

ARQ (Automatic Repeat reQuest)

controllo congiunto di

- errore
- flusso
- sequenzasu una connessione



Controllo di flusso

- Obiettivo:
 - o 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
 - olivello di linea (2)
 - o livello di trasporto (4)



Tecniche ARQ

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

Culture

Protocollo stop and wait (1)

il trasmettitore

- 1. invia una PDU
- 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

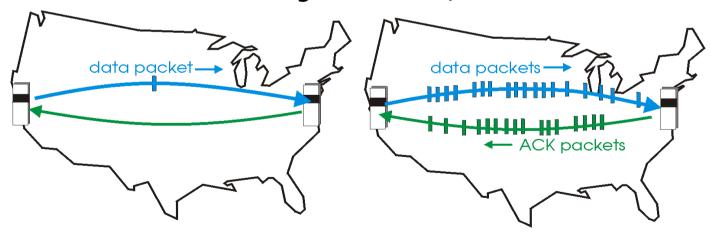
La tecnica è detta "piggybacking"



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)
- o il range di numeri di sequenza aumenta
- o necessità di buffering al sender e/o al receiver

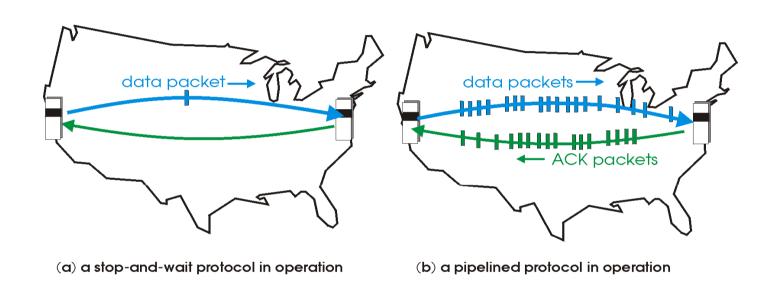


(a) a stop-and-wait protocol in operation

(b) a pipelined protocol in operation



Protocolli Pipeline



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

http://culture.deis.unical.it



Metodi di trattare gli errori nel pipelining: selective repeat

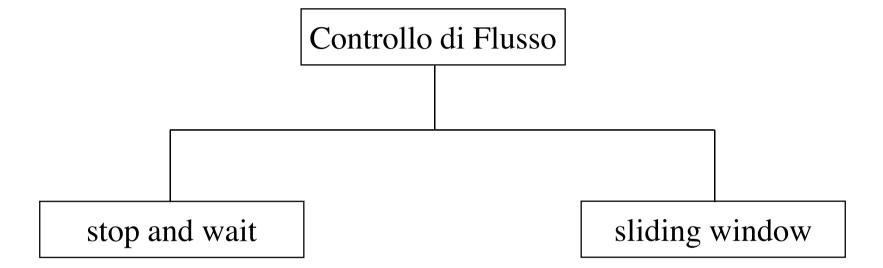
- Consiste nel fare in modo che il livello data link del ricevente memorizzi tutti i pacchetti corretti che seguono quello danneggiato.
- □ In questo caso il mittente dovrà ritrasmettere solo quel pacchetto.



Controllo di flusso



Controllo di flusso

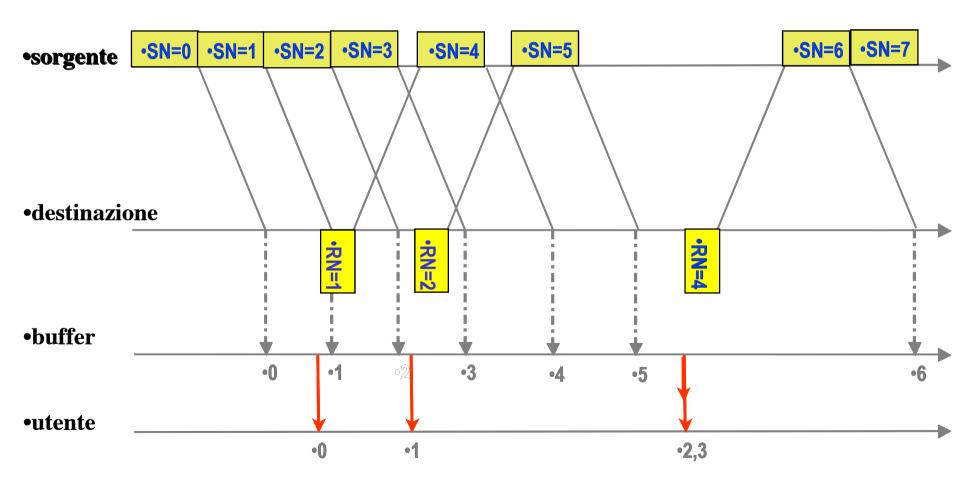




Protocolli sliding window (1)

- 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)

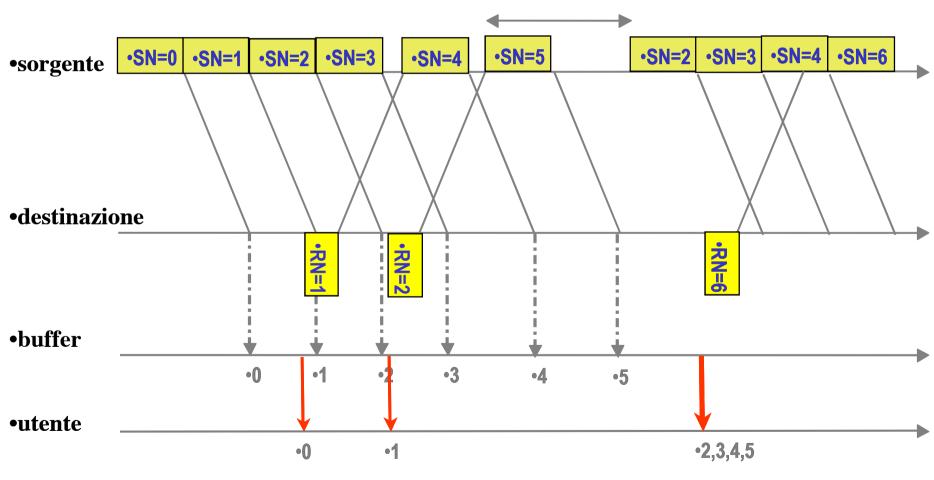




Ritrasmissioni

- 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)



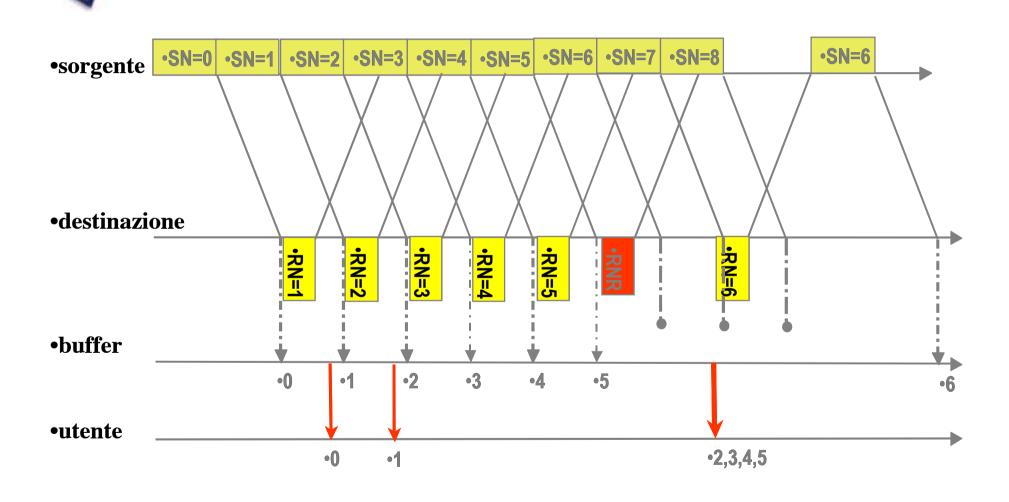


Uso del messaggio RNR

- 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:
 - oritardare l'invio dei riscontri per un tempo massimo e poi usare il RNR



Uso del messaggio RNR: esempio



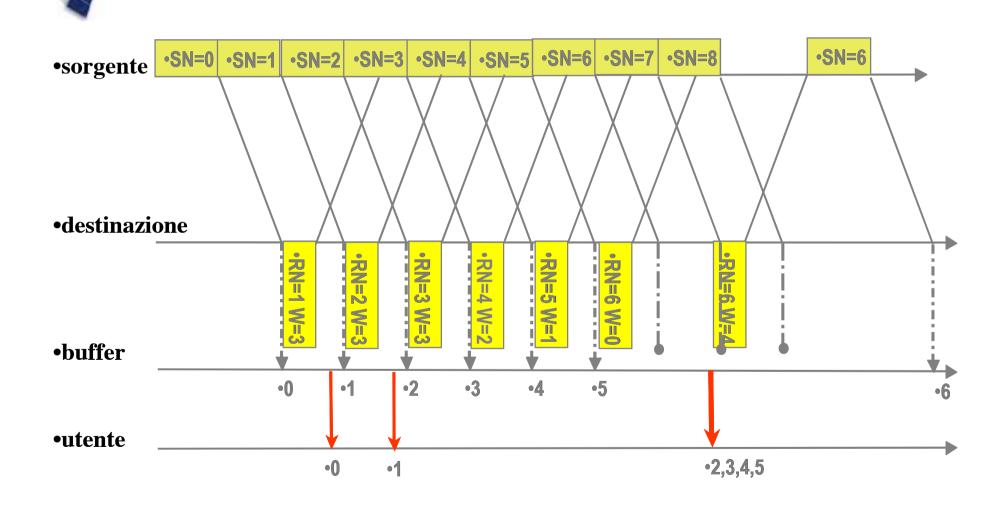


Uso del campo W

- 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
 - onon è necessario che il ricevitore dica la "verità" sullo spazio restante R
 - o 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)
 - o può usare dei meccanismi adattativi