1. Livello di Rete - Introduzione

Il livello di rete si occupa di instradare i pacchetti dalla sorgente alla destinazione attraverso percorsi potenzialmente complessi. Questo livello deve gestire sia l'instradamento (routing) che il controllo del traffico.

1.1 Algoritmi di Routing Fondamentali

Bellman-Ford (Distance Vector)
Questo algoritmo distribuito funziona attraverso lo scambio di informazioni tra router vicini:

- Ogni router mantiene una tabella delle distanze verso ogni destinazione
- Periodicamente scambia queste informazioni con i vicini
- Aggiorna le proprie tabelle quando riceve informazioni migliori
- Particolarmente adatto per routing interno (RIP)

Dijkstra (Link State)
Un approccio che calcola i percorsi ottimali basandosi su una visione completa della rete:

- Ogni router conosce la topologia completa della rete
- Calcola i percorsi migliori verso tutte le destinazioni
- Utilizzato in OSPF e altri protocolli moderni
- Più stabile e veloce nella convergenza

1.2 Tipologie di Routing

Routing Statico
Il routing statico viene configurato manualmente dall'amministratore:

- Percorsi fissi e predefiniti
- Utilizzato in reti piccole e stabili
- Richiede poca manutenzione una volta configurato
- Non si adatta ai cambiamenti di rete

Routing Dinamico
Il routing dinamico si adatta automaticamente ai cambiamenti:

- I router si scambiano informazioni sulla topologia
- Reagisce automaticamente ai guasti
- Sceglie sempre il percorso migliore disponibile
- Richiede più risorse ma offre maggiore flessibilità

1.3 Controllo del Traffico

Leaky Bucket
Un meccanismo di traffic shaping che:

- Regola il traffico come un secchio che perde
- Mantiene una velocità di uscita costante
- Limita i burst di traffico
- Ideale per traffico che richiede banda costante

Token Bucket
Un approccio più flessibile che:

- Accumula "gettoni" per la trasmissione
- Permette burst controllati
- Offre maggiore flessibilità rispetto al leaky bucket
- Adatto per traffico a burst

2. Livello di Trasporto

2.1 TCP (Transmission Control Protocol)

Stabilimento della Connessione (Three-way Handshake)

1. Client → Server: SYN (Richiesta di sincronizzazione)
2. Server → Client: SYN-ACK (Conferma e sincronizzazione)
3. Client → Server: ACK (Conferma finale)

Controllo del Flusso
TCP gestisce attivamente il flusso dei dati:

- Utilizza una finestra scorrevole dinamica
- Il ricevente può controllare la velocità di trasmissione
- Previene il sovraccarico del ricevente
- Si adatta alle condizioni della rete

Gestione della Congestione
Meccanismi per evitare il sovraccarico della rete:

- Slow Start: parte lento e accelera gradualmente
- Congestion Avoidance: evita saturazione
- Fast Recovery: gestisce perdite occasionali
- Timeout: gestisce perdite gravi

2.2 UDP (User Datagram Protocol)

Caratteristiche Principali
UDP è semplice e veloce:

- Nessuna connessione da stabilire
- Nessuna garanzia di consegna
- Header minimale (8 byte)
- Ideale per applicazioni time-sensitive

2.3 Differenze Dettagliate tra TCP e UDP

Il confronto tra TCP e UDP è fondamentale per comprendere quando utilizzare ciascun protocollo.

Affidabilità e Controllo

TCP garantisce:

- Consegna ordinata dei pacchetti attraverso numeri di sequenza
- Conferma di ricezione (acknowledgments) per ogni segmento
- Ritrasmissione automatica dei pacchetti persi
- Eliminazione dei duplicati

UDP invece:

- Non garantisce l'ordine di arrivo
- Non conferma la ricezione
- Non ritrasmette pacchetti persi
- Potrebbe consegnare duplicati

Prestazioni e Overhead

TCP comporta:

- Maggiore latenza iniziale per stabilire la connessione
- Header più grande (20 byte + opzioni)
- Overhead per gestione stato e controllo
- Buffer di ricezione e trasmissione

UDP offre:

- Nessuna latenza iniziale
- Header minimo (8 byte)
- Nessun overhead di stato
- Nessun buffer necessario

Casi d'Uso Specifici

TCP è ottimale per:

- Trasferimento file (FTP)
- Web browsing (HTTP/HTTPS)
- Email (SMTP, IMAP, POP3)
- Database remoti
- Qualsiasi applicazione che richiede affidabilità

UDP è preferibile per:

- Streaming video live
- Videoconferenze
- Giochi online multiplayer
- DNS (Domain Name System)
- Monitoraggio e telemetria

2.4 Porte e Socket in Dettaglio

Sistema di Porte

Le porte forniscono punti di accesso per i servizi di rete:

Well-known Ports (0-1023):

- Porta 20/21: FTP
- Porta 22: SSH
- Porta 23: Telnet
- Porta 25: SMTP
- Porta 53: DNS
- Porta 80: HTTP
- Porta 443: HTTPS

Registered Ports (1024-49151):

- Utilizzate da applicazioni utente
- Registrate presso IANA
- Esempio: MySQL (3306), PostgreSQL (5432)

Dynamic Ports (49152-65535):

- Utilizzate per connessioni temporanee
- Allocate dinamicamente dal sistema
- Usate come porte sorgente per client

Socket e Programmazione di Rete

Un socket rappresenta un endpoint di comunicazione:

Componenti di un Socket:

- Indirizzo IP (identificazione host)
- Numero di porta (identificazione servizio)
- Protocollo utilizzato (TCP/UDP)

Operazioni Socket TCP:

1. Creazione socket
2. Binding a indirizzo/porta
3. Listen (per server)
4. Accept (per server)
5. Connect (per client)
6. Send/Receive dati
7. Close connessione

Operazioni Socket UDP:

1. Creazione socket
2. Binding (opzionale)
3. SendTo/ReceiveFrom
4. Close

3. Controllo di Flusso Avanzato

3.1 Finestra Scorrevole

Il meccanismo della finestra scorrevole permette trasmissione efficiente:

- Dimensione finestra determina quanti byte possono essere in volo
- Ricevente può modificare dimensione finestra
- Permette pipeline di dati
- Gestisce acknowledgment cumulativi

3.2 Gestione della Congestione Dettagliata

Fasi della Congestione:

1. Slow Start
    - Inizia con finestra piccola
    - Raddoppia ad ogni ACK

    - Continua fino a soglia
2. Congestion Avoidance
    - Incremento lineare
    - Più conservativo
    - Previene saturazione
3. Fast Retransmit/Recovery
    - Rileva perdite da ACK duplicati
    - Ritrasmette senza attendere timeout
    - Mantiene throughput più alto

3.3 Timeout e Ritrasmissioni

Calcolo RTO (Retransmission Timeout):

- Basato su RTT (Round Trip Time)
- Usa media mobile e deviazione
- Si adatta alle condizioni di rete
- Evita ritrasmissioni premature

4. Protocolli Aggiuntivi e di Supporto

4.1 DHCP (Dynamic Host Configuration Protocol)

DHCP permette l'assegnazione automatica degli indirizzi IP. Il suo funzionamento avviene attraverso quattro fasi fondamentali:

- DHCP Discover: il client cerca server DHCP disponibili
- DHCP Offer: il server propone una configurazione
- DHCP Request: il client richiede formalmente la configurazione
- DHCP Acknowledge: il server conferma l'assegnazione

4.2 ARP (Address Resolution Protocol)

ARP gestisce la mappatura tra indirizzi IP e MAC. Il processo funziona così:

- Un host invia una richiesta ARP in broadcast sulla rete locale
- L'host con l'indirizzo IP cercato risponde con il proprio MAC
- Le informazioni vengono memorizzate nella cache ARP
- La cache viene aggiornata periodicamente

4.3 NAT (Network Address Translation)

NAT permette di condividere un singolo indirizzo IP pubblico tra più dispositivi:

- Traduce gli indirizzi IP privati in pubblici
- Mantiene una tabella delle connessioni attive
- Gestisce il multiplexing delle porte
- Fornisce un livello base di sicurezza

5. Il Problema della Contesa e Accesso al Canale

La contesa del canale trasmissivo è un problema fondamentale nelle reti di comunicazione, specialmente in quelle wireless. Quando più stazioni cercano di comunicare contemporaneamente sullo stesso canale, possono verificarsi collisioni che degradano le prestazioni della rete.

5.1 - Problemi Fondamentali

Il Terminale Nascosto (Hidden Terminal)
Immaginiamo tre nodi A, B e C, dove B può comunicare con entrambi, ma A e C non possono sentirsi a vicenda. In questo scenario:

- A inizia a trasmettere a B
- C, non potendo rilevare la trasmissione di A, inizia anche lui a trasmettere a B
- Le trasmissioni collidono in B, causando la perdita di entrambi i pacchetti

Il Terminale Esposto (Exposed Terminal)
In una configurazione simile:

- B sta trasmettendo ad A
- C vuole trasmettere a D
- C rileva la trasmissione di B e si astiene dal trasmettere
- In realtà, C potrebbe trasmettere senza problemi perché la sua trasmissione non interferirebbe con quella di B ad A

5.2 - Protocolli di Accesso al Mezzo

CSMA (Carrier Sense Multiple Access)

Il CSMA è una famiglia di protocolli che cercano di minimizzare le collisioni ascoltando il canale prima di trasmettere:

CSMA/CD (Collision Detection)

- La stazione ascolta il canale prima di trasmettere
- Se il canale è libero, inizia la trasmissione
- Durante la trasmissione continua a monitorare il canale
- Se rileva una collisione:
    1. Interrompe immediatamente la trasmissione

2. Invia un segnale di jamming
3. Attende un tempo casuale (backoff esponenziale)
4. Riprova a trasmettere

CSMA/CA (Collision Avoidance)
Usato principalmente nelle reti wireless dove il CD è difficile:

1. La stazione che vuole trasmettere ascolta il canale
2. Se il canale è libero, aspetta un tempo DIFS (DCF InterFrame Space)
3. Inizia un conteggio alla rovescia casuale (contention window)
4. Se durante il conteggio rileva trasmissioni, si ferma
5. Se arriva a zero, trasmette

CDMA (Code Division Multiple Access)

Il CDMA è una tecnica più sofisticata che permette a più stazioni di trasmettere simultaneamente sullo stesso canale:

Principio di Funzionamento:

- Ogni stazione ha un codice univoco (sequenza di chip)
- I dati vengono moltiplicati per il codice
- Le trasmissioni si sovrappongono nel tempo e nella frequenza
- Il ricevitore usa il codice corretto per estrarre il segnale desiderato

Vantaggi:

- Maggiore efficienza spettrale
- Resistenza alle interferenze
- Sicurezza intrinseca
- Nessuna necessità di sincronizzazione temporale

Svantaggi:

- Maggiore complessità implementativa
- Necessità di controllo accurato della potenza
- Limitazioni sulla capacità totale

Protocolli Basati su Contesa

Esistono anche protocolli che gestiscono esplicitamente la contenzione:

ALOHA:

- Puro: trasmette appena ha dati da inviare

- Slotted: trasmette solo all'inizio di slot temporali

Token Ring:

- Un token circola tra le stazioni
- Solo chi ha il token può trasmettere
- Eliminazione delle collisioni ma overhead del token