1: Channel Service*, Recommendation CCSDS 201.0-B-3, Blue Book, Issue 3, Washington D.C., June, 2000, http://www.ccsds.org/all_books.html#telecommand

[3] *Telecommand, Part 2: Data Routing Service*, Recommendation CCSDS 202.0-B-2, Blue Book, Issue 2, Washington D.C., November, 1998, http://www.ccsds.org/all_books.html#telecommand

[4] *Telecommand, Part 2.1: Command Operation Procedures*, Recommendation CCSDS 202.1-B-1, Blue Book, Issue 1, Washington D.C., October, 1991, http://www.ccsds.org/all_books.html#telecommand

[5] *Telecommand, Part 3: Data Management Service*, Recommendation CCSDS 203.0-B-1, Blue Book, Issue 1, Washington D.C., January, 1987, http://www.ccsds.org/all_books.html#telecommand

[6] *Telemetry: Summary of Concept and Rationale*, Report CCSDS 100.0-G-1, Green Book, Issue 1, Washington D.C., December, 1987, http://www.ccsds.org/all_books.html#telemetry

[7] *Telemetry Channel Coding*, Recommendation CCSDS 101.0-B-3, Blue Book, Issue 3, Washington D.C., May, 1992,

http://www.ccsds.org/all_books.html#telemetry

[8] *Packet Telemetry*, Recommendation CCSDS 102.0-B-2, Blue Book, Issue 2, Washington D.C., January, 1987, http://www.ccsds.org/all_books.html#telemetry

[9] *Packet Telemetry Services*, Recommendation CCSDS 103.0-B-1, Blue Book, Issue 1, Washington D.C., May, 1996, http://www.ccsds.org/all_books.html#telemetry

[10] A. Lausetti, F. Filippi, *Elementi di meccanica del volo*, Levrotto & Bella, Torino, Italy, 1956

[11] R. Foschini, *Nozioni di aerodinamica e meccanica del volo*, Esculapio, Bologna, Italy, 1980

[12] A. Leich, *GPS Satellite Surveying*, John Wiley & Sons, New York, USA, 1990

[13] F. Crosilla, L. Mussio, *Il sistema di posizionamento globale satellitare GPS*, CISM, Udine, Italy, 1991

[14] T. Logsdon, *Understanding the Navstar: GPS, GIS, and IVHS*, 2nd ed., Van Nostrand Reinhold, New York, USA, 1995

[15] S. Focardi, I. Massa, U. Uguzzoni *Fisica generale: meccanica e termodinamica*, Ambrosiana, Milano, Italy, 1999

[16] R. Noyes, *Fuel cells for public utility and industrial power*, Noyes Data Corporation, Park Ridge, New Jersey, USA, 1977

[17] W. Mitchell jr., *Fuel Cells*, Accademic Press, New York, USA, 1963

[18] C. Hu, R. M. White *Solar Cells from basic to advanced systems*, McGraw Hill, New York, USA, 1983

[19] M. Sittig, *Solar cells for photovoltaic generation of elettricity: materials, devices and application*, Noyes Data Corporation, Park Ridge, New Jersey, USA, 1979

[20] T. S. Rappaport, *Wireless communications: principles and practice*, Prentice Hall International Editions, New Jersey, USA, 1996

[21] W. C. Jakes jr., *Microwave mobile communications*, John Wiley & Sons, North Andover, Massachusetts, USA, 1974

[22] L. J. Cimini jr., *Analysis and simulation of digital mobile channel using orthogonal frequency division multiplexing*, IEEE Transactions on Communications, Vol. 33, No. 7, July, 1985

[23] M. Alard, R. Lassalle, *Principles of modulation and channel coding for digital broadcasting for mobile receivers*, EBU Review, Tech. No. 224, August, 1987

[24] L. Lo Presti, F. Neri, *L'analisi dei segnali*, 2nd ed., C.L.U.T., Torino, Italy, 1992

[25] S. Benedetto, E. Biglieri, *Trasmission with wireless applications*, Plenum, 1998

[26] A. Baciotti, F. Ricci, *Analisi matematica volume I*, Liguori editore, Napoli, Italia, Ottobre, 1994

[27] J. G. Proakis, M. Salehi, *Communication systems engineering*, Prentice Hall International Editions, New Jersey, USA, 1994

[28] J. Saltz, S. B. Weinstein, *Fourier transform communication system*, presented at the Ass. Comput. Machinary Conference Computers and Communication, Pine Mountains, October, 1969

[29] S. Benedetto, E. Biglieri, V. Castellani, *Digital transmission theory*, Prentice Hall International Editions, Englewood Cliffs, New Jersey, USA, 1987

[30] F. Fagnola, G. Pistone, *Primo semestre di probabilità* , C.L.U.T., Torino, Italy, 1996

[31] P. A. Laurent, *Exact and approximate construction of digital phase modulations by superposition of amplitude modulated pulse (AMP)*, IEEE Transactions on Communications, Vol. COM-34, No. 2, February, 1986

[32] K. Kuchi, V. K. Prabhu, *Power spectral density of GMSK modulation using matrix methods*, IEEE Military Communications Conference Proceedings, Vol. 1, 1999

[33] K. Kuchi, V. K. Prabhu, *Spectral occupancy and error rate considerations of coherent GMSK*, IEEE Wireless Communications and Networking Conference, Vol. 3, 1999

[34] A. Linz, A. Hendrickson, *Efficient implementation of an I-Q GMSK modulator*, IEEE Transactions on Circuit and Systems II: Analog and Digital Signal Processing, Vol. 43, Issue 1, January, 1999

[35] K. Murota, K. Hirade, *GMSK modulation for digital mobile radio telephony*, IEEE Transactions on Communications, Vol. COM-29, No. 7, July, 1981

[36] *TOPSIM IV: user manual*, Politecnico di Torino, Torino, Italy, 1993

[37] L. Lo Presti, M. Mondin, *Design of optimal FIR raised cosine filter*,

Electronics Letters, Vol. 25, No. 7, March, 1989

[38] G. David Forney jr., *Concatenated codes*, M.I.T. Press, Cambridge, Massachusetts, USA, 1966

[39] W. W. Peterson, *Error correcting codes*, M.I.T. Press and John Wiley & Sons, New York, USA, 1961

[40] S. B. Wicker, V. K. Bhargava, *Reed-Solomon codes and their application*, IEEE Press, New York, USA, 1994

[41] R. E. Blahut, *Theory and practice of error control codes*, Addison-Wesley Publishing Company, Reading, Massachusetts, USA, 1983

[42] G. C. Clark jr., J. B. Cain, *Error-correction coding for digital communications*, Plenum, New York, USA, 1981

[43] A. Dholakia, *Introduction to convolutional codes with applications*, Kluwer Accademic Publishers, Norwell, Massachusetts, USA, 1994

[44] J. D. Spragins, J. L. Hammond, K. Pawlikowski *Telecomunications protocols and design*, Addison-Wesley Publishing Company, Reading, Massachusetts, USA, 1991

[45] K. Sigmon, *Matlab: primer*, 4th ed., CRC Press, Boca Raton, Florida, USA, 1994

[46] C. A. Balanis, *Antenna theory: analysis and design*, 2nd ed., John Wiley & Sons, New York, USA, 1997

[47] J. D. Kraus, *Antennas*, McGraw Hill, New York, USA, 1950

[48] W. L. Stutzman, G. A. Thiele *Antenna theory and design*, John Wiley & Sons, New York, USA, 1981

[49] G. Vecchi, *Appunti sull'irradiazione e le antenne*, C.L.U.T., Torino, Italy, 1998

[50] G. Monti Guarnieri, *Nozioni sui ponti radio*, IV ed., Tamburini editore, Milano, Italia, 1971

[51] L. J. Ippolito jr., *Radiowave propagation in satellite communication*, Van Nostrand Reinhold Company, New York, USA, 1986