Lo Stack TCP/IP

Lo stack TCP/IP è composto da quattro livelli. È stato sviluppato prima della definizione del modello OSI, che avrebbe dovuto sostituirlo, ma con il tempo si è imposto come lo standard de facto, specialmente su Internet.

I due protocolli principali della suite TCP/IP sono IP (Internet Protocol), che opera al livello rete, e TCP (Transmission Control Protocol), che appartiene al livello trasporto. Quest'ultimo è responsabile della comunicazione end-to-end tra processi che si trovano su host differenti.

Servizi del Livello Trasporto

Il livello di trasporto consente a due processi su host diversi di comunicare direttamente. Questo livello offre un controllo completo end-to-end. Un protocollo di trasporto può gestire più connessioni simultanee verso lo stesso host, distinguendo a quale processo inviare ciascun messaggio grazie a meccanismi basati su porte e socket.

Le Porte e le Socket

Poiché su un host possono essere attivi più processi che generano richieste, è necessario un sistema per identificare correttamente il destinatario di ogni pacchetto. Per questo vengono utilizzate le porte, numeri a 16 bit assegnati dall'IANA.

Le porte sono suddivise in tre categorie:

- Well Known Ports: da 0 a 1023
- Registered Ports: da 1024 a 49151
- Dynamic/Private Ports: da 49152 a 65535

Ogni connessione è identificata da una quintupla formata da: protocollo, indirizzo IP del client, indirizzo IP del server, numero di porta del client, numero di porta del server. Questa struttura è detta association. La parte locale dell'associazione (protocollo, IP e porta) è chiamata socket. Le socket permettono di distinguere le comunicazioni tra diversi processi su uno stesso host.

Multiplexing e Demultiplexing

Tutti i protocolli del livello trasporto forniscono funzionalità di multiplexing e demultiplexing. Il multiplexing consiste nel raccogliere dati da diverse applicazioni, aggiungere un header e inviarli al livello rete. Il demultiplexing è il processo inverso: ricevere un segmento, esaminarne l'header e consegnare i dati al processo corretto.

I Principali Protocolli di Trasporto

I protocolli più usati in questo livello sono UDP (User Datagram Protocol) e TCP (Transmission Control Protocol). Le unità dati per questi protocolli sono rispettivamente il datagramma (TPDU per UDP) e il segmento (TPDU per TCP).

TCP

Il TCP è il protocollo più diffuso per il livello trasporto. È connection-oriented e affidabile, ed è utilizzato da applicazioni che necessitano di una trasmissione sicura, come FTP (trasferimento file), SMTP (posta elettronica) e HTTP (pagine web).

Il Segment TCP è composto da un header (20 byte) con al suo interno:

- Numero porta sorgente
- Numero porta destinazione
- Sequenza di numeri
- Acknowledgement number
- Lunghezza header
- Checksum, window size
- Urgent pointer

Poi troviamo il parametro options e data.

Instaurazione e Abbattimento Sessione TCP

Instaurazione di una Connessione TCP – Three-Way Handshake

Affinché possa essere instaurata una connessione TCP tra due host, è necessario che l'host ricevente acconsenta mediante una sequenza composta da tre fasi, nota come Three-Way Handshake:

1. Host 1 invia un segmento TCP con il flag SYN impostato a 1. Viene inoltre generato un numero di sequenza iniziale (sequence number) scelto in modo casuale, indicato con X.
2. Host 2, se accetta di stabilire la connessione, risponde con un segmento in cui:
   - il flag SYN è impostato a 1
   - il flag ACK è impostato a 1
   - l'Ack Number è posto a X + 1
   - viene generato un nuovo numero di sequenza Y
3. Host 1 risponde con un segmento contenente:
   - flag ACK impostato a 1
   - l'Ack Number impostato a Y + 1

A questo punto la connessione è considerata stabilita e si può procedere con la trasmissione dei dati.

Trasferimento Dati: Affidabilità e Controlli

Durante la fase di trasmissione:

- TCP garantisce una trasmissione affidabile e ordinata dei dati
- sono attivi meccanismi di controllo degli errori, controllo del flusso (flow control) e controllo della congestione (congestion control)
- ogni segmento ricevuto viene confermato (ACK) e i numeri di sequenza aiutano a mantenere l'ordine dei dati ed evitare duplicazioni o perdite

Chiusura della Connessione TCP – Double-Way Handshake

Poiché la connessione TCP è bidirezionale, la chiusura avviene in maniera indipendente per ciascuna direzione. La procedura di chiusura prende il nome di Double-Way Handshake:

1. Un host (es. Host 1) invia un segmento TCP con il flag FIN impostato a 1
2. Host 2 risponde con un segmento contenente:
   - flag ACK = 1
   - l'Ack Number = X + 1
3. Se Host 2 ha ancora dati da trasmettere, continua a farlo. Quando termina, invia un segmento con FIN = 1 e numero di sequenza Y
4. Infine, Host 1 conferma con ACK = 1 e Ack Number = Y + 1

Controllo della Congestione in TCP

Per individuare una congestione il TCP utilizza dei timer per misurare il tempo trascorso tra l'invio di un segmento e la ricezione del relativo ack. Se questo non arriva entro un determinato tempo si genera un timeout.

TCP ipotizza che la perdita di dati sia per una congestione della rete e agisce di conseguenza. Tuttavia, esistono versioni di TCP che tengono conto anche di errori di trasmissione.

TCP implementa una serie di algoritmi, specificati nella RFC 5681, per il controllo della congestione. Tutti fanno uso della finestra di congestione, che indica il massimo numero di byte non confermati che possono trovarsi ancora nella rete.

```
maxWindow = min(FinestraDiCongestione, FinestraDiRicezione)
```

Dove:

- FinestraDiCongestione: max numero di byte che la rete può trasmettere
- FinestraDiRicezione: quanti byte il destinatario è in grado di ricevere
- maxWindow: minimo tra i due valori, usato per decidere quanti byte trasmettere

Dei 4 algoritmi utilizzati, si analizzano:

- Slow Start: la finestra di congestione cresce esponenzialmente fino al threshold (es. 64KB)
- Congestion Avoidance: oltre il threshold, la crescita è lineare
- In caso di timeout: la soglia viene dimezzata e la finestra torna al valore iniziale

Controllo di Flusso e degli Errori

Il protocollo Sliding Window è un meccanismo usato da TCP per garantire un controllo efficiente del flusso dei dati tra mittente e destinatario.

Permette di inviare più byte consecutivi senza attendere un ACK per ognuno, entro una finestra.

La dimensione della finestra dipende dallo spazio disponibile nel buffer del ricevente. In caso di buffer pieno, la finestra si restringe; se c'è spazio, si allarga.

Ogni segmento ha un numero di sequenza, così TCP può rilevare l'ordine dei pacchetti e accorgersi di eventuali perdite, ritrasmettendo solo i dati mancanti.

UDP

UDP (User Datagram Protocol) offre solo le funzionalità di multiplexing/demultiplexing. Il servizio è connectionless e non affidabile.

Il datagram UDP si compone di:

- source port number
- destination port number
- length (o checksum coverage length per UDP-Lite)
- checksum
- data

UDP-Lite

UDP-Lite è una versione modificata di UDP che accetta pacchetti anche se parzialmente corrotti.

Permette di proteggere con il checksum solo una parte del pacchetto. Utile per applicazioni come streaming video, audio o VoIP, dove è meglio ricevere qualcosa, anche se imperfetto, piuttosto che perdere tutto.

Confronto tra UDP e TCP

TCP e UDP condividono:

- le funzionalità di multiplexing/demultiplexing
- l'uso delle porte

TCP è da preferire quando serve affidabilità e integrità dei dati. UDP è indicato quando le prestazioni sono più importanti dell'assenza di perdite.