

Implementazione del Frame Ethernet e IP

Croccolo, Bellini, Bacci

5/02/2025

1 Introduzione

Questa relazione descrive il codice che implementa la generazione e la decapsulazione di un frame Ethernet contenente un datagramma IP. Il codice è suddiviso tra C++ e Python per la creazione e l'analisi del frame.

2 Descrizione Generale del Codice

Il codice fornisce un'implementazione del processo di incapsulamento e decapsulamento di un messaggio all'interno di un frame Ethernet contenente un datagramma IP.

Il codice C++ si occupa di:

- Creare l'header IP con i campi necessari per l'instradamento del pacchetto.
- Costruire un frame Ethernet incorporando l'header IP.
- Convertire il messaggio in binario e inserirlo nel pacchetto.
- Calcolare il checksum dell'header IP per verificare l'integrità dei dati.
- Salvare il frame risultante in un file.

Il codice Python invece si occupa della decapsulazione:

- Legge il frame Ethernet dal file.
- Estrae l'header IP e successivamente il messaggio contenuto nel pacchetto.
- Converte il messaggio dal formato binario al testo leggibile.

3 Header IP e Ethernet

L'header IP è un componente fondamentale del pacchetto e contiene vari campi, tra cui:

- **Versione:** Specifica la versione del protocollo IP utilizzata.
- **Lunghezza Header:** Indica la dimensione dell'header IP.
- **Tipo di Servizio (TOS):** Determina la priorità del pacchetto.

- **Lunghezza Totale:** Specifica la lunghezza totale del datagramma IP.
- **Identificazione:** Usata per identificare frammenti di un pacchetto.
- **Flags e Offset di Frammentazione:** Necessari per la gestione della frammentazione.
- **Time to Live (TTL):** Indica il numero massimo di router attraversabili.
- **Protocollo:** Definisce il protocollo di trasporto (es. TCP o UDP).
- **Checksum Header:** Campo di verifica dell'integrità dell'header.
- **Indirizzi IP sorgente e destinazione:** Specificano i nodi di origine e destinazione.
- **Dati:** Contiene il messaggio trasmesso.

L'header Ethernet include:

- **Preambolo:** Sequenza di sincronizzazione per la trasmissione.
- **SFD (Start Frame Delimiter):** Indica l'inizio del frame.
- **Indirizzo MAC Destinazione:** Identifica il destinatario del frame.
- **Indirizzo MAC Sorgente:** Identifica il mittente del frame.
- **Tipo:** Specifica il tipo di protocollo trasportato (es. IPv4).
- **Dati:** Contiene l'header IP e il messaggio.
- **FCS (Frame Check Sequence):** Usato per la verifica degli errori.

4 Processo di Incapsulamento

L'incapsulamento del messaggio avviene nel codice C++ attraverso i seguenti passaggi:

1. Lettura del messaggio da un file di testo.
2. Conversione del messaggio in binario.
3. Creazione dell'header IP e calcolo del checksum.
4. Creazione del frame Ethernet e inserimento dell'header IP.
5. Salvataggio del frame generato in un file.

5 Processo di Decapsulazione

Nel codice Python, la decapsulazione avviene nel seguente modo:

1. Lettura del frame Ethernet dal file.
2. Estrazione dell'header IP dal frame.
3. Recupero dei dati contenuti nell'header IP.
4. Conversione dei dati binari in testo leggibile.

6 Conclusione

Questa implementazione dimostra il processo di incapsulamento e decapsulamento dei dati nei protocolli Ethernet e IP. Il codice illustra come le informazioni vengono strutturate a livello di rete e livello di collegamento, evidenziando il ruolo chiave di questi protocolli nella comunicazione tra dispositivi.