

Progetto Incapsulamento

Emanuele Capasso 4BI

February 15, 2025

1 Introduzione

Il programma, nella prima parte, si occupa di leggere un messaggio da un file di testo (`Messaggio.txt`), convertirlo in binario, aggiungergli gli header dei livelli rete, collegamento e fisico della struttura ISO/OSI e infine scrivere il frame completo in un altro file di testo (`Frame.txt`).

Successivamente il frame verrà decapsulato completamente e verrà mostrato il contenuto del messaggio inizialmente inviato.

1.1 Programma di incapsulamento

Il programma è suddiviso in tre file: **Main.cpp**, **Funzioni.cpp** e **Funzioni.h**. Il file **Main.cpp** contiene il codice che permette l'esecuzione del tutto, **Funzioni.cpp** implementa le funzioni che eseguono le varie istruzioni e **Funzioni.h** contiene i prototipi delle funzioni e le definizioni delle struct degli header dei diversi livelli, contenenti i propri campi.

Il programma inizia leggendo un messaggio dal file **Messaggio.txt** utilizzando la funzione **leggiDaFile()**.

Una volta letto, il messaggio, grazie alla funzione **convertiInBinario()**, viene convertito in binario, a 8 bit.

Successivamente, viene creato l'header del livello rete utilizzando la struttura `IpHeader`. La funzione **creaDatagram()** imposta i vari campi dell'header.

L'header del livello collegamento viene creato nella funzione **creaFrame()**. L'header IP, insieme anche al messaggio iniziale, viene incapsulato all'interno del frame.

Infine, il frame completo viene scritto, in binario, in un file di testo chiamato **Frame.txt** utilizzando la funzione **scriviSuFile()**.

1.2 Programma di decapsulamento

Una volta che il frame completo è stato scritto nel file **Frame.txt**, è stato scritto un altro programma per il processo inverso a quello affrontato nella sezione precedente, ossia il decapsulamento. Il protagonista di questa istruzione è il file **Decapsulamento.py**, scritto in linguaggio Python e non C++.

La funzione **leggiDaFile()** si occupa di leggere il contenuto del file **Frame.txt**. Una volta letto il frame, il programma decapsula il livello Ethernet, utilizzando la funzione **decapsulaEthernet()**, suddividendo il frame nei vari campi del suo header vari campi. Il campo **data** viene separato dal resto del frame, in quanto contenente sia l'header IP sia il messaggio finale.

Successivamente, tramite l'uso della funzione **decapsulaIP()**, il programma decapsula il datagramma, estraendo i vari campi restituendo solamente il campo **data**, che contiene il messaggio vero e proprio inviato dal mittente.

Una volta estratto il messaggio, che è rappresentato come una stringa binaria, il programma lo converte in stringa, ripristinando il suo aspetto originale e mostrando successivamente il suo contenuto.