

Relazione progetto reti

Martina Ravaioli

June 2025

Contents

1	Descrizione del Sistema	2
2	Funzionamento Client	2
3	Funzionamento Server	2
4	Caratteristiche del Protocollo	2
5	Caratteristiche implementative	3
6	Diagrammi di flusso	4

1 Descrizione del Sistema

Il sistema, come da consegna, simula il protocollo go-back-n tra un client e un server usando socket UDP. Il sistema ha un ratio di simulazione di lost packages del 20 per cento. In caso di pacchetti persi senza duplicate ack li reinvia una volta scaduto il timeout di 0.5 secondi. Il progetto è composto da:

- Client: parte che manda i pacchetti
- Server: parte che riceve i pacchetti e risponde con i relativi ACK

2 Funzionamento Client

il client si occupa di mandare i messaggi, inizialmente li manda in burst fino al riempimento della Window ovvero il limite di pacchetti che può mandare prima di ricevere almeno un acknowledgement dal server. Una volta raggiunto il limite si mette in attesa e ad ogni ack ricevuto manda un nuovo pacchetto. tutti i pacchetti che manda sono in ordine crescente da 0, l'unica occasione in cui il client manda un pacchetto con numero fuori dall'ordine è quella in cui l'ack ricevuto dal server sia un ack duplicato che sta ad indicare l'ultimo pacchetto ricevuto senza errori.

3 Funzionamento Server

il server viene attivato e poi resta in attesa di un collegamento da parte del client. Una volta arrivato il messaggio controlla il numero e se è il numero che si aspettava di ricevere, il numero che rispetta l'ordine in cui vuole i pacchetti, allora invia il messaggio di acknowledgement con lo stesso numero. Se il numero non è quello che si aspettava allora manda il numero dell'ultimo pacchetto ricevuto correttamente per fare in modo che il client reinvii i messaggi successivi a quello.

4 Caratteristiche del Protocollo

Il protocollo go back n ARQ gestisce la mancata ricezione di un pacchetto. In particolare oltre a gestire i pacchetti non ricevuti, gestisce l'ordine con cui arrivano. Se non arrivano nell'ordine giusto sono da considerarsi persi e di conseguenza il server risponderà ad un pacchetto fuori ordine con un ack duplicato che rappresenta l'ultimo ricevuto correttamente. Questo protocollo per quanto perda molto in fattore temporale dovendo reinviare anche pacchetti che non lo necessitavano guadagna altrettante in memoria non dovendo, in caso di pacchetti persi, ricordare tutti i pacchetti inviati successivamente fino al corretto invio di quello perso.

5 Caratteristiche implementative

La simulazione dei pacchetti persi è gestita casualmente tramite la libreria random. Il timeout viene usato per la gestione dei pacchetti persi senza un duplicate ack. Gli ack duplicati vengono ignorati dal client così che possano essere gestiti attraverso il timeout. I pacchetti fuori ordine vengono scartati dal server come se non fossero stati inviati.

una volta terminata la trasmissione entrambi mandano a schermo le proprie statistiche quali: Lato Server:

- quanti pacchetti ha ricevuto
- quanti pacchetti fuori ordine ha ricevuto
- quanti pacchetti di acknowledgement ha inviato

Lato Client:

- quanti pacchetti ha inviato
- quanti pacchetti ha ritrasmesso
- quanti pacchetti di acknowledgement ha ricevuto
- quanti pacchetti persi ha simulato

6 Diagrammi di flusso

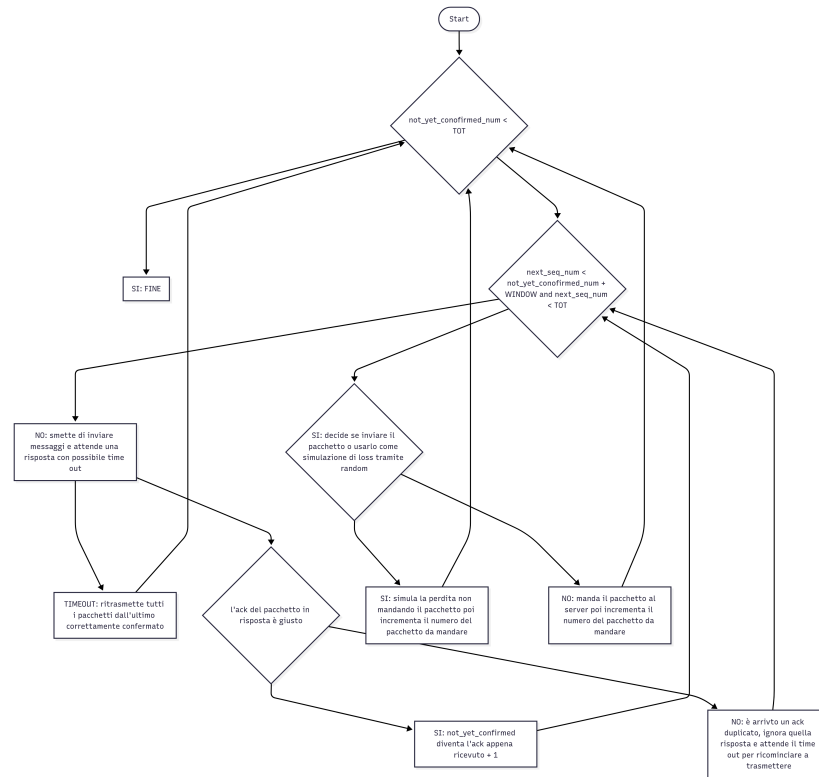


Figure 1: Diagramma di flusso del funzionamento del Client

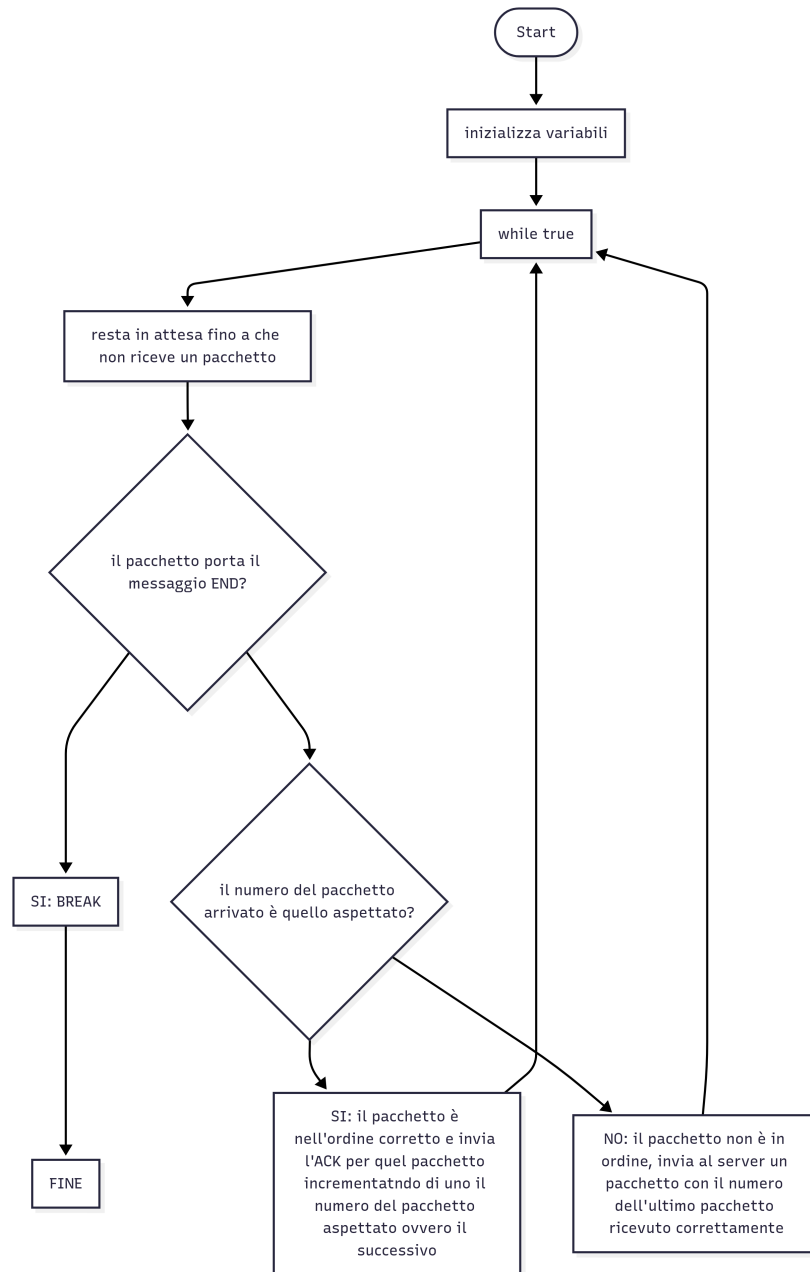


Figure 2: Diagramma di flusso del funzionamento del Server