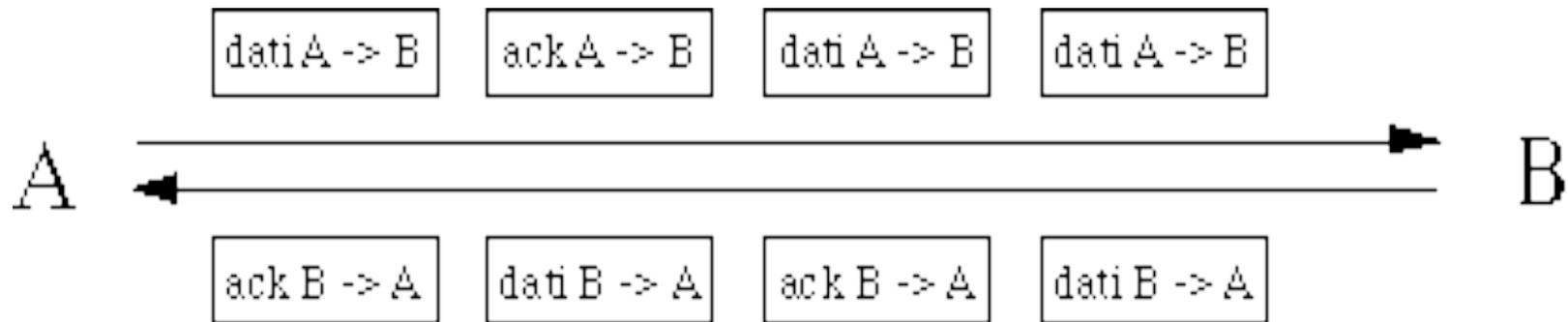


Comunicazione fra **A** e **B**:



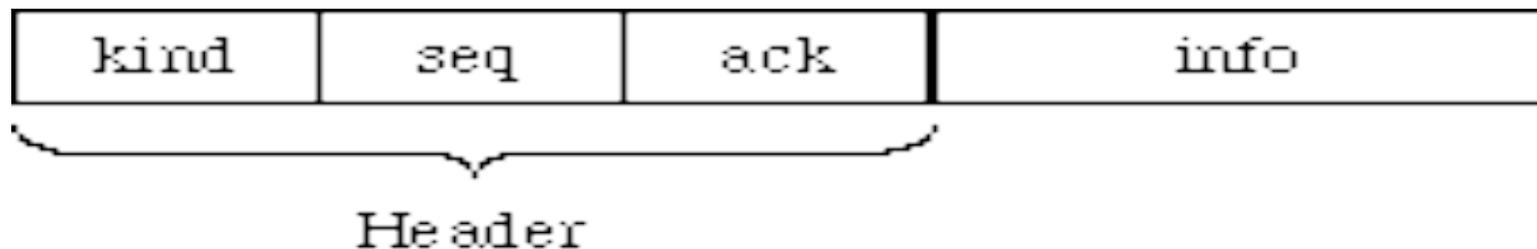
da **A** verso **B** viaggiano i frame di

- * dati inviati da **A** verso **B**

+

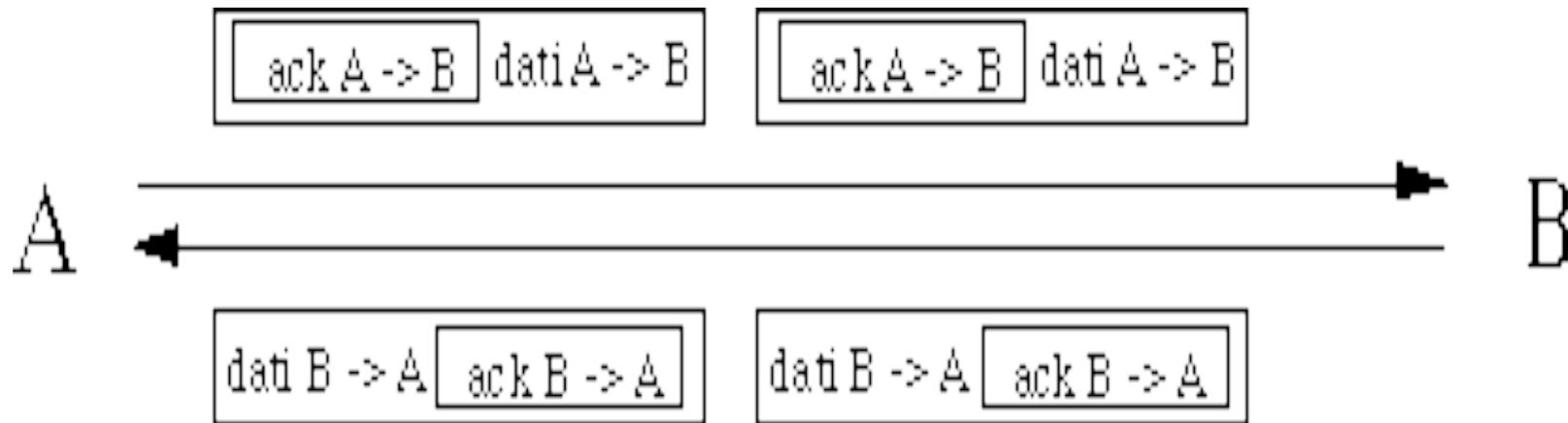
- * ack inviati da **A** in risposta ai dati ricevuti da **B**;

Viceversa da **B** verso **A**;



Il campo **kind** specifica il tipo di frame (dati o ack).

Piggybacking: quando **B** deve inviare un ack ad **A**, lo inserisce (in un apposito campo) nel prossimo frame dati che **B** deve inviare ad **A**.



Per quanto si può aspettare un frame su cui trasportare un ack?

Si stabilisce un tempo di attesa limite trascorso il quale si crea un apposito frame di ack.

Sliding window Protocolli (finestra scorrevole)

Ad ogni frame viene assegnato un numero di sequenza compreso tra 0 a $2^n - 1$ (*seq* è di n bit);

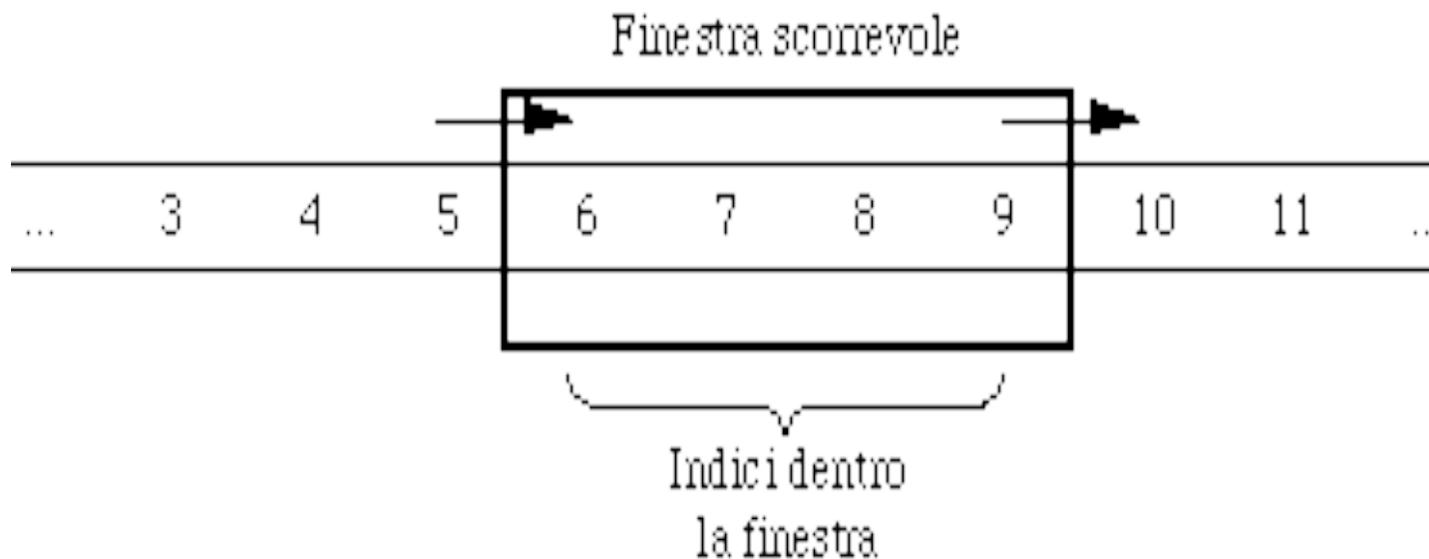
Il mittente fa scorrere una finestra sulla sequenza di indici;

gli indici in finestra corrispondono ai frame da spedire o spediti, ma non ancora confermati;

* quando arriva un ack, il corrispondente indice esce dalla finestra;

* ad ogni pacchetto in arrivo dal livello network, vengono assegnati nuovi indici;

I frame spediti ma non confermati devono essere mantenuti in memoria per la possibile ritrasmissione.



Se il buffer è pieno, il livello data link deve costringere il livello network a sospendere la consegna di pacchetti.

Il destinatario

mantiene una finestra corrispondente agli indici dei frame che possono essere accettati.

quando arriva un frame il cui indice è fuori dalla finestra viene scartato (non invia il relativo ack).

quando arriva un frame il cui indice è entro la finestra:

- * il frame viene accettato;
- * viene spedito il relativo ack;
- * la finestra viene spostata in avanti.

La finestra del destinatario rimane sempre della stessa dimensione (se è pari a 1 il destinatario accetta un solo frame e quindi i frame devono arrivare nell'ordine giusto).



Le finestre di mittente e destinatario non devono necessariamente avere uguali dimensioni, né uguali limiti inferiori o superiori.

Protocollo a finestra scorrevole di un bit: mittente e destinatario usano una finestra scorrevole di dimensione uno (*coincide con il protocollo **stop-and-wait***).

Mittente - quando invia un frame, fa partire un timer:

*se arriva l'ack con lo stesso numero di sequenza del frame che si sta cercando di trasmettere, si avanza la finestra e si passa a trasmettere il frame successivo;

*se arriva un ack diverso o scade il timer, si ritrasmette il frame;

Destinatario –

*quando arriva un frame corretto, invia comunque un ack col corrispondente numero di sequenza;

*se il frame non è un duplicato lo passa al livello network e avanza la finestra.

Se il *round-trip time* (tempo di andata e ritorno) del segnale è alto (500 ms per i canali satellitari) i protocolli stop-and-wait risultano altamente inefficienti (molto tempo sprecato in attesa degli ack).

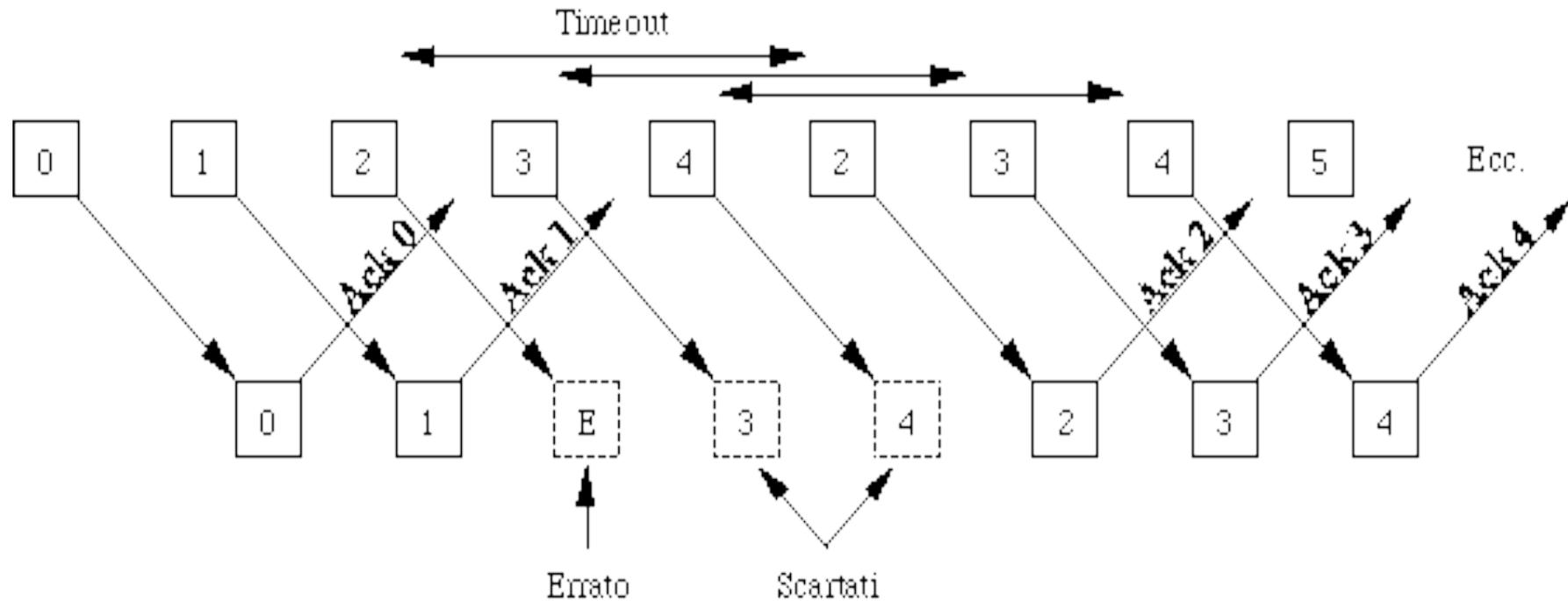


Protocolli *go-back-n* e *selective repeat*

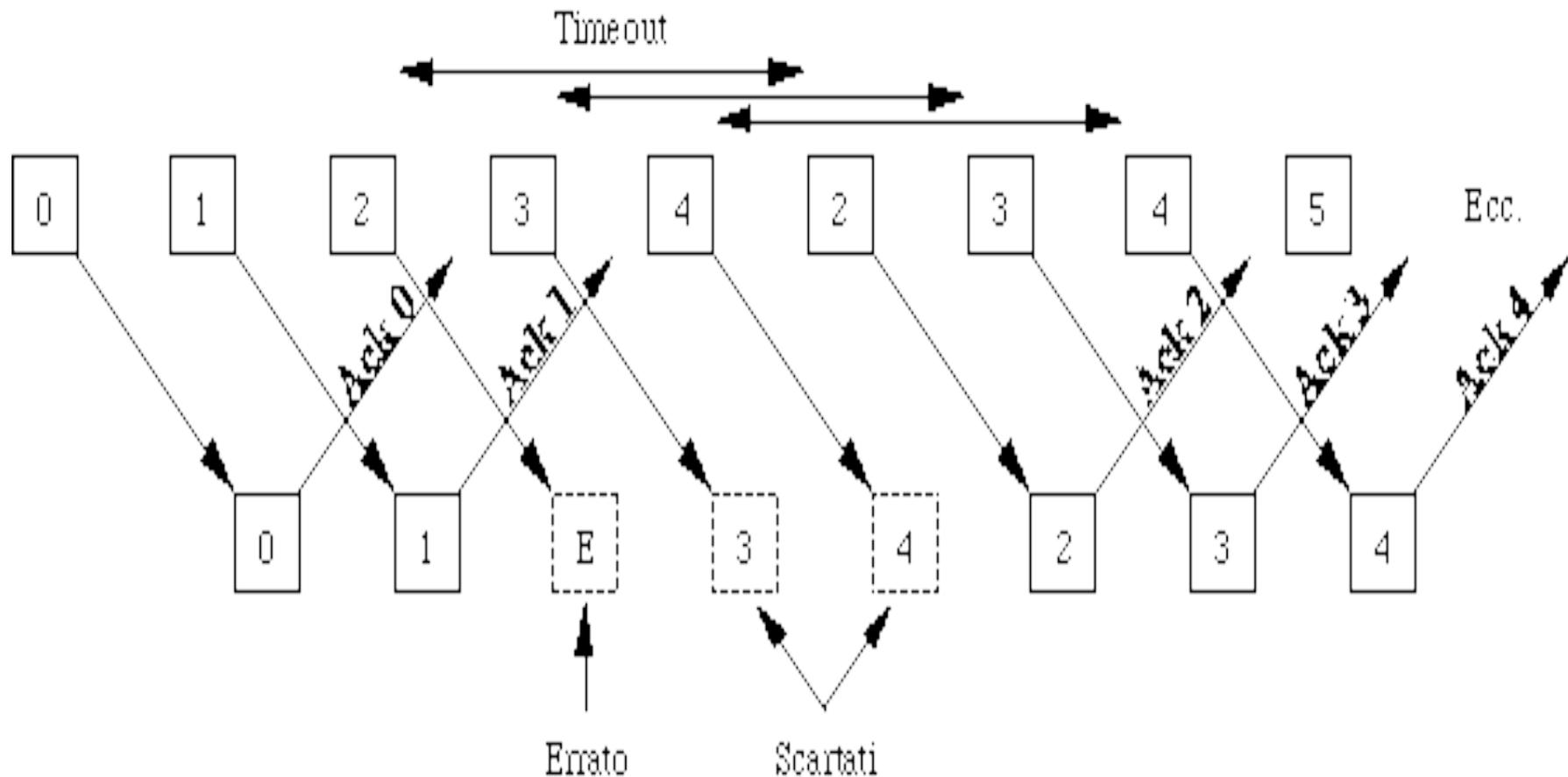
Pipelining: si inviano un certo numero di frame senza attendere l'ack dei precedenti.

Se un frame nel mezzo della sequenza subisce alterazioni (si perde o va scartato), il mittente invia molti altri frame prima di riceverne notizia.

Protocollo *go-back-n*: se arriva un frame danneggiato o con un numero di sequenza non progressivo, il destinatario ignora tale frame e tutti i successivi, non inviando i relativi ack (in pratica accetta i frame solo nell'ordine giusto);



Poiché il mittente va in time-out sul frame sbagliato, e su tutti quelli successivi, provvede a ritrasmettere la sequenza di frame a partire da quello per il quale si è verificato il time-out



Il mittente deve mantenere in un apposito buffer tutti i frame non confermati. Se il buffer si riempie, il mittente deve bloccare il livello network fino a che non si ricrea dello spazio. Purtroppo, vi è spreco di banda se il tasso d'errore è alto e/o il time-out è lungo.

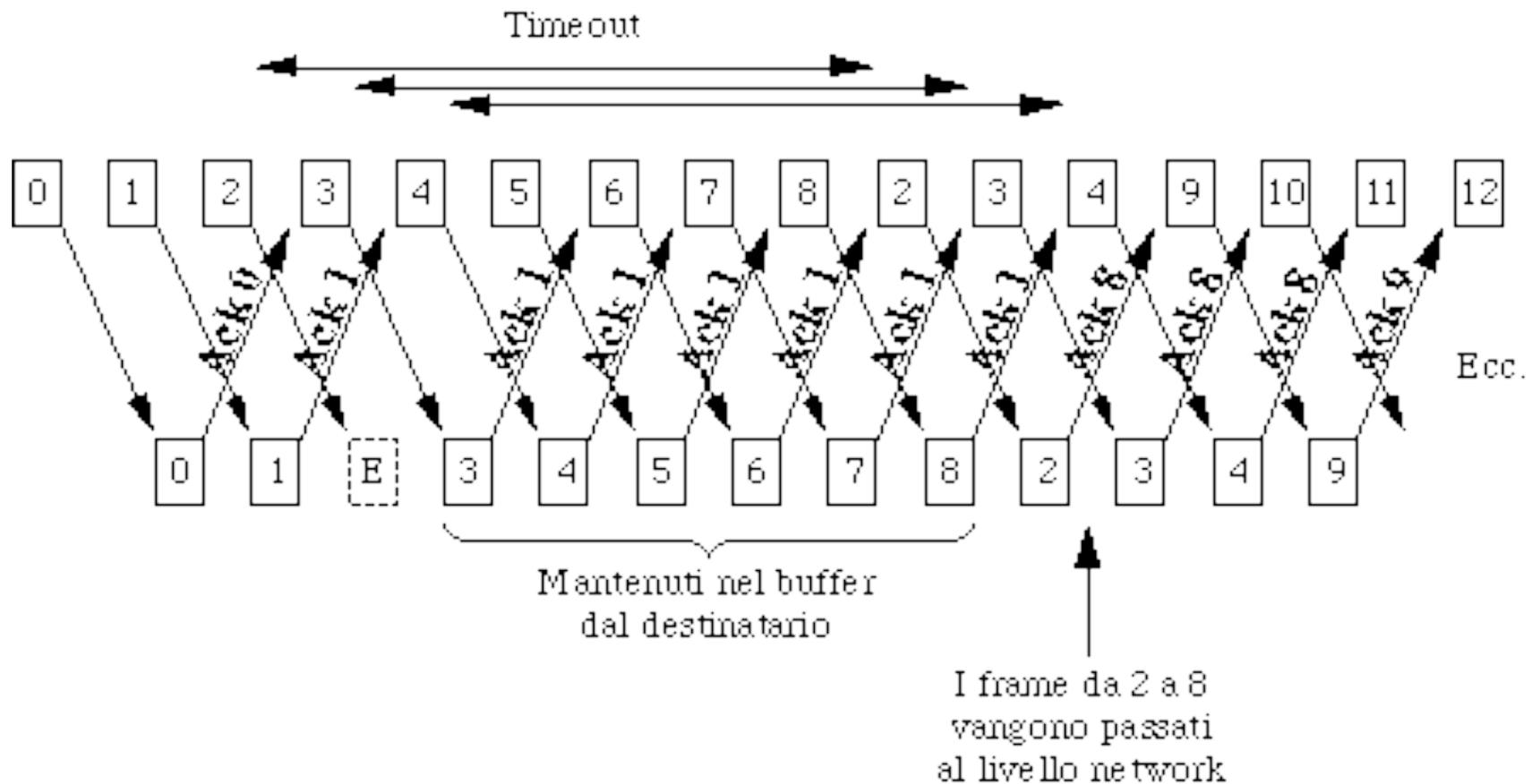
Selective repeat: *per ogni frame ricevuto correttamente, il destinatario invia un ack con il numero più alto della sequenza completa arrivata fino a quel momento.*

In caso di frame errato il *destinatario*:

- * sospende la trasmissione di ack ma continua la ricezione dei frame successivi, che mantiene nel proprio buffer.

- * appena lo riceve nuovamente (e senza errori) lo consegna, assieme a tutti i frame contigui che ha mantenuto nel buffer, al livello network;

Il *mittente*, quando si verifica un timeout, rispedisce il frame corrispondente all'indice su cui è andato in timeout.



In caso di errore il destinatario, invece di sospendere l'invio degli ack, per ogni frame ricevuto correttamente continua ad inviare l'ack col numero della sequenza completa arrivata fino a quel momento. Quando il mittente va in time-out rispedisce il frame corrispondente all'indice su cui è andato in time-out.

Alcune considerazioni:

**mittente e destinatario devono entrambi gestire un buffer:*

◦il mittente per contenere i frame non confermati;

◦il destinatario per contenere i frame successivi ad un errore;

lo spreco di banda già basso, si può ulteriormente diminuire attraverso l'uso di **NACK, da impiegare quando:*

◦arriva un frame danneggiato;

◦arriva un frame diverso da quello atteso.

* in entrambi i precedenti protocolli sono necessari timer multipli, un timer per ogni frame inviato e non confermato.

Il ricevente per inviare gli ack usa, se possibile, il piggybacking, altrimenti invia un apposito frame.

Protocolli data link

I protocolli data link più diffusi oggi:

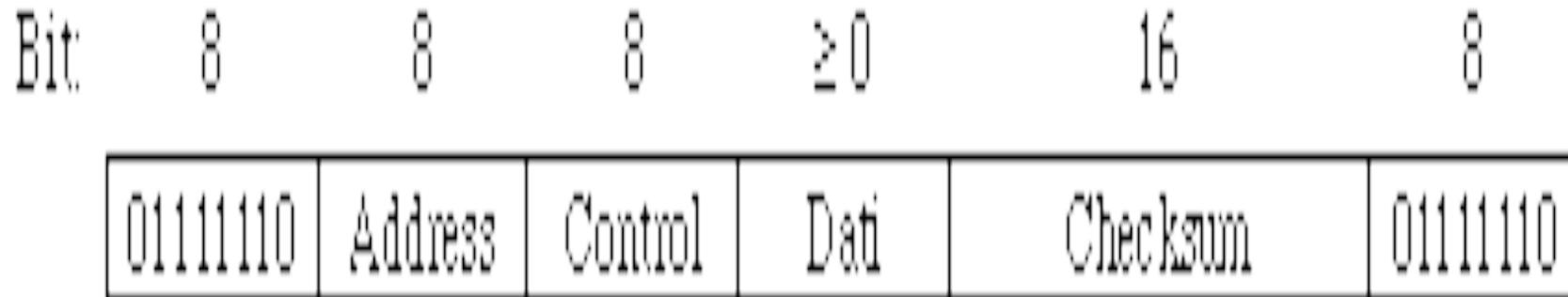
HDLC (standard ISO);

SLIP (architettura TCP/IP);

PPP (suo successore);

discendono dal protocollo *SDLC* (*Synchronous Data Link Control*), nato nell'ambito dell'architettura SNA (IBM).

HDLC (High Level Data Link Control): protocollo bit oriented, usa la tecnica del bit stuffing.



Address utilizzato nelle linee multipunto, identifica i diversi terminali (necessario per il dialogo fra un concentratore e diversi terminali)

Control contiene numeri di sequenza, ack, ecc.

Dati contiene i dati da trasportare

Checksum è calcolata con CRC-CCITT

Caratteristiche dell'HDLC:

- * usa una finestra scorrevole con numeri di sequenza a 3 bit, contenuti nel campo *Seq* del campo Control;
- * utilizza il campo *Next* del campo Control, per il piggybacking;
- * prevede tre tipi di frame (identificati dai primi due bit di Control):

Information - per la trasmissione dati;

Supervisory - per comandare diverse modalità di ritrasmissione;

Unnumbered - (manca il numero di sequenza)
per finalità di controllo o per trasportare il traffico di connessioni non affidabili.

SLIP (Serial Line IP): presentato nel 1984, è il primo protocollo, di livello data link, proposto per collegare via modem macchine Sun TCP/IP ad Internet (usa character stuffing).

Limitazioni:

- non c'è controllo degli errori;

- supporta solo IP statici;

- non è uno standard ufficiale di Internet.

Point to Point Protocol: adatto sia a connessioni telefoniche che a linee router-router, fornisce le seguenti funzionalità:

- *framing;
 - *rilevamento degli errori;
 - *protocollo di controllo per attivare, testare e disattivare la linea (***LCP Link Control Protocol***);
 - *supporto di molteplici protocolli di livello network;
 - *una famiglia di protocolli per negoziare opzioni di livello network (***Network Control Protocol***)
- Ad esempio per ogni livello network supportato c'è un differente NCP (nel caso di IP, l'NCP viene usato per negoziare un IP dinamico);

Point to Point Protocol

Il traffico prodotto (nelle fasi iniziali e finali) dai protocolli LCP e NCP viene trasportato dentro i frame PPP.

Il protocollo è modellato su HDLC, ma con alcune differenze:

- *essendo character-oriented utilizza il character stuffing (quindi i frame sono costituiti da un numero intero di byte);

- *c'è un campo apposito per il supporto multiprotocollo offerto al livello network.

Byte:	1	1	1	1	Variabile	2 oppure 4	1
	Flag	Address	Control	Protocol	Dati	Checksum	Flag
	01111110	11111111	00000011				01111110

Flag: come in HDLC

Address: sempre 11111111: di fatto non ci sono indirizzi, in quanto non c'è più l'idea di gestire linee multipunto

Control: il default (00000011) indica un unnumbered frame, quindi relativo ad un servizio non affidabile

Protocol: indica il protocollo relativo al pacchetto che si trova nel payload (LCP, NCP, IP, IPX, Appletalk, ecc.)

Payload: è di lunghezza variabile e negoziabile, il default è 1500 byte

Checksum: normalmente è di due byte