Elaborato - Traccia 2

Simulazione Go-Back-N ARQ

Alice Beccati

19/05/2025

Obiettivo del progetto

L'obiettivo della simulazione è implementare il protocollo Go-Back-N ARQ in un ambiente non affidabile utilizzando socket UDP, implementato in linguaggio Python. La simulazione riproduce il comportamento di un client che trasmette un messaggio diviso in pacchetti numerati a un server, che deve rispondere con ACK solo per i pacchetti ricevuti in ordine, gestendo una finestra scorrevole ed eventuali perdite di pacchetti.

Descrizione del protocollo Go-Back-N

Go-Back-N è un protocollo di tipo ARQ (Automatic Repeat reQuest) utilizzato per garantire l'affidabilità nella trasmissione dei dati su canali non affidabili. Il protocollo prevede che il mittente possa inviare più pacchetti consecutivi (fino a un massimo definito dalla dimensione della finestra) senza attendere una conferma per ciascuno. Ogni pacchetto è identificato da un numero di sequenza (sequence number).

Il ricevente, invece, accetta soltanto i pacchetti che arrivano corretti e nell'ordine previsto. Se riceve un pacchetto con un numero di sequenza diverso da quello atteso, lo scarta e invia nuovamente un ACK per l'ultimo pacchetto ricevuto correttamente. Questo comportamento è detto "cumulative ACK".

In caso di perdita di un pacchetto il mittente ritrasmette tutti i pacchetti a partire da quello non ancora confermato. Questo meccanismo permette di garantire che tutti i dati arrivino al destinatario nell'ordine corretto, anche in presenza di perdite.

Il protocollo risulta inefficiente se vengono sbagliati molti pacchetti, ma permette di fare in modo che l'unico a tenere i pacchetti in memoria sia il mittente.

Architettura dell'implementazione

Tecnologia: socket UDP con indirizzamento locale (localhost).

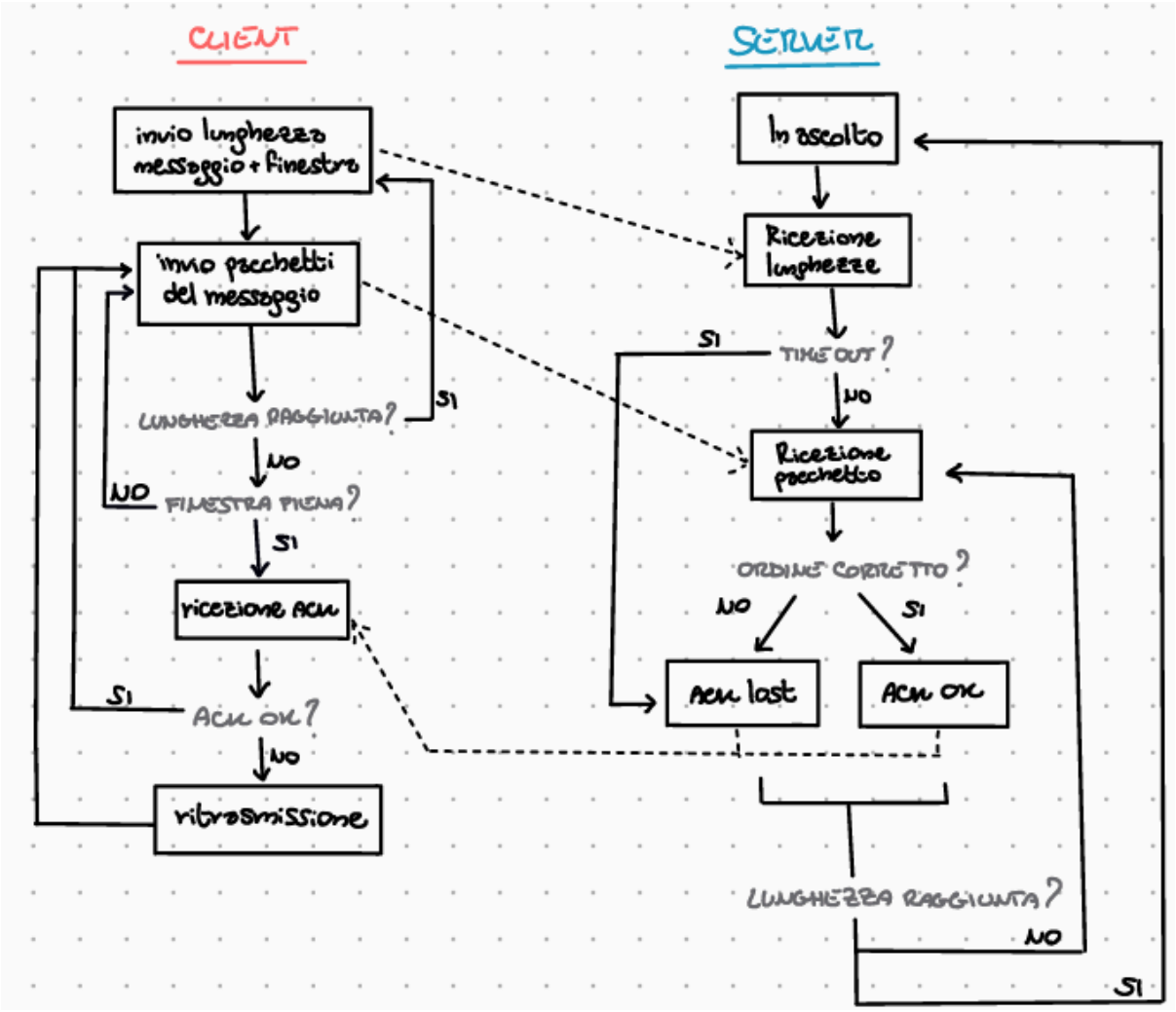
- Formato pacchetto: "<sequence_number>:<dato>".
- Formato ACK e timeout: "<ack_number>:ack" o "<ack_number>:time".

Funzionamento del client

- Inserisce un messaggio da inviare e lo converte in bit.
- Invia la lunghezza del messaggio al server.
- Inserisce la lunghezza della finestra e la invia al server.
- Invia i pacchetti secondo la struttura definita sopra, scomponendo ogni bit del messaggio e li inserisce man mano nella finestra Go-Back-N.
- Simula perdita di pacchetti con una probabilità (es. 1%).
- Riceve ACK: se corretto avanza la finestra; se errato, ritrasmette da *windMinIndex* (l'inizio della finestra).
- Vengono calcolati dati statistici: numero pacchetti ritrasmessi, pacchetti persi e ack corretti ricevuti.

Funzionamento del server

- Riceve la lunghezza iniziale del messaggio e della finestra.
- Viene settato un tempo di timeout in funzione della dimensione della finestra.
- Aspetta i pacchetti uno alla volta.
- Verifica se il pacchetto è nell'ordine atteso (*expectedN*).
  - Se sì: invia un ACK e aggiorna il prossimo pacchetto che si aspetta di ricevere.
  - Se no: invia ACK il cui sequence number è quello dell'ultimo pacchetto corretto.
- In caso di timeout: viene restituito il sequence number come nel caso del pacchetto perso, con la differenza che viene specificato il motivo dell'errata ricezione tramite l'aggiunta della stringa "time" nella risposta per il client.



Gestione degli errori e dei timeout

La simulazione della perdita di un pacchetto è impostata tramite una probabilità percentuale di successo/insuccesso. In caso di fallimento il pacchetto non viene inviato.

A lato server, gli errori vengono gestiti tramite il controllo del numero di sequenza: se un pacchetto arriva fuori ordine rispetto a quello atteso, viene scartato e il server risponde con un ACK cumulativo riferito all'ultimo pacchetto ricevuto correttamente.

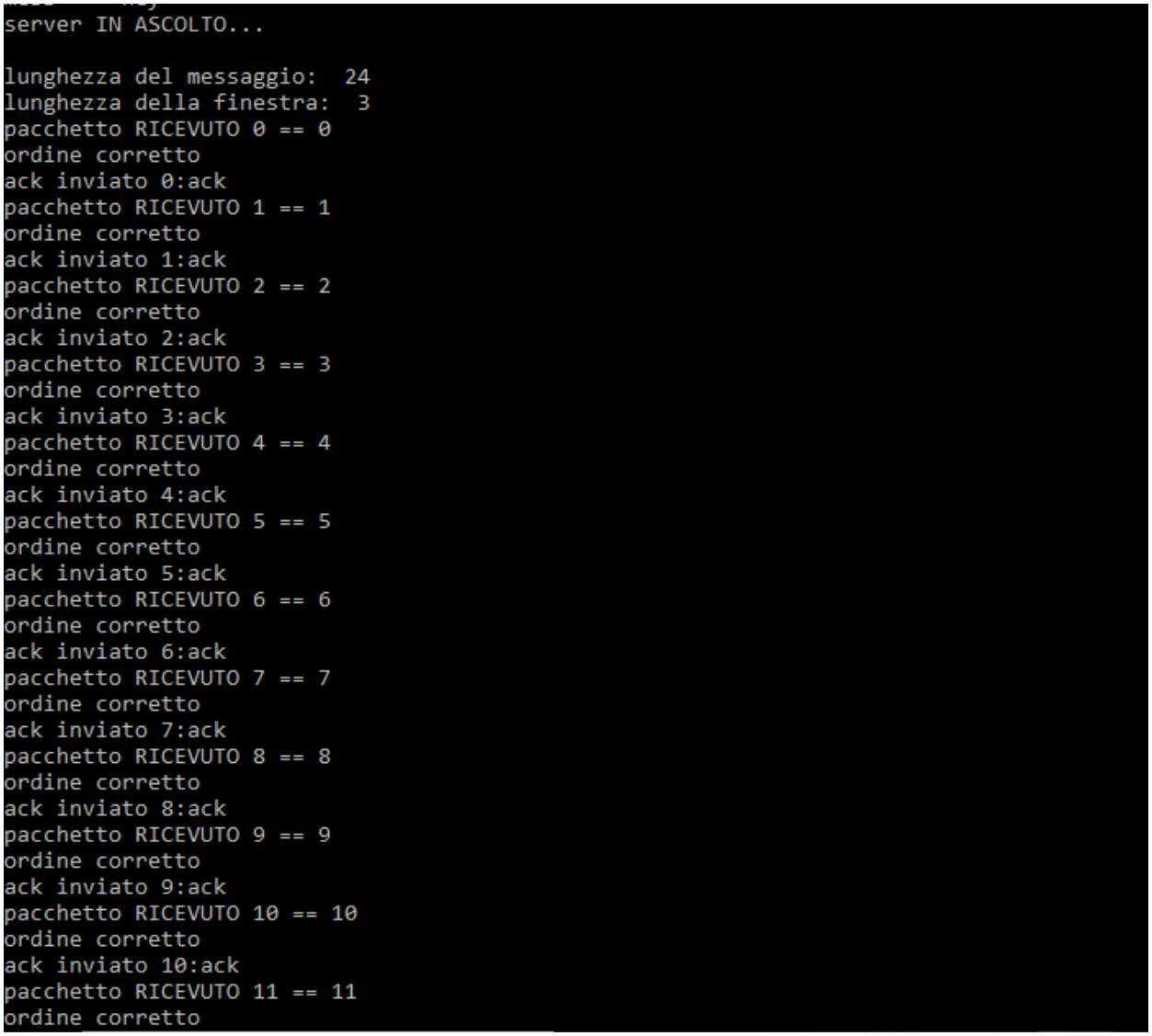
Mentre, nel caso un pacchetto ritardi troppo ad arrivare, si gestisce la perdita dell'informazione tramite un timeout. Il quale "scatta" dopo un tempo calcolato in funzione della dimensione della finestra.

Conclusioni

La simulazione dimostra l'efficacia del protocollo Go-Back-N nel garantire l'affidabilità sopra UDP. Il comportamento del client e del server sono conformei alle specifiche del protocollo. Il progetto ha consolidato la comprensione delle meccaniche di trasmissione affidabile su rete non affidabile.

Log della console

- ➔ Server



```
pacchetto RICEVUTO 20 == 21
ordine errato, pacchetto perso
ack inviato 20:ack
pacchetto RICEVUTO 18 == 21
ordine errato, pacchetto perso
ack inviato 20:ack
pacchetto RICEVUTO 19 == 21
ordine errato, pacchetto perso
ack inviato 20:ack
pacchetto RICEVUTO 20 == 21
ordine errato, pacchetto perso
ack inviato 20:ack
pacchetto RICEVUTO 18 == 21
ordine errato, pacchetto perso
ack inviato 20:ack
pacchetto RICEVUTO 19 == 21
ordine errato, pacchetto perso
ack inviato 20:ack
pacchetto RICEVUTO 20 == 21
ordine errato, pacchetto perso
ack inviato 20:ack
pacchetto RICEVUTO 18 == 21
ordine errato, pacchetto perso
ack inviato 20:ack
pacchetto RICEVUTO 19 == 21
ordine errato, pacchetto perso
ack inviato 20:ack
pacchetto RICEVUTO 20 == 21
ordine errato, pacchetto perso
ack inviato 20:ack
pacchetto RICEVUTO 21 == 21
ordine corretto
ack inviato 21:ack
pacchetto RICEVUTO 22 == 22
ordine corretto
ack inviato 22:ack
pacchetto RICEVUTO 23 == 23
ordine corretto
ack inviato 23:ack
mess ==  hey
server IN ASCOLTO...
```

➔ Client

```
MESSAGGIO : hey
sending "hey"
011010000110010101111001
Inserisci la dimensione della finestra -> 3
pacchetto  0 inviato :  0
pacchetto  1 inviato :  1
pacchetto  2 inviato :  1
ack ricevuto:  0
descrizione:  ack
ack OK
pacchetto  3 inviato :  0
ack ricevuto:  1
descrizione:  ack
ack OK
pacchetto  4 inviato :  1
ack ricevuto:  2
descrizione:  ack
ack OK
pacchetto  5 inviato :  0
ack ricevuto:  3
descrizione:  ack
ack OK
pacchetto  6 inviato :  0
ack ricevuto:  4
descrizione:  ack
ack OK
pacchetto  7 inviato :  0
ack ricevuto:  5
descrizione:  ack
ack OK
pacchetto  8 inviato :  0
ack ricevuto:  6
descrizione:  ack
ack OK
pacchetto  9 inviato :  1
ack ricevuto:  7
```

```
ack ricevuto:  20
descrizione:  ack
ack ERRORE
ritrasmissione pacchetto...
pacchetto  21 inviato :  0
pacchetto  22 inviato :  0
pacchetto  23 inviato :  1
ack ricevuto:  20
descrizione:  ack
ack ERRORE
ritrasmissione pacchetto...
pacchetto  21 inviato :  0
pacchetto  22 inviato :  0
pacchetto  23 inviato :  1
ack ricevuto:  20
descrizione:  ack
ack ERRORE
ritrasmissione pacchetto...
pacchetto  21 inviato :  0
pacchetto  22 inviato :  0
pacchetto  23 inviato :  1
ack ricevuto:  20
descrizione:  ack
ack ERRORE
ritrasmissione pacchetto...
pacchetto  21 inviato :  0
pacchetto  22 inviato :  0
pacchetto  23 inviato :  1
ack ricevuto:  21
descrizione:  ack
ack OK
ack ricevuto:  22
descrizione:  ack
ack OK
ack ricevuto:  23
descrizione:  ack
ack OK
Su  24  pacchetti :
-  123  ritrasmessi
-  24  ack corretti ricevuti
-  2  persi
MESSAGGIO :
```