



Lezione 5: Livello di Collegamento II Sottolivello Data-Link



Sommario della Lezione

- ❑ Ancora sui codici d'errore
- ❑ Controllo di flusso
 - Protocolli elementari stop and wait
 - Protocolli sliding window
- ❑ HDLC



Ancora sui codici d'errore



Distanza di Hamming

- ❑ Consideriamo due parole di codice

01100011

01110101

- ❑ E' possibile determinare quanti bits differiscono.
- ❑ Distanza di Hamming = 3 bits
- ❑ Se due parole di codice hanno una distanza di Hamming d , vuol dire che saranno necessari d errori di singoli bit per convertire l'una nell'altra.



Codifica a blocco per controllo d'errore

n bit



m bit utente

n-m bit parità

2^m possibili combinazioni



Rilevazione-Correzione Errori

- ❑ Per rilevare d errori è necessario un codice con distanza $d+1$, poichè con questo codice non c'è modo che d errori di bit singolo mutino una codeword valida in un'altra codeword valida.
- ❑ Per correggere d errori è necessario un codice con distanza di $2d+1$ poichè le sue parole valide sono così distanti che anche con d mutazioni, la parola originale è ancora la più vicina valida e può quindi essere individuata.



Correzione errori

- ❑ Immaginiamo di voler progettare un codice con m bits di messaggio e r bits di controllo che garantisca la correzione di tutti gli errori singoli.
- ❑ Ognuno dei 2^m messaggi legali ha n parole di codice illegali a distanza 1.
- ❑ Ognuno dei 2^m messaggi legali necessita $n+1$ combinazioni di bit dedicate. Poichè il numero totale di combinazioni è 2^n , si devono avere
$$(n+1)2^m \leq 2^n.$$
- ❑ Usando $n=m+r$, questo requisito diventa
$$(m+r+1) \leq 2^r.$$
- ❑ Dato m , questo definisce un limite inferiore al numero di bit di controllo necessari per correggere un errore singolo.



Checksum: Cyclic Redundancy Check (1)

❑ **Codici polinomiali (CRC=Cyclic Redundancy Code)**

- I bit di una stringa M di m bit da proteggere sono visti come i coefficienti (0 e 1) di un polinomio $M(x)$; l' i -esimo bit è il coefficiente di x^{i-1}
- Ad una stringa di k bit corrisponde un polinomio di grado $k-1$

continua....





Checksum: Cyclic Redundancy Check (2)

- ❑ Quando si usano CRC, trasmettitore e ricevitore si devono accordare su un polinomio generatore $G(x)$ in anticipo. Sia il primo che l'ultimo bit del pattern generato devono essere uguali a 1.
- ❑ Per calcolare la checksum per un frame di m bits, corrispondente al polinomio $M(x)$, il frame dev'essere più lungo del polinomio generatore.
- ❑ L'idea è di appendere la checksum alla fine del frame in modo tale che il polinomio rappresentato dal frame e dalla checksum sia divisibile per $G(x)$.
- ❑ Quando il ricevitore riceve il frame con la checksum, cerca di dividerlo per $G(x)$. Se c'è un resto allora significa che c'è stato un errore.



Checksum: Cyclic Redundancy Check (3)

- ❑ Si appendono r zeri in coda alla stringa da proteggere, così si ottiene una stringa di $m+r$ bit che corrisponde al polinomio $x^rM(x)$
- ❑ Si divide $x^rM(x)$ per $G(x)$, usando la divisione modulo 2, e si calcolano il quoziente $Q(x)$ e il resto $R(x)$ (polinomio di grado $r-1$)
- ❑ Si sottrae il resto (formato al più da r bit) da $x^rM(x)$ usando la sottrazione modulo 2, il risultato è la trama da trasmettere

$$\mathbf{T(x) = x^rM(x) + R(x) = G(x)Q(x)}$$

- ❑ Il polinomio risultante $T(x)$ è divisibile per $G(x)$



Checksum: Cyclic Redundancy Check (4)

- ❑ In ricezione si ottiene $Y(x) = T(x) + E(x)$, dove $E(x)$ rappresenta l'eventuale sequenza di errori
- ❑ Si calcola il resto della divisione di $Y(x)/G(x)$; se il resto è zero la stringa ricevuta è corretta (cioè $E(x)=0$)
 - ❑ Nota: il CRC fallisce se $E(x)$ è divisibile per $G(x)$
- ❑ Per errori singoli $E(x)=x_j$, che non è divisibile per $G(x)$ se $G(x)$ contiene almeno due termini
- ❑ Per un qualunque numero dispari di errori, $E(x)$ non è divisibile per $G(x)$ se $G(x)$ contiene il fattore $(x+1)$
- ❑ Un qualsiasi burst di errori è rivelato, in quanto $E(x)=x_j(x_k-1+\dots+1)$, con $k=r$ e $G(0)=1$
- ❑ Per ogni $G(x)$ esiste un valore minimo di n tale che x^{n+1} è divisibile per $G(x)$, e $n=2r-1$



Checksum: Cyclic Redundancy Check (5)

- ❑ Il CRC rivela tutti gli errori a burst con lunghezza minore di $r+1$ bit
 - tecnica usata da ATM, HDLC, ecc.



Polinomi utilizzati per $G(x)$

□ $x^{12} + x^{11} + x^3 + x^2 + x^1 + 1$

CRC-12

□ $x^{16} + x^{15} + x^2 + 1$

CRC-16

□ $x^{16} + x^{12} + x^5 + 1$

CRC-CCITT



ARQ (Automatic Repeat reQuest)

controllo congiunto di

- ☐ errore
- ☐ flusso
- ☐ sequenza

su una connessione



Controllo di flusso

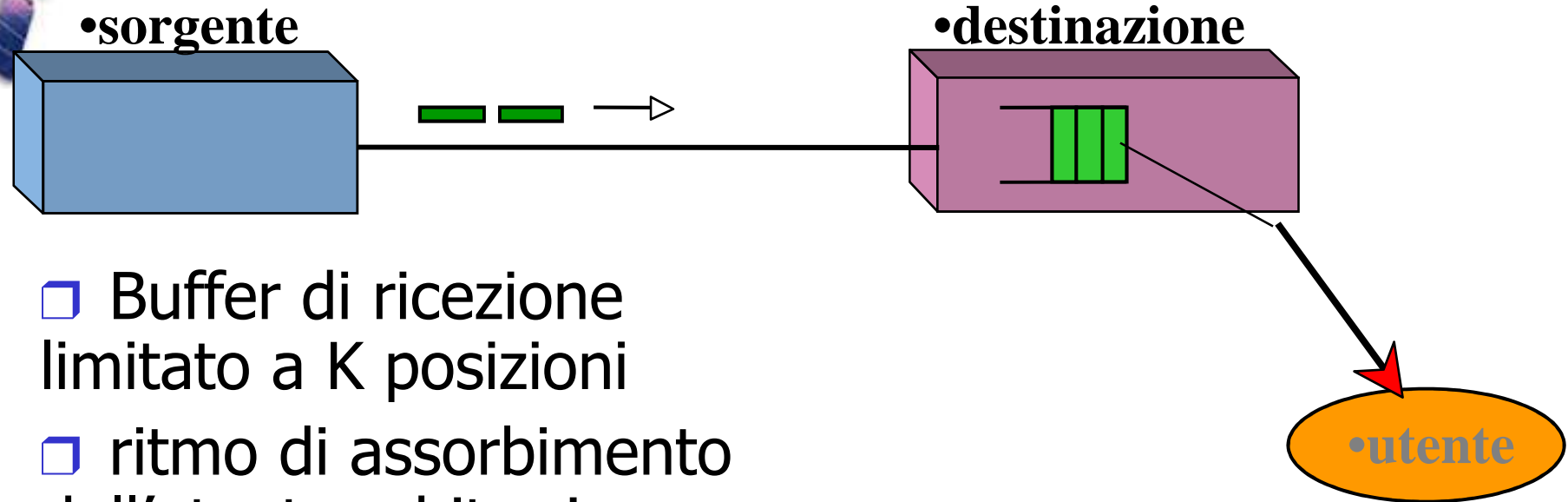
❑ Obiettivo:

- regolare la velocità di invio delle unità informative da una sorgente ad una destinazione in modo che tale velocità non sia superiore a quella con la quale le unità informative vengono smaltite a destinazione

❑ Livelli

- livello di linea (2)
- livello di trasporto (4)

Schema di esempio



- ❑ Buffer di ricezione limitato a K posizioni
- ❑ ritmo di assorbimento dell'utente arbitrario
- ❑ obiettivo: evitare che pacchetti vadano persi perché all'arrivo trovano il buffer pieno



Tecniche ARQ

- ❑ **Stop and wait (Alternating bit)**
- ❑ **Go back N**
- ❑ **Selective repeat**



Protocollo *stop and wait* (1)

il trasmettitore

1. invia una PDU
2. attiva un orologio (tempo di timeout)
3. si pone in attesa della conferma di ricezione (acknowledgment - ACK)
4. se scade il timeout prima dell'arrivo della conferma, ripete la trasmissione

il trasmettitore, quando riceve un ACK

1. controlla la correttezza dell'ACK
2. controlla il numero di sequenza
3. se l'ACK è relativo all'ultima PDU trasmessa, si abilita la trasmissione della prossima PDU

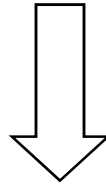
il ricevitore

1. riceve una PDU
2. controlla la correttezza della PDU
3. controlla il numero di sequenza
4. se la PDU è corretta invia la conferma di ricezione



Protocollo *stop and wait* (2)

- Problemi:
 - stretta alternanza di flusso nelle trasmissioni mittente-ricevente-mittente-ricevente...
Sarebbe sufficiente un canale half-duplex
 - problemi se il frame di riscontro si perde nella trasmissione



Il mittente deve attendere un ack positivo prima di continuare a trasmettere: Protocolli con **PAR** (Positive Acknowledgement with Retrasmission) oppure **ARQ** (Automatic Repeat reQuest)



Piggybacking

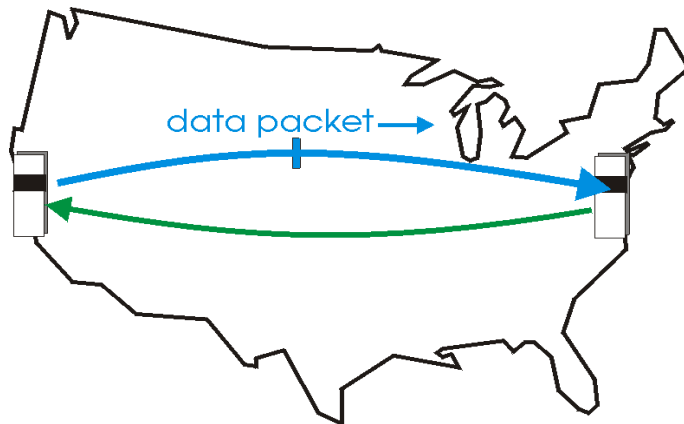
Quando il flusso dati è bidirezionale è possibile includere nell'intestazione della PDU dati un campo con l'informazione di riscontro (ACK) per il flusso dati che sta fluendo in direzione opposta

La tecnica è detta “piggybacking”

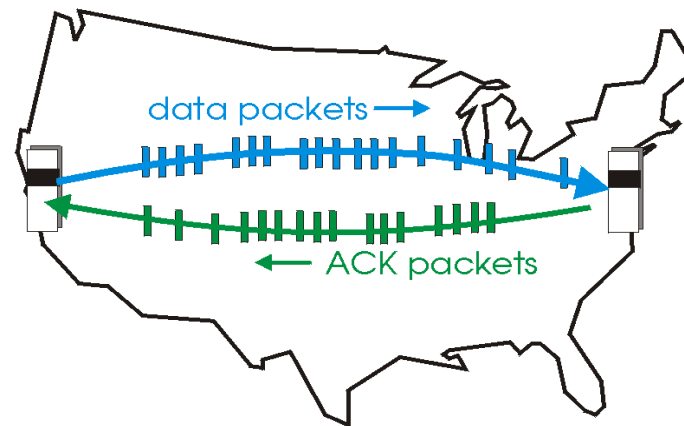
Protocolli Pipeline

Pipelining: al sender è consentito l'invio di pacchetti multipli senza che debba aspettare i riscontri

- i pacchetti in transito è come se riempissero un canale (pipeline)
- il range di numeri di sequenza aumenta
- necessità di buffering al sender e/o al receiver



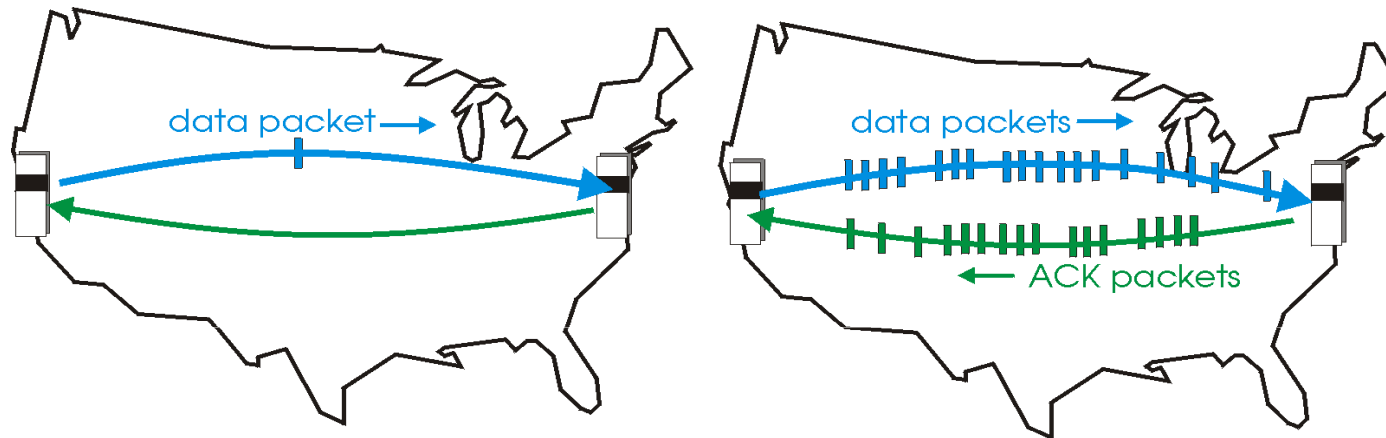
(a) a stop-and-wait protocol in operation



(b) a pipelined protocol in operation



Protocolli Pipeline



(a) a stop-and-wait protocol in operation

(b) a pipelined protocol in operation

Due tipi di protocolli pipeline: *go-Back-N, selective repeat*



Metodi di trattare gli errori nel pipelining: go back n

- ❑ Consiste nel far in modo che il ricevente scarti tutti i pacchetti successivi ad uno in errore, non spedendo alcun ack per quelli scartati
- ❑ Finestra ricevente di dimensione 1
- ❑ Il data link si rifiuta di accettare pacchetti, eccetto quello che, in sequenza, deve passare al livello rete. Quando scade il timer del mittente, esso ricomincerà a trasmettere dall'ultimo pacchetto non ricevuto.
- ❑ **Svantaggio:** spreco di banda se il tasso d'errore è elevato



Metodi di trattare gli errori nel pipelining: selective repeat

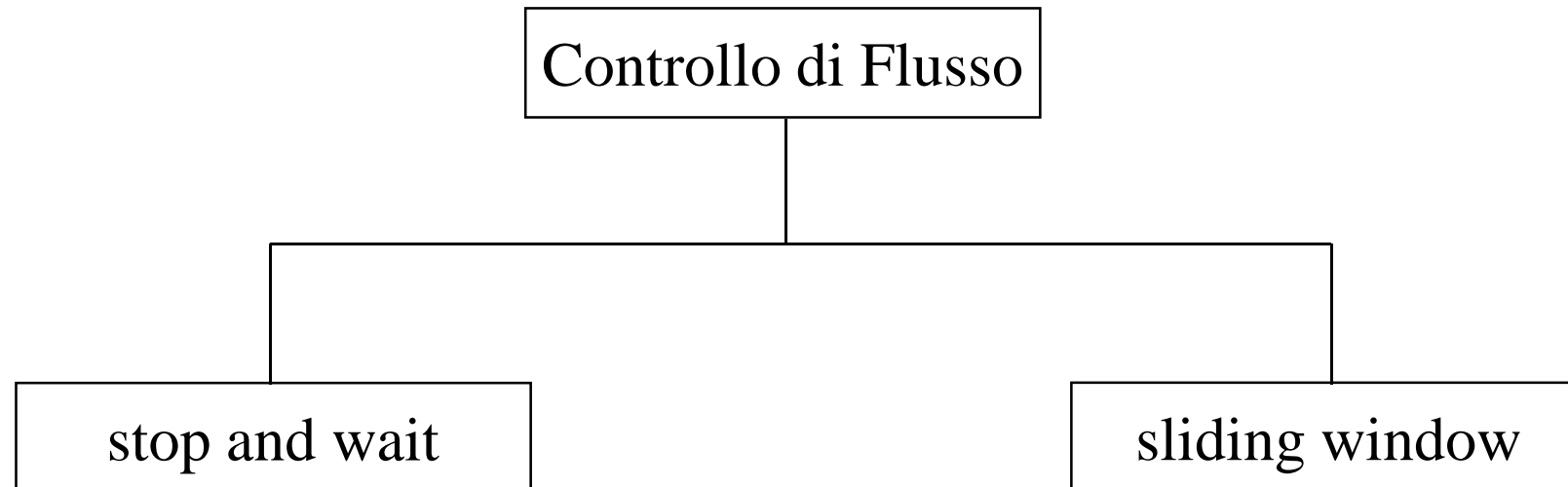
- ❑ Consiste nel fare in modo che il livello data link del ricevente memorizzi tutti i pacchetti corretti che seguono quello danneggiato.
- ❑ In questo caso il mittente dovrà ritrasmettere solo quel pacchetto.



Controllo di flusso



Controllo di flusso



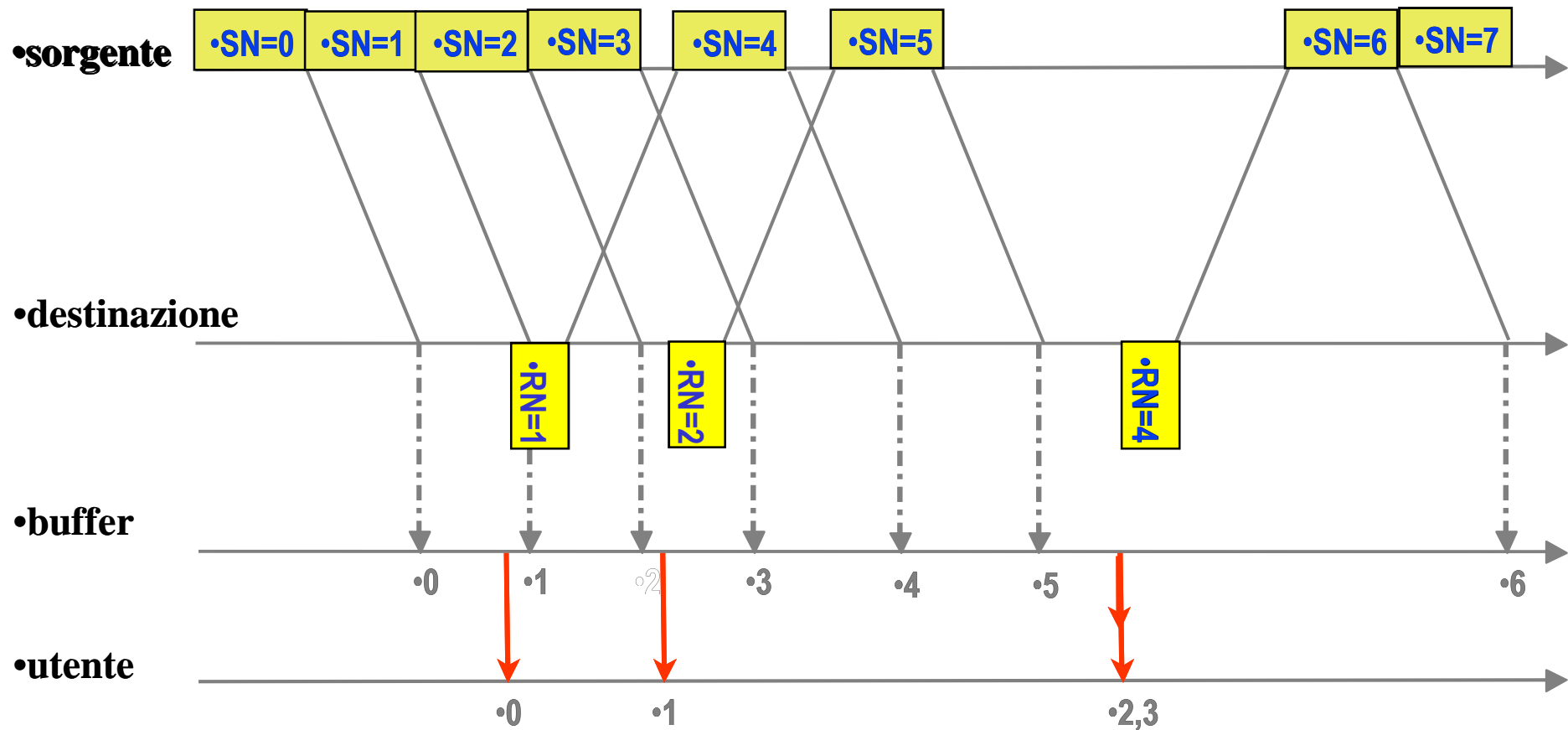


Protocolli *sliding window* (1)

- ❑ è possibile usare un meccanismo come quello del go back n
- ❑ la sorgente non può inviare più di W trame (stessa funzione del parametro n) senza aver avuto il riscontro
- ❑ i riscontri vengono inviati dal ricevitore solo quando i pacchetti vengono assorbiti dall'utente



Protocolli *sliding window* (2)



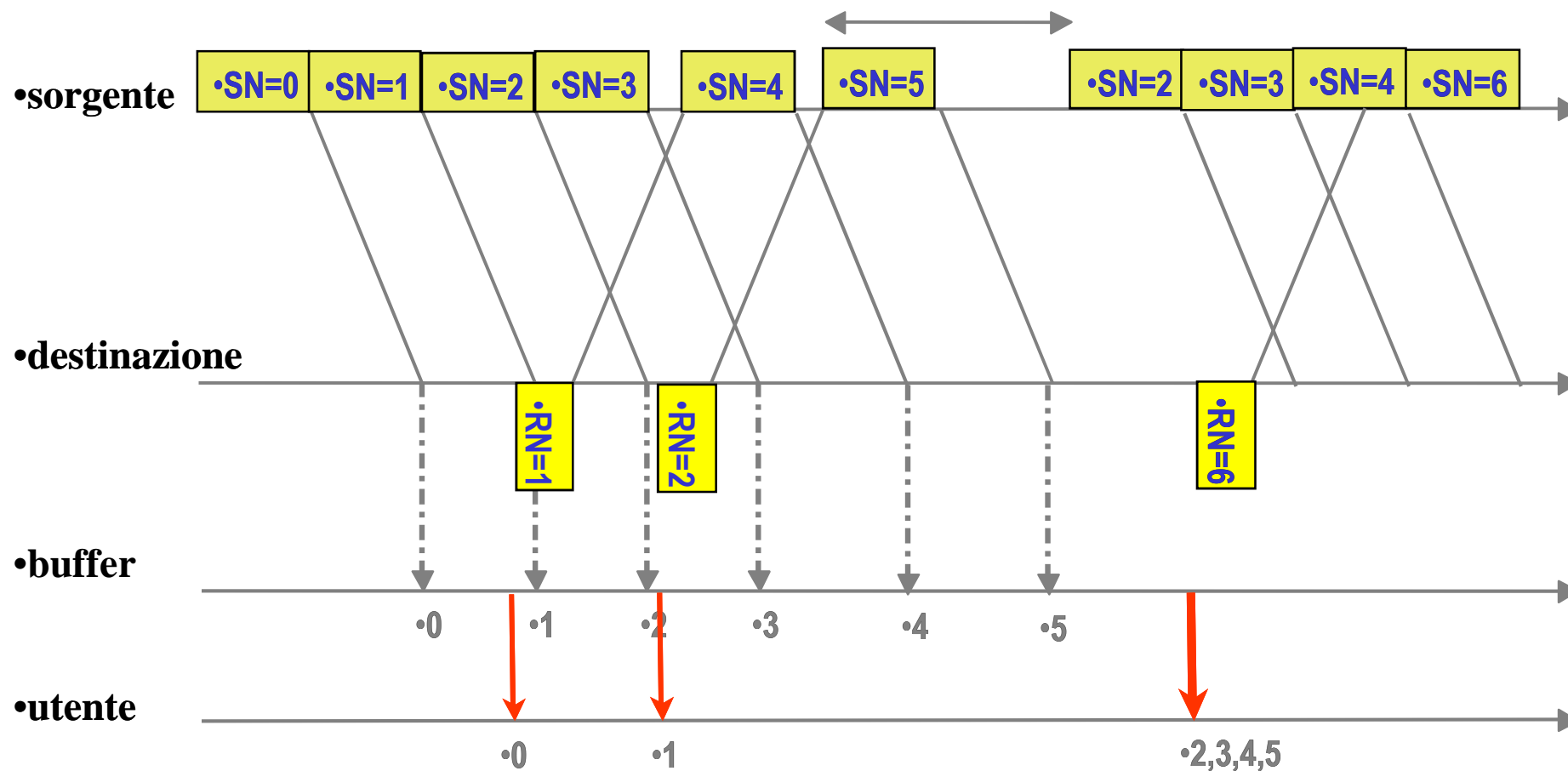


Ritrasmissioni

- ❑ il meccanismo descritto di controllo di flusso è strettamente legato al meccanismo di controllo d'errore e questo può essere fonte di problemi
- ❑ se il ricevitore ritarda molto l'invio dei riscontri, il trasmettitore inizia la ritrasmissione perché scade il time-out
- ❑ aumentare troppo il time-out non è ovviamente una soluzione radicale in quanto l'aumento del time-out aumenta i ritardi in caso di errore



Protocolli *sliding window* (3)



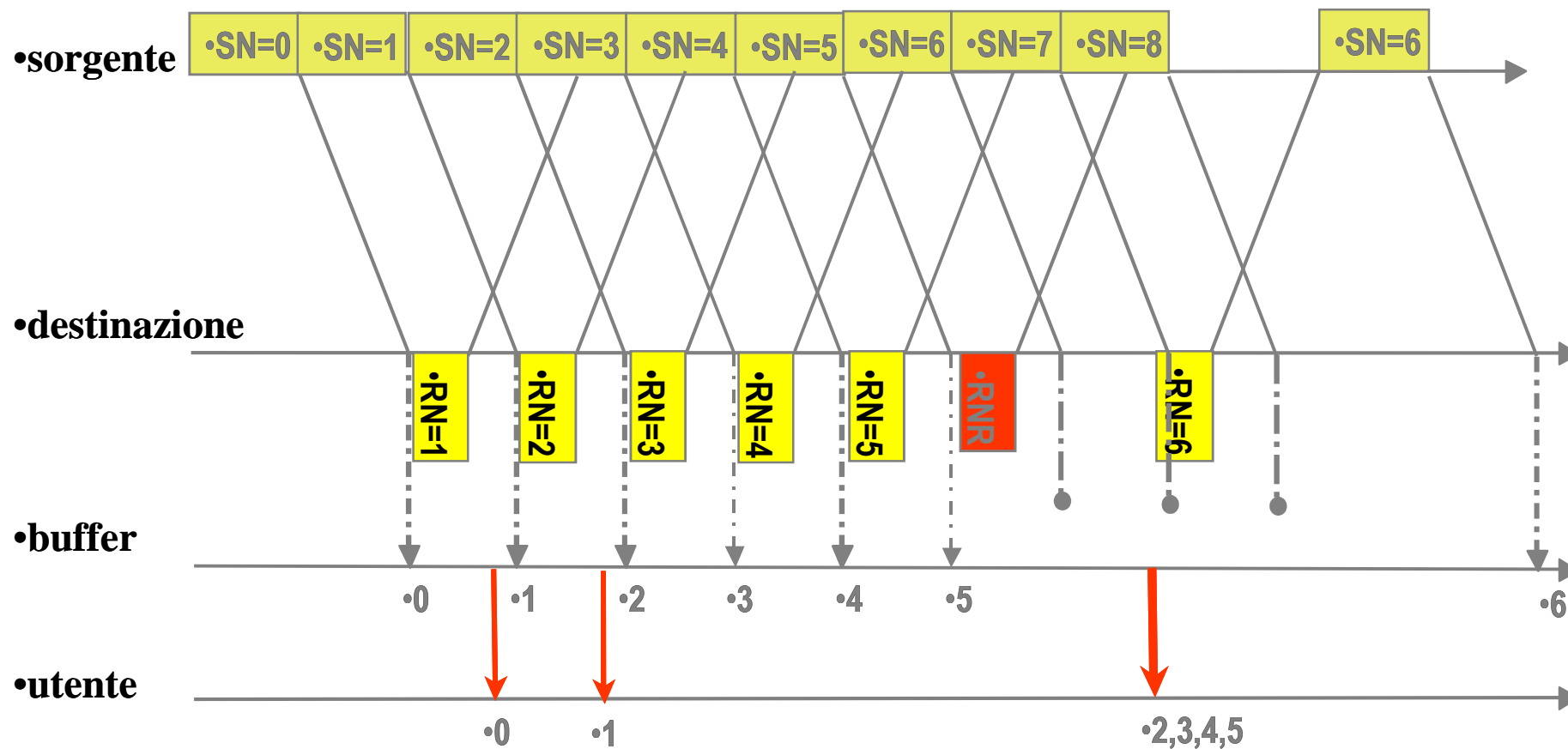


Uso del messaggio RNR

- ❑ Il problema può essere affrontato
 - trasmettendo regolarmente i riscontri all'arrivo dei pacchetti
 - usando un messaggio speciale di RNR (Receiver Not Ready) per segnalare che il buffer è pieno
- ❑ approccio ibrido:
 - ritardare l'invio dei riscontri per un tempo massimo e poi usare il RNR



Uso del messaggio RNR: esempio



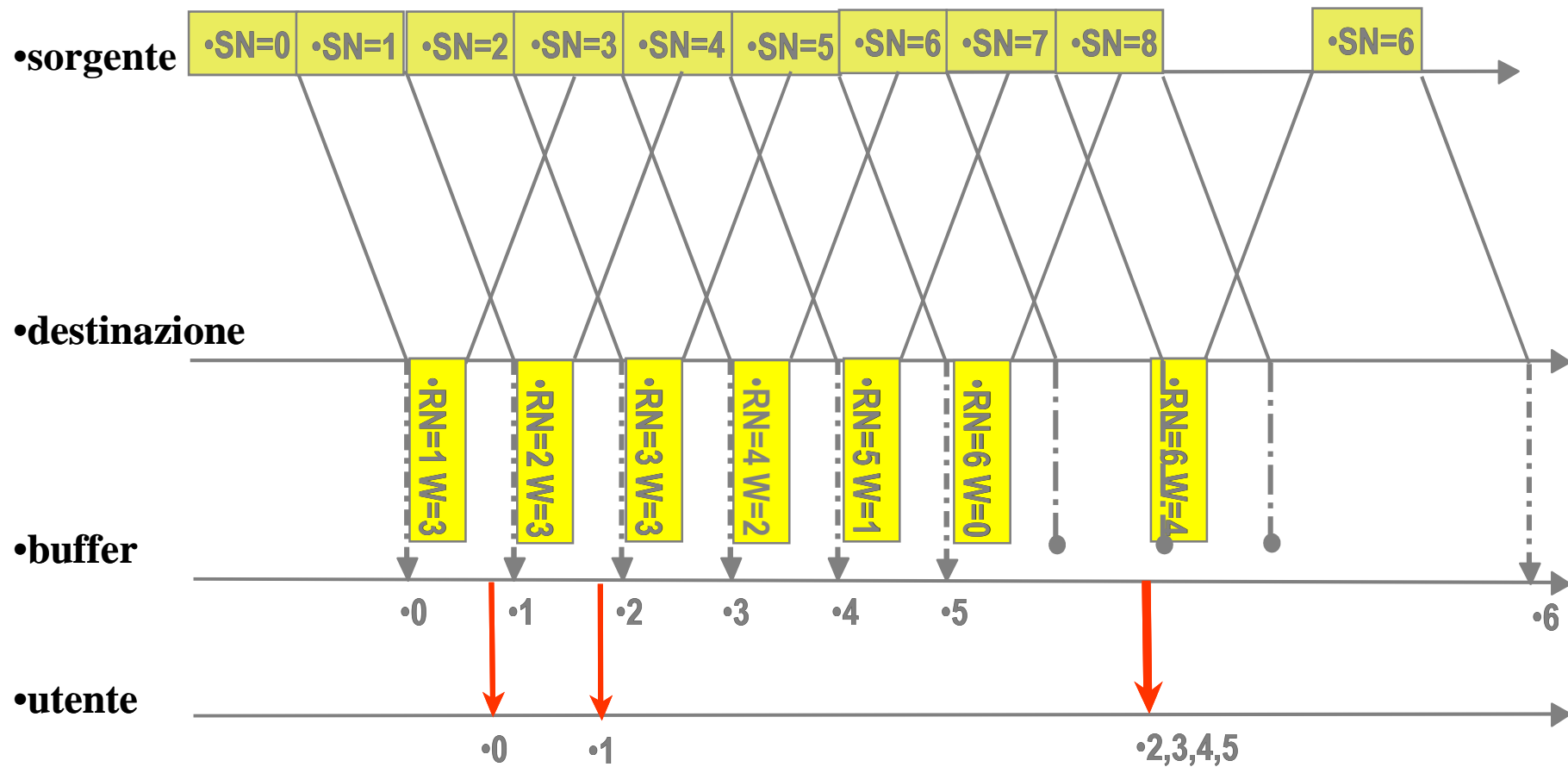


Uso del campo W

- ❑ Il problema può essere risolto in modo radicale separando i meccanismi di controllo d'errore e di controllo di flusso a finestra
- ❑ si inserisce nei riscontri (o nell'header delle trame in direzione opposta) un campo finestra W
- ❑ il ricevitore invia i riscontri sulla base dell'arrivo dei pacchetti
- ❑ usa il campo W per indicare lo spazio rimanente nel buffer
- ❑ è questo l'approccio usato nel TCP



Uso del campo W: esempio





Uso del campo W

- gestione della finestra
 - non è necessario che il ricevitore dica la “verità” sullo spazio restante R
 - può tenersi un margine di sicurezza ($W=R-m$)
 - può aspettare che il buffer si sia svuotato per una frazione (ad es. $W=0$ se $R < K/2$ ed $W=R$ altrimenti)
 - può usare dei meccanismi adattativi



Protocollo HDLC



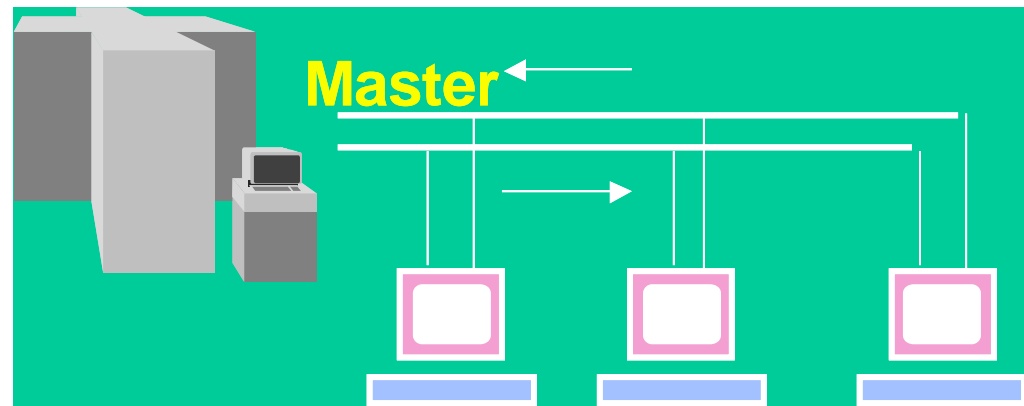
HDLC High-level Data Link Communications

- ❑ standard ISO
- ❑ deriva dal protocollo proprietario SDLC (Synchronous Data Link Control) di IBM per reti SNA
- ❑ può operare in molti modi differenti e con diversi meccanismi di controllo d'errore e di flusso
 - half-duplex o full-duplex
 - master-slave (sbilanciato) o peer-to-peer (bilanciato)

Modalità di funzionamento (1)

□ **Normal Response Mode (NRM)**

- Una stazione primaria è collegata a una o più stazioni secondarie tipicamente in modalità half-duplex. Solo la stazione primaria può inviare i comandi e le stazioni secondarie trasmettono solo a seguito di un permesso (polling) esplicito inviato dalla stazione primaria

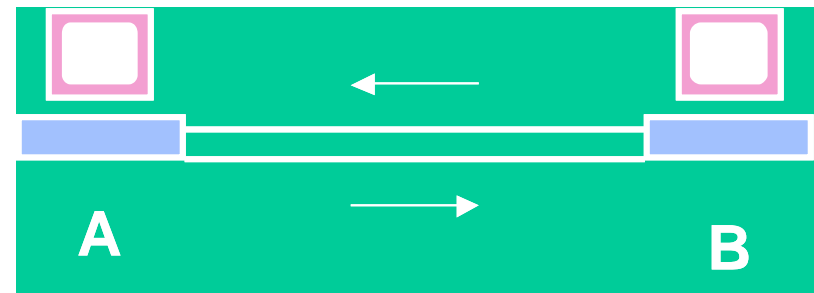




Modalità di funzionamento (2)

□ **Asynchronous Response Mode (ARM)**

- Anche in questo caso come nel NRM il colloquio è di tipo sbilanciato, ma la stazione secondaria ha la possibilità di iniziare una trasmissione senza il permesso esplicito della stazione primaria iniziando così un colloquio full-duplex. (poco usata)

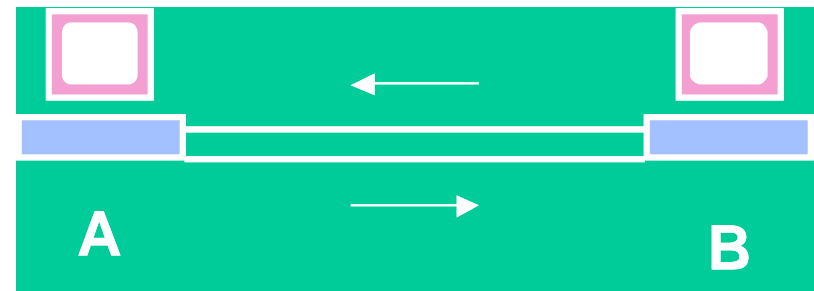




Modalità di funzionamento

□ **Asynchronous Balanced Mode (ABM)**

- Fornisce una modalità di funzionamento bilanciato su configurazioni punto-punto tra stazioni "combinate" che possono, in modalità full-duplex, inviare informazioni in modo indipendente ed asincrono.
- E' l'unico modo di funzionamento conforme con lo stack OSI!





HDLC Modes

| | NRM | ARM | ABM |
|--------------|---------------------|---------------------|----------|
| Station type | Primary & secondary | Primary & secondary | Combined |
| Initiator | Primary | Either | Any |



Formato della trama HDLC (1)



Header

Trailer



Formato della trama HDLC (2)

1 byte

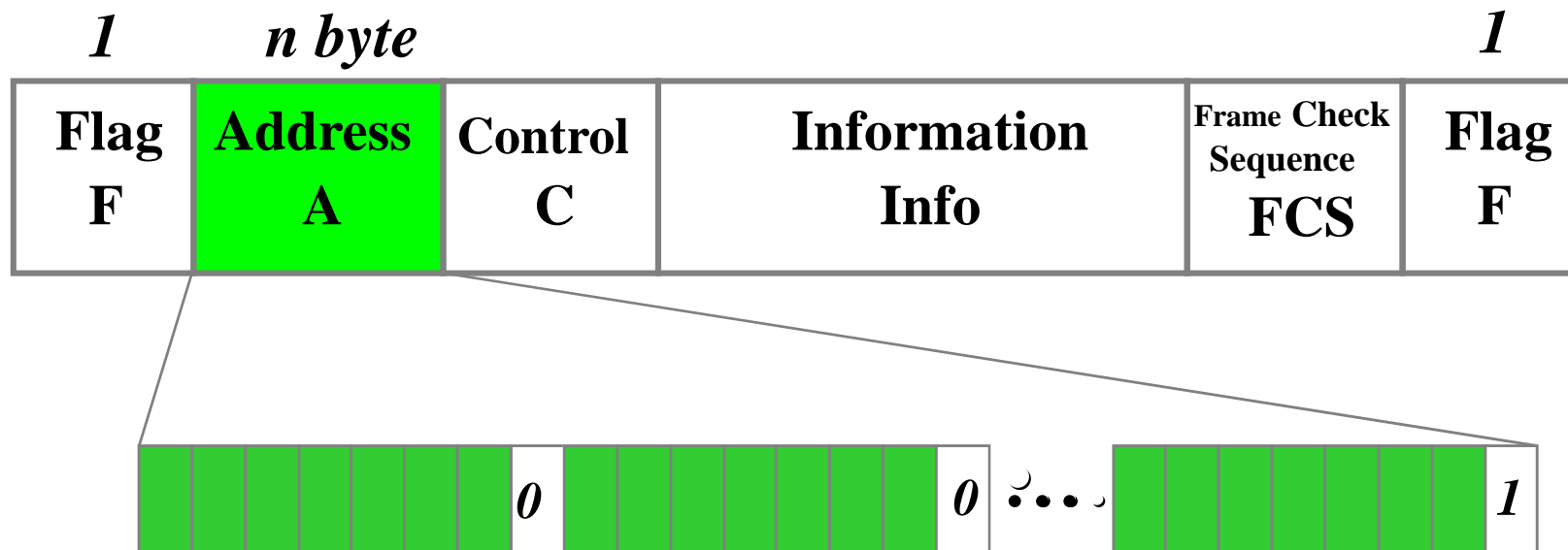
1 byte

| | | | | | |
|-------------------------|----------------------------|----------------------------|-------------------------------------|--|-------------------------|
| Flag F | Address A | Control C | •Information •Info | Frame Check Sequence •FCS | Flag F |
|-------------------------|----------------------------|----------------------------|-------------------------------------|--|-------------------------|

Flag *01111110*

- uso del bit stuffing
- in alcune configurazioni trasmissione continua dei flag in caso di mancanza di informazione

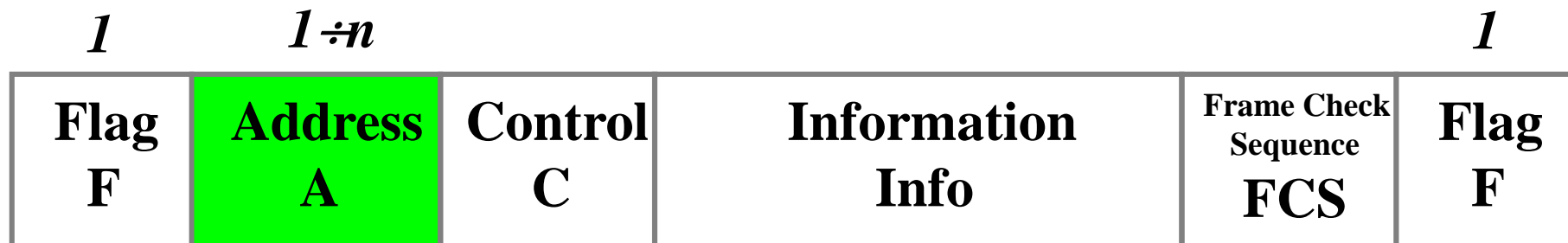
Formato della trama HDLC (3)



- normalmente di 8 bit, ma può essere di n byte
- l'ultimo bit di ogni byte è usato per indicare se segue un ulteriore byte del campo A



Formato della trama HDLC (4)



- L'indirizzo contenuto può essere quello della stazione destinataria o quello della stazione sorgente
 - nelle modalità sbilanciate (NRM, ARM) è sempre quello della stazione secondaria (sia nei comandi sia nelle risposte)
 - nella modalità ABM è quello della stazione destinataria

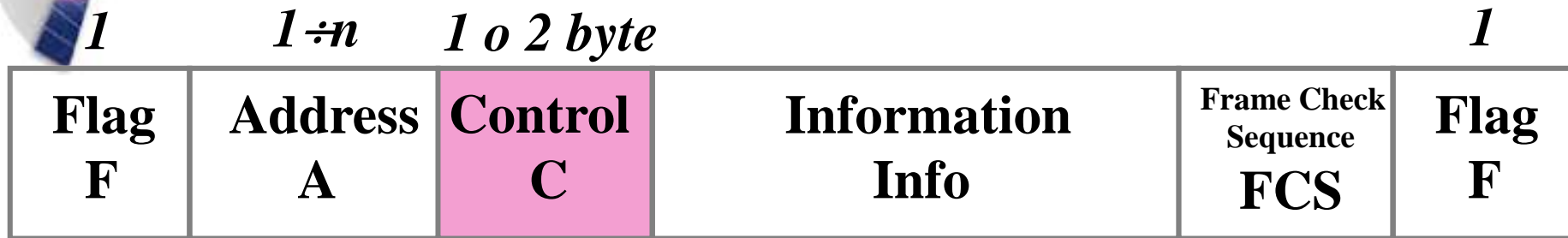


Formato della trama HDLC (5)

| | | | | | |
|-------------------------|----------------------------|----------------------------|-----------------------------------|---|-------------------------|
| <i>1</i> | <i>1 ÷ n</i> | <i>1 o 2 byte</i> | | | <i>1</i> |
| Flag F | Address A | Control C | Information Info | Frame Check Sequence FCS | Flag F |

- Il campo controllo distingue i tipi di trama e contiene le informazioni di controllo relative ad ogni tipo
- i primi bit distinguono il tipo
- gli altri contengono il controllo vero e proprio

Formato della trama HDLC (6)



Information (I)

- Sono trame numerate per la trasmissione di informazione d'utente contenuta nel campo.

Supervisory (S)

- Sono trame numerate per il controllo dell'invio del flusso di informazione (ad es. riscontri non associati ad informazione in senso opposto).

Unnumbered (U)

- Sono trame non numerate usate per l'invio di informazione di controllo (ad es. per l'instaurazione delle connessioni) o per l'invio di informazione in modalità senza connessione.



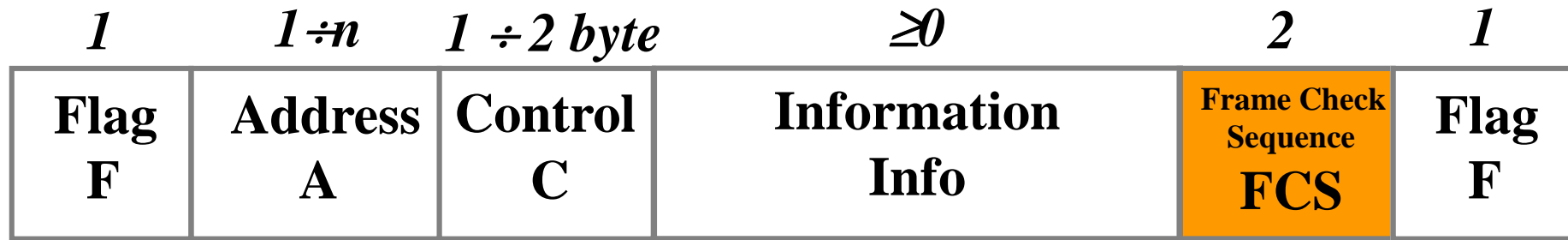
Formato della trama HDLC (7)

| | | | | | |
|-------------------------|----------------------------|----------------------------|-----------------------------------|---|-------------------------|
| <i>1</i> | <i>1 ÷ n</i> | <i>1 ÷ 2 byte</i> | <i>≥ 0</i> | | <i>1</i> |
| Flag F | Address A | Control C | Information Info | Frame Check Sequence FCS | Flag F |

- ❑ Contiene l'informazione d'utente (dei livelli superiori)
- ❑ può non essere presente
 - è presente solo nella trame I e nella trame U usate per trasferimento di informazione in modalità connectionless
- ❑ lunghezza variabile

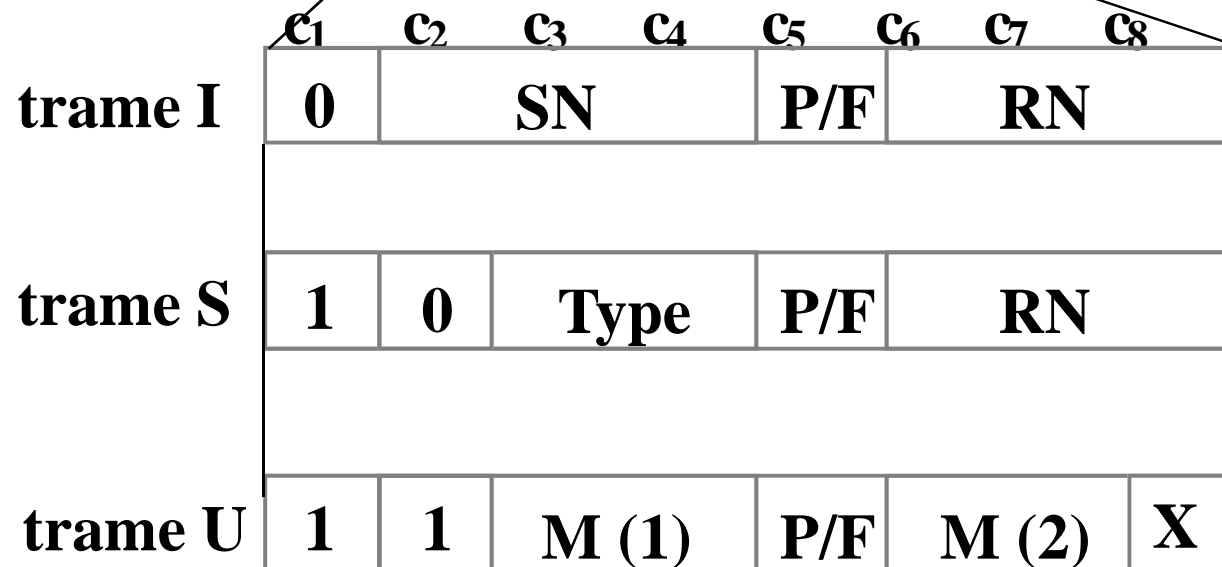


Formato della trama HDLC (8)



- ❑ Contiene il codice rivelatore d'errore usato per riconoscere le trame errate

Formato della trama HDLC (9)





Formato trame di controllo HDLC: Trame S

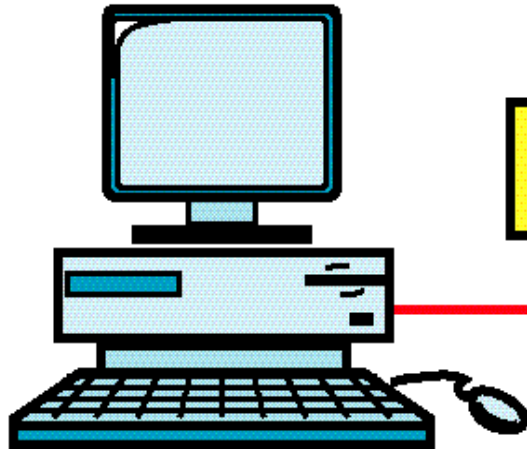
| | | | | | |
|----------------|----------|----------|-------------|------------|-----------|
| trame S | 1 | 0 | Type | P/F | RN |
|----------------|----------|----------|-------------|------------|-----------|

- ❑ Sono usate per inviare riscontri (senza piggyback) e controllo
 - RR (Receiver Ready), campo type 00, che è normalmente usato come ACK e il campo RN contiene la prossima trama attesa (riscontro delle trame fino a RN-1)
 - RNR (Receiver Not Ready), campo type 10, serve a bloccare l'invio di trame da parte dell'altra stazione e, contemporaneamente a riscontare le trame fino a RN-1
 - REJ (Reject), campo type 01, serve a richiedere la ritrasmissione delle trame da RN in avanti e, contemporaneamente, a riscontrare le trame fino a RN-1
 - SREJ (Selective Reject), campo type 11, è usato per richiedere la ritrasmissione della sola trama con numero RN



Uso del campo Poll/Final (1)

Primary



P = 1, RR

Secondary



Poll

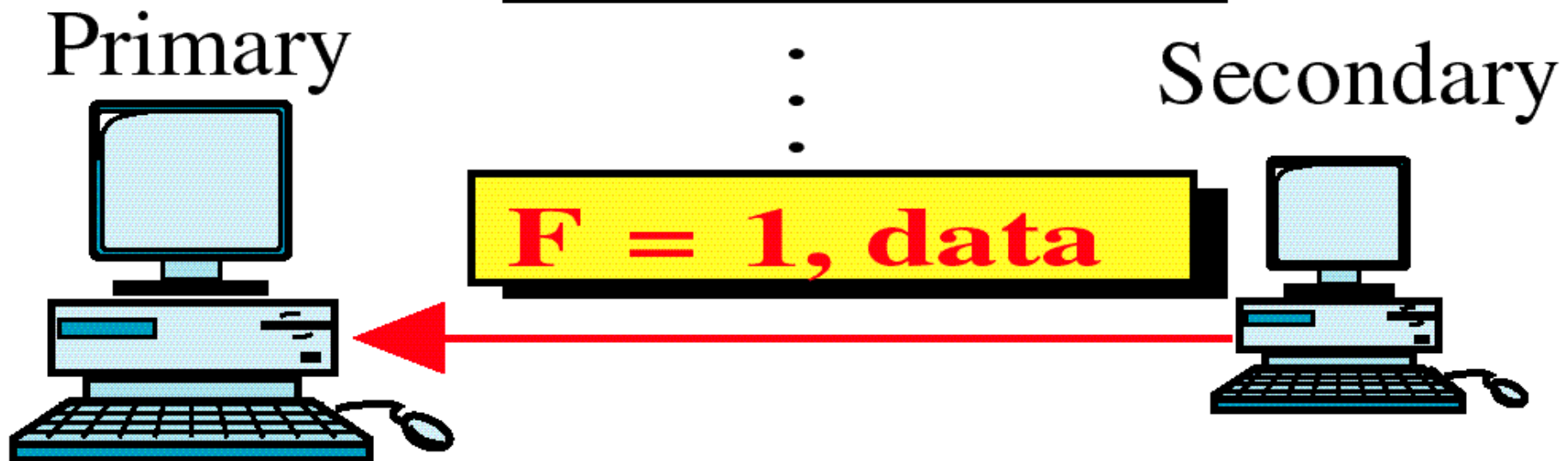


Uso del campo Poll/Final (2)

$F = 0$, data

⋮

$F = 1$, data

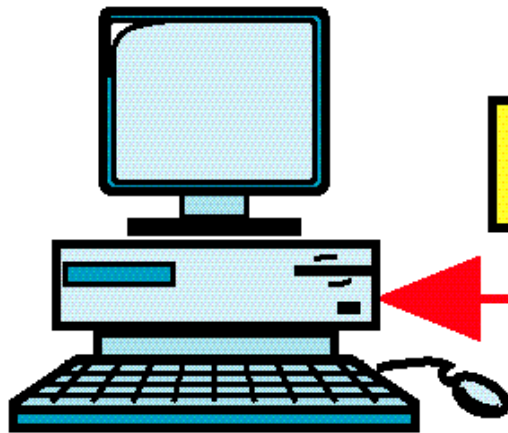


Positive response to poll



Uso del campo Poll/Final (3)

Primary



Secondary



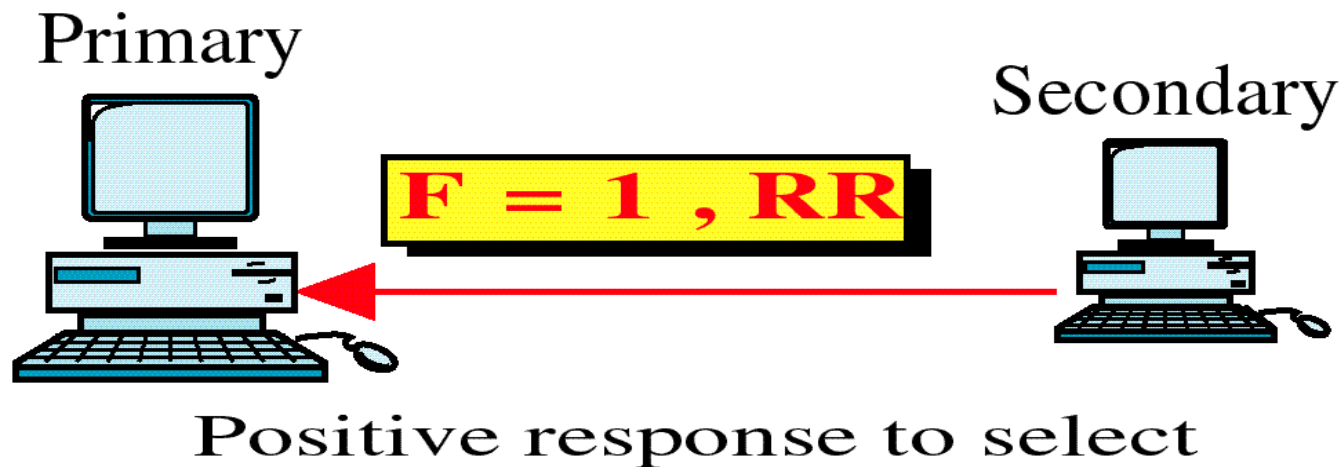
F = 1 , RR

Negative response to poll

Uso del campo Poll/Final (4)

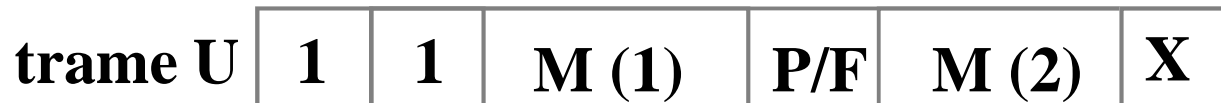


Uso del campo Poll/Final (5)





Formato trame di controllo HDLC: Trame U



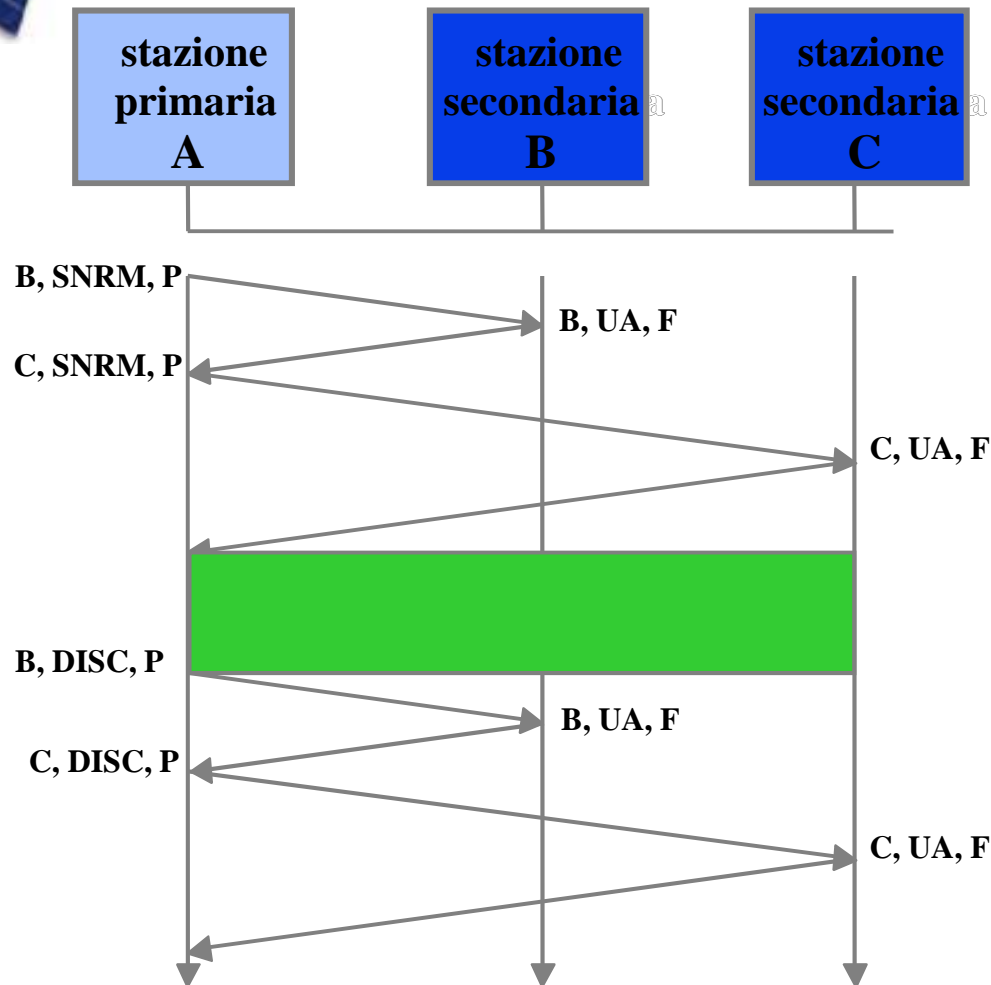
- Le trame U hanno funzioni di controllo aggiuntivo (per esempio nell'instaurazione della connessione) e vengono usate anche per la trasmissione di informazione in modalità senza connessione e senza riscontro.
- Il campo M (di 4 bit divisi in due sotto-campi M(1) e M(2)) è usato per definire fino a 32 comandi (non tutti sono definiti ed usati), mentre il bit X non è definito.



Instaurazione della connessione

- La fase di instaurazione della connessione avviene mediante lo scambio di messaggi che consentono di definire il modo di trasferimento (SNRM, SARM, SABM).
- Alla fine della fase di trasferimento dati la connessione viene chiusa mediante il comando di DISC.

Instaurazione della connessione: modalità NRM



simbologia :
(address, command, P/F bit)

il flag P/F è settato con significato di polling per sollecitare una risposta da parte della secondaria. Nella risposta della secondaria il bit è settato con significato di final ad indicare che il controllo ripassa alla stazione primaria.

Instaurazione della connessione: modalità ABM

