**Capitolo: Il Livello di Trasporto nel Modello OSI**

**1. Introduzione al Livello di Trasporto**

Il **livello di trasporto** è il quarto livello del modello OSI e una parte essenziale della suite di protocolli TCP/IP. Fornisce servizi fondamentali per la comunicazione end-to-end tra processi su dispositivi remoti. La sua funzione principale è garantire che i dati siano trasferiti in modo affidabile, efficiente e strutturato tra applicazioni in esecuzione su dispositivi diversi.

**2. Funzioni Principali del Livello di Trasporto**

1. **Comunicazione End-to-End:**
   * Stabilisce una connessione logica tra processi applicativi su dispositivi remoti.
   * Utilizza indirizzi IP (per identificare gli host) e numeri di porta (per identificare i processi).
2. **Multiplexing e Demultiplexing:**
   * **Multiplexing:** Permette a più applicazioni di condividere la stessa connessione di rete, distinguendole tramite numeri di porta.
   * **Demultiplexing:** Smista i dati ricevuti verso l'applicazione appropriata.
3. **Incapsulamento e Decapsulamento:**
   * **Incapsulamento:** I dati ricevuti dal livello applicativo vengono trasformati in segmenti (TCP) o datagrammi (UDP).
   * **Decapsulamento:** I segmenti o datagrammi vengono elaborati e consegnati al livello applicativo.
4. **Controllo di Flusso:**
   * Previene il sovraccarico del ricevitore regolando la velocità di trasmissione.
   * Utilizza tecniche come la **Sliding Window**.
5. **Controllo degli Errori:**
   * Identifica e corregge errori nei dati trasmessi.
   * Garantisce che i dati vengano ricevuti nell'ordine corretto e senza duplicati.
6. **Controllo della Congestione:**
   * Regola la velocità di trasmissione per evitare il sovraccarico della rete, migliorando le prestazioni complessive.

**3. Protocolli Principali del Livello di Trasporto**

**3.1 User Datagram Protocol (UDP)**

* **Caratteristiche:**
  + Protocollo leggero, non orientato alla connessione.
  + Non garantisce affidabilità, ordine o correzione degli errori.
  + Adatto a applicazioni che richiedono velocità, come streaming, VoIP e DNS.
* **Struttura del Datagramma UDP:**

| **Campo** | **Lunghezza (bit)** | **Descrizione** |
| --- | --- | --- |
| Numero Porta Mittente | 16 | Identifica il processo mittente. |
| Numero Porta Destinatario | 16 | Identifica il processo destinatario. |
| Lunghezza | 16 | Dimensione totale del datagramma. |
| Checksum | 16 | Verifica l'integrità di dati e intestazione. |

* **Checksum UDP:**
  + Comprende:
    - Pseudo-header (indirizzi IP e protocollo).
    - Header UDP.
    - Dati del datagramma.
  + Un checksum non valido provoca lo scarto del datagramma.

**3.2 Transmission Control Protocol (TCP)**

* **Caratteristiche:**
  + Protocollo orientato alla connessione.
  + Garantisce affidabilità, ordine e assenza di duplicati.
  + Supporta la comunicazione **full-duplex**.
* **Struttura del Segmento TCP:**

| **Campo** | **Lunghezza (bit)** | **Descrizione** |
| --- | --- | --- |
| Numero Porta Sorgente | 16 | Identifica l'applicazione mittente. |
| Numero Porta Destinatario | 16 | Identifica l'applicazione destinataria. |
| Numero di Sequenza | 32 | Indica il numero del primo byte trasmesso. |
| Numero di Riscontro (ACK) | 32 | Indica il numero del prossimo byte atteso. |
| Lunghezza Intestazione | 4 | Dimensione dell'intestazione in multipli di 4 byte. |
| Flag di Controllo | 6 | Specifica lo stato della connessione (SYN, ACK). |
| Dimensione Finestra | 16 | Quantità di dati che il mittente può inviare. |
| Checksum | 16 | Verifica l'integrità del segmento. |
| Puntatore Urgente | 16 | Indica dati urgenti, se presenti. |

* **Funzionalità di TCP:**
  + Numerazione dei byte per garantire l'ordine.
  + Ritrasmissione per pacchetti persi.
  + Timer per gestire ritardi e congestioni.

**4. Meccanismi del Livello di Trasporto**

**4.1 Apertura e Chiusura della Connessione (TCP):**

1. **Apertura (Three-Way Handshake):**
   * **SYN:** Il client invia un segmento con flag SYN.
   * **SYN-ACK:** Il server risponde con un segmento SYN-ACK.
   * **ACK:** Il client conferma con un segmento ACK.
2. **Chiusura (Four-Way Handshake):**
   * **FIN:** Il mittente invia un segmento con flag FIN.
   * **ACK:** Il ricevitore risponde con un segmento ACK.
   * **FIN:** Il ricevitore invia un segmento FIN.
   * **ACK:** Il mittente risponde con un ACK finale.

**4.2 Controllo di Flusso:**

* **Sliding Window:**
  + Determina quanti dati il mittente può inviare senza attendere un riscontro (ACK).
  + Migliora l'efficienza adattando la finestra di trasmissione alle capacità del ricevitore.

**4.3 Controllo della Congestione:**

* **Slow Start:** Incrementa esponenzialmente la finestra di congestione iniziale.
* **Congestion Avoidance:** Incrementa linearmente la finestra per evitare congestione.
* **Fast Retransmit:** Ritrasmette segmenti persi dopo tre ACK duplicati.
* **Fast Recovery:** Evita di ripartire da zero dopo una congestione moderata.

**4.4 Timer in TCP:**

1. **Timer di Ritrasmissione (RTO):** Calcolato in base all'RTT.
2. **Timer di Persistenza:** Mantiene attiva la connessione in caso di finestra di ricezione nulla.
3. **Timer Keepalive:** Previene la chiusura inattesa della connessione.
4. **Timer 2MSL:** Assicura che tutti i segmenti siano ricevuti prima della chiusura definitiva.

**5. Confronto tra TCP e UDP**

| **Caratteristica** | **TCP** | **UDP** |
| --- | --- | --- |
| Affidabilità | Sì (ritrasmissioni, ACK) | No |
| Ordinamento | Sì | No |
| Connessione | Orientato alla connessione | Non orientato alla connessione |
| Overhead | Alto | Basso |
| Applicazioni Tipiche | File transfer, email | Streaming, VoIP, DNS |

**6. TCP Tahoe e TCP Reno**

* **TCP Tahoe:**
  + Riparte da **Slow Start** dopo 3 ACK duplicati.
  + Riduzione drastica della finestra di congestione.
* **TCP Reno:**
  + Introduce **Fast Recovery**, evitando Slow Start completa.
  + Più efficiente durante congestioni moderate.

**7. Considerazioni Finali**

Il livello di trasporto è fondamentale per garantire comunicazioni affidabili e ottimizzate nelle reti moderne. TCP offre un controllo rigoroso su flusso, errori e congestione, rendendolo ideale per applicazioni critiche, mentre UDP privilegia velocità e semplicità, adattandosi a esigenze come streaming e VoIP. La comprensione di questo livello è cruciale per progettare sistemi di comunicazione robusti ed efficienti.