Reti di telecomunicazioni

sav

2025

Contents

Chapter 1

Livello 2 Protocolli di linea

1.1 Livello 2 Frame DHLC

1.1.1 Struttura del frame

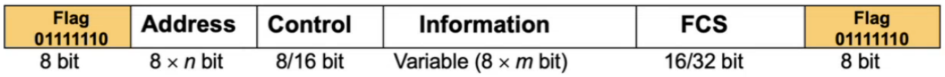


Figure 1.1: Struttura del frame HDLC

- flag: sono posti ad inizio e fine frame così da delimitare il frame e far capire a chi lo riceve quando inizia e quando finisce il frame, questi flag sono SEMPRE 8 bit in questa sequenza: 01111110, ossia 7E in esadecimale
- address: indirizzo del trasmettitore, ricevitore o broadcast(se tutti 1)
- control: indica il tipo di frame
- information: dati del livello superiore, payload
- FCS(frame check sequence):

1.1.2 Bit stuffing

A proposito dei flag usati nel frame(011111110) che succede se questa sequenza invece fa parte della informazione utile del frame e non come flag delimitatore?

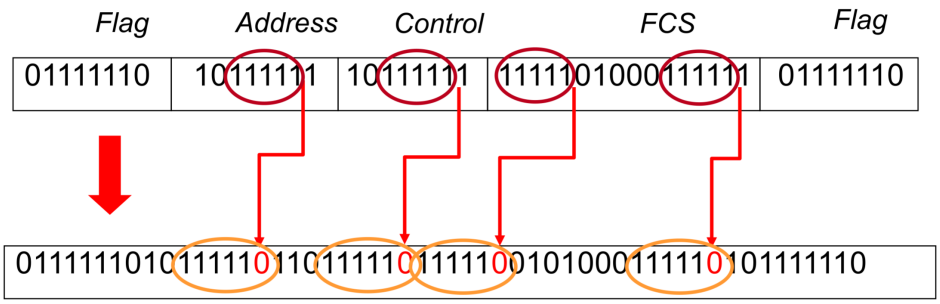


Figure 1.2: Esempio di bit stuffing

Risolvo inserendo uno 0 ogni cinque 1 ripetuti; questo fa perdere la molteplicità di 8 del frame.

1.1.3 Rivelazione dell'errore con CRC

HDLC numera i frame e usa il protocollo di linea, utilizzando 3 o 7 bit, usando una tecnica di piggybacking per gestire il flusso che viene dalla direzione opposta(finestre di trasmissione e ricezione).

Il CRC è un metodo per il calcolo di somme di controllo (checksum). Il nome deriva dal fatto che i dati d'uscita sono ottenuti elaborando i dati di ingresso i quali vengono fatti scorrere ciclicamente in una rete logica.

Sfrutta l'algebra dei campi finiti ed è utile per l'individuazione di errori casuali nella trasmissione dati (a causa di interferenze, rumore di linea, distorsione), il CRC non è invece affidabile per verificare la completa correttezza dei dati contro tentativi intenzionali di manomissione.

Polinomio generatore Un codice CRC è definito dal suo polinomio generatore di ordine r:
esempio r = 3

$$G(x) = x^3 + x^2 + 1 \quad \Rightarrow \quad 1101 \tag{1.1}$$

Cosa avviene in trasmissione

Con una scelta opportuna del polinomio generatore è possibile rilevare errori sul singolo bit etc... ;
setup del CRC per la trasmissione:

Traduco i bit che compongono header e payload in un polinomio P(x) di coefficienti d; perciò di grado d-1.

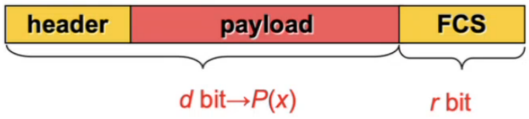


Figure 1.3: Schema del calcolo di CRC, P(x)

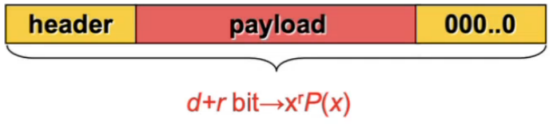


Figure 1.4: Schema del calcolo di CRC, $x^r P(x)$

Si moltiplica P(x), ricavato qui sopra, per x^r(shift di r posizioni, le stesse del poliniomio generatore), ottendo il polinomio $x^r P(x)$ di grado d+r-1, con coefficienti d+r.
In seguito allo shift, il campo FCS del frame sarà costutuito da r zeri.

Calcolo e verifica del CRC in trasmissione In trasmissione si divide il polinomio $x^r P(x)$ per G(x), il resto di questa operazione[il polinomio R(x)] sono i bit del CRC, da inserire nel FCS del frame. Vedi operatore XOR appendice basi.

$$\frac{x^r P(x)}{G(x)} = Q(x) \oplus \frac{R(x)}{G(x)} \tag{1.2}$$

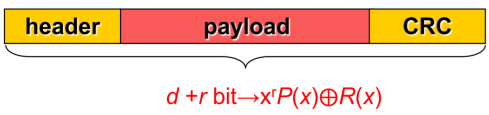


Figure 1.5: Inserimento bit in FCS

Devo quindi inserire i bit del polonomio R(x) all'interno del FCS, effettuo l'operazione:

$$P^{Tx} = x^r P(x) \oplus R(x) \tag{1.3}$$

Verifica CRC

C'è la necessità di verificare se il CRC calcolato ed inserito nel FCS sia corretto.

Polinomio relativo è necessario calcolare il polonomio relativo $P^{Rx}(x)$ per verificare CRC; questo polinomio non è altro che la trama di d+r bit ricevuti in ricezione.

Condizione necessaria Infine per verificare CRC c'è da tenere a mente la condizione necessaria affinchè il CSC sia corretto, ossia che il resto della divisione tra polinomio relativo[$P^{Rx}(x)$] e polonomio generatore[G(x)] sia nullo:

$$\frac{P^{Rx}(x)}{G(x)} = Q'(x) \oplus R'(x) \tag{1.4}$$

Può accadere che il resto sia nullo con trama ricevuta errata (la condizione è necessaria, non sufficiente).

Non si può quindi essere certi della correttezza della trama (c'è sempre una probabilità non nulla di falso positivo).

Se il resto NON è nullo, invece, c'è la certezza che la trama sia errata perché la condizione necessaria è stata violata

Dimostrazione condizione necessaria Nel caso in cui la trama ricevuta sia corretta, il polinomio che costruiamo da questa trama deve essere uguale al polinomio originale(trama trasmessa), cioè:

$$P^{Rx}(x) = P^{Tx}(x) \tag{1.5}$$

Se effettuiamo la divisione per il polinomio generatore:

$$\frac{P^{Rx}(x)}{G(x)} = \frac{P^{Tx}(x)}{G(x)} = \frac{x^r P(x) \oplus R(x)}{G(x)} = \frac{x^r P(x)}{G(x)} \oplus \frac{R(x)}{G(x)} \tag{1.6}$$

$$\frac{P^{Rx}(x)}{G(x)} = \frac{x^r P(x)}{G(x)} \oplus \frac{R(x)}{G(x)} = Q(x) \oplus \frac{R(x)}{G(x)} \oplus \frac{R(x)}{G(x)} = Q(x) \tag{1.7}$$

Esempio calcolo CRC

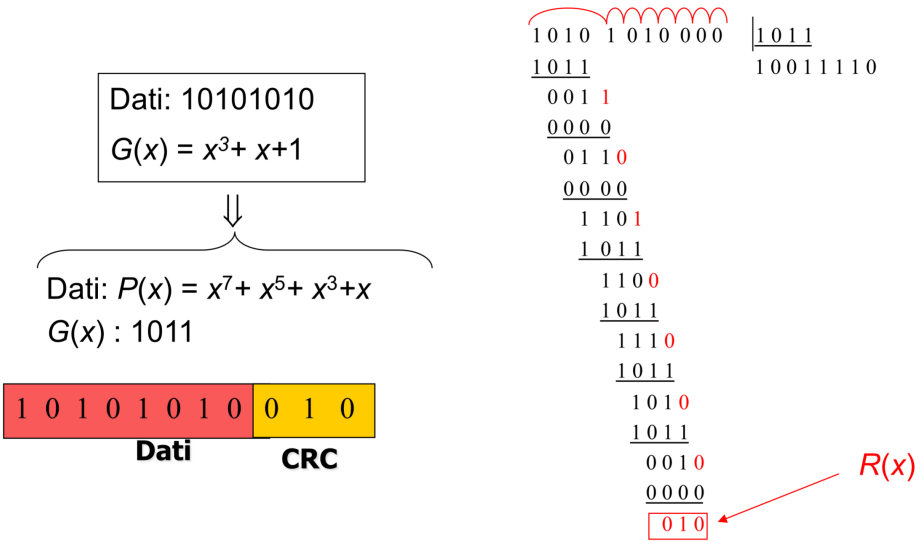
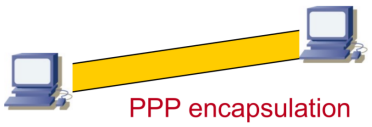


Figure 1.6: Esempio di calcolo CRC

1.1.4 Protocollo point to point (PPP)

IETF RFC 1661, 1662
protocollo di livello 2
punto-punto
diversi protocolli di livello superiore



Flag	Address	Control	Protocol	Information	FCS	Flag
8 bit	8×n bit	8/16 bit	16 bit	Variable (8 × m bit)	16/32 bit	8 bit

Figure 1.7: Protocollo Point to Point

Il protocollo Point to Point è comunemente usato nello stabilire connessioni dirette tra due nodi.

Il frame è quasi identico a quello del HDLC, ha un campo aggiuntivo, quello protocol: specifica il protocollo di livello 3 che stiamo trasportando.

Network bit order - Big Endian e Little Endian A differenza dello standard HDLC, big-endian, il protocollo PPP non è endianness specifico.

L'ordine dei byte (conosciuto anche come big-endian, little-endian o middle-endian a seconda dei metodi differenti), in informatica, indica modalità differenti usate dai calcolatori per immagazzinare all'interno della memoria dati di dimensione superiore al byte;dipende essenzialmente dall'architettura hardware usata.

"I termini big-endian e little-endian derivano dai nomi di due popolazioni che abitavano nelle favolose isole di Lilliput e Blefuscu nel romanzo I viaggi di Gulliver di Jonathan Swift. Queste erano entrate in rivalità per il modo in cui aprivano le uova - rompendo la punta o il fondo: a Lilliput, per editto dell'imperatore il cui figlio una volta si tagliò aprendo un uovo dall'estremità più grande, fu ordinato di aprire le uova dall'estremità più piccola (little endians); a Blefuscu si rifugiarono gli oppositori che volevano conservare la tradizione di rompere le uova dall'estremità più grande (big endians). A causa di questa differenza e della sua legittimazione imperiale era scoppiata tra le due isole una guerra sanguinosa."

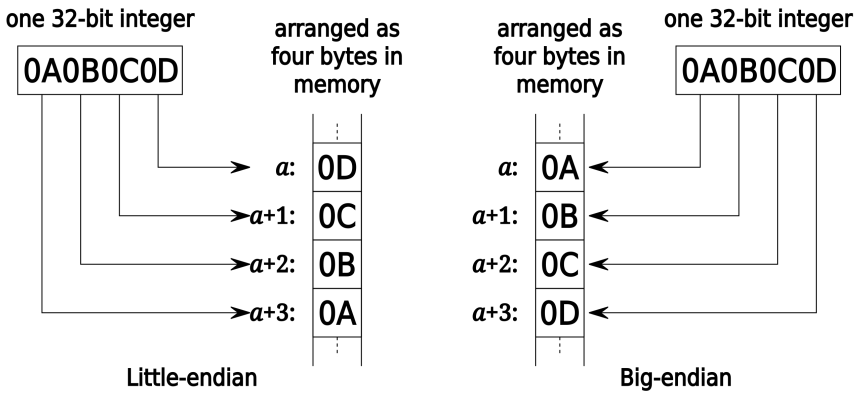


Figure 1.8: Big Endian e Little Endian

Byte stuffing

Anche qui, vista la struttura del pacchetto molto simile a quella del frame HDLC, c'è il rischio di confondere il dato informativo con un flag; perciò si attua il byte stuffing.

Sequenza di flag 0x7d Se si vuole evitare la sequenza 0x7e (01111110), si antepone la sequenza di controllo 0x7d (01111101); al posto della sequenza 0x7e inseriamo la sua versione in XOR con la sequenza 0x20 (00100000).

A quel punto avremo individuato la sequenza "pericolosa" all'interno del frame, ossia 0x7e, avremo un identificativo, 0x7d, che ci dice dov'è situata nel frame e la sostituiremo con il suo valore in XOR con 0x20.

Quindi quando all'interno del frame leggo il byte 0x7d, il nuovo flag che indica che c'è stato byte stuffing, non leggiamo 0x7d ma il byte successivo, quello calcolato tramite il 0x7e XOR 0x20.

In ricezione se applico nuovamente XOR 0x20 al byte precedentemente modificato con lo XOR, allora otterrò nuovamente il byte iniziale, 0x7e.

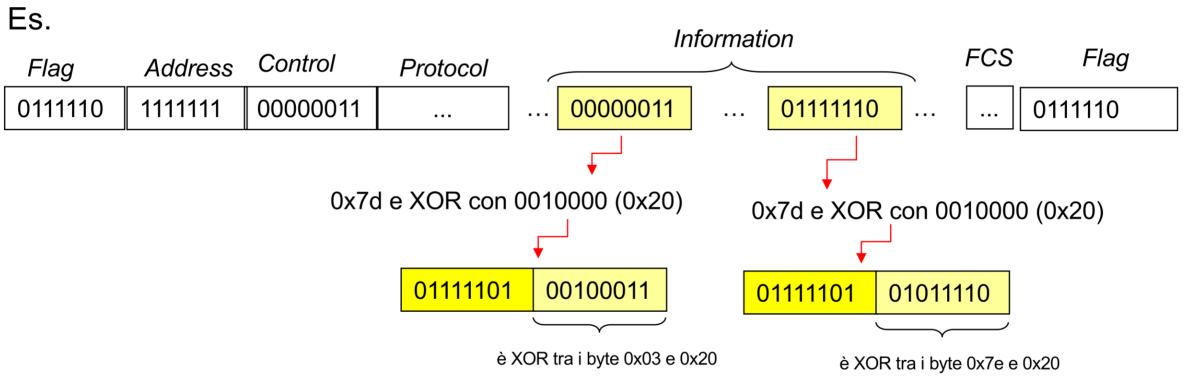


Figure 1.9: Esempio di byte stuffing

Nel caso in cui incappiamo con una "falsa" sequenza di flag del byte stuffing, ossia 0x7d, la trasmettermo come 0x7d, 0x5d.

Struttura PPP

Il PPP utilizza due protocolli per la gestione del collegamento tra nodi:

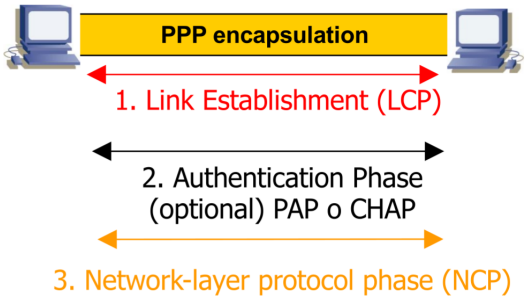


Figure 1.10: Encapsulation PPP

- LCP(Link Control Protocol): grazie al quale si apre la connessione, si ha la verifica della qualità del collegamento e rende possibile l'autenticazione(PAP o CHAP)
- NCP(Network Control Protocol): con cui è possible scegliere e configurare uno o più protocolli di rete(es. IP)

Autenticazioni

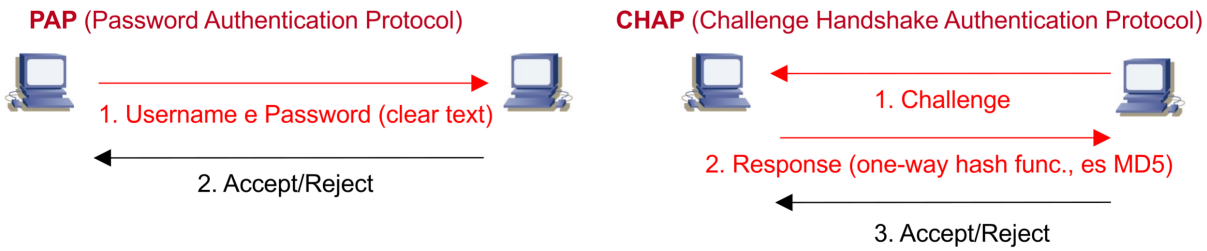


Figure 1.11: Autenticazione PAP e CHAP

Autenticazione PAP Password Authentication Protocol (PAP) invia username e password in chiaro. È semplice ma poco sicuro, poiché le credenziali possono essere intercettate(password e username sono solitamente criptate).

Autenticazione CHAP Challenge Handshake Authentication Protocol (CHAP) utilizza un meccanismo di challenge-response con hash per autenticare in modo più sicuro, evitando di trasmettere la password in chiaro.

1.2 Protocolli di accesso

A livello due viene svolto un ruolo importante dai protocolli di accesso, che possono essere di due tipologie:

- Accesso ordinato:
  - Ogni nodo ha un turno prestabilito per trasmettere, evitando collisioni.
  - Utilizzato in sistemi come il polling o il token passing.
  - Garantisce un utilizzo equo del canale, ma può introdurre ritardi se un nodo non ha dati da trasmettere.
- Accesso casuale:
  - senza rilevazione del canale:
    - I nodi trasmettono senza verificare se il canale è libero.
    - Può portare a collisioni frequenti, come nel protocollo ALOHA puro.
  - con rilevazione del canale:
    - senza rivelazione di collisioni:
      - I nodi verificano se il canale è libero prima di trasmettere.
      - Non rilevano collisioni, quindi i dati persi devono essere ritrasmessi dopo un timeout.
      - Esempio: Carrier Sense Multiple Access (CSMA).
    - con rilevazione di collisioni:
      - I nodi verificano se il canale è libero e rilevano eventuali collisioni durante la trasm.
      - In caso di collisione, interrompono la trasm. e ritentano dopo un intervallo casuale.
      - Esempio: CSMA/CD (utilizzato in Ethernet).

1.2.1 Troughput di rete

Calcolo del Troughput

1.2.2 Protocollo ALOHA puro

ALOHA è un protocollo di accesso multiplo casuale senza rilevazione del canale,

Calcolo del Troughput di ALOHA puro

Prestazioni ALOHA puro

1.2.3 Protocollo ALOHA slotted

Prestazioni ALOHA slotted

1.2.4 Protocolli Carrier Sense Multiple Access - CSMA

CSMA/CD

CSMA/CA