Appunti di Reti di Calcolatori

Versione Approfondita

Sintesi delle lezioni e delle esercitazioni

19 giugno 2025

Indice

1 Introduzione e Concetti Fondamentali

1.1 Cos'è una Rete di Calcolatori?

Una rete di calcolatori è un insieme di **dispositivi autonomi e interconnessi** in grado di scambiarsi informazioni.

- **Classificazione per estensione:**
  - **PAN** (Personal Area Network), **LAN** (Local Area Network), **MAN** (Metropolitan Area Network), **WAN** (Wide Area Network).
  - **Internet**: La "rete di reti", che interconnette le precedenti su scala globale.

1.2 Misure di Prestazione

Le prestazioni di una rete si misurano con diversi indici, a seconda dell'applicazione.

- **Banda (Bandwidth/Bitrate)**: Quantità di dati trasferibili nell'unità di tempo (es. $\mathrm{Mbit\,s^{-1}}$).
- **Ritardo (Delay/Latency)**: Tempo che un pacchetto impiega per andare da sorgente a destinazione.
- **Jitter**: La **variazione del ritardo** nel tempo. È cruciale per applicazioni real-time: un jitter elevato causa una ricezione a scatti e instabile, rendendo inutilizzabili servizi come videochiamate o gaming online.

  Esempio del Prof: Il Furgone vs la Fibra. Un furgone pieno di hard disk che viaggia da Milano a Roma ha una banda altissima (trasporta terabyte di dati) ma un ritardo enorme. Una connessione in fibra ha una banda inferiore ma un ritardo bassissimo. La rete "migliore" dipende dal requisito: per un backup di grandi dimensioni, il furgone è superiore; per una videochiamata, la fibra è indispensabile.

  Avvertimento del Prof: Leggere i Contratti. Fate attenzione ai termini. Un contratto che promette una banda *"fino a* $1\,\mathrm{Gbit\,s^{-1}}$*"* non garantisce nulla, se non un tetto massimo teorico. Una garanzia seria specificherebbe una banda minima, *"almeno X"*, ma nessun provider lo fa per le utenze domestiche.

1.3 Topologie di Rete

La "mappa" dei collegamenti fisici. La scelta della topologia influenza l'efficienza e la resilienza della rete.

- **Anello (Ring)**: Ogni nodo è connesso a due vicini. Protocolli come *Token Ring* si basano su questa topologia per gestire l'accesso al mezzo. È molto vulnerabile: se un nodo o un collegamento si rompe, il "token" si perde e l'intera rete si ferma.
- **Stella (Star)**: Tutti i nodi sono connessi a un dispositivo centrale (Hub o Switch). È la topologia più comune oggi nelle LAN.
  - **Vantaggi**: Semplice da gestire e robusta alla rottura di un singolo cavo (solo il nodo collegato viene isolato).
  - **Svantaggi**: Il dispositivo centrale è un **single point of failure**. Se si guasta, tutta la rete crolla.
- **Bus**: Tutti i nodi sono connessi a un unico cavo condiviso. Se il bus si spezza, la rete viene partizionata in due segmenti non comunicanti.
- **Albero (Tree)**: Struttura gerarchica, ottima per la scalabilità e per un instradamento efficiente dei segnali. Viene usata per organizzare domini e sottodomini di rete.

1.4 Commutazione di Circuito vs. di Pacchetto

Commutazione di Circuito (es. vecchia telefonia)

- **Funzionamento**: Viene stabilito un **percorso fisico dedicato** e le risorse (canale) vengono **riservate** per tutta la durata della comunicazione (fase di setup).
- **Vantaggi**: Prestazioni garantite, ritardo costante, pacchetti in ordine. Una volta stabilito il circuito, la comunicazione è estremamente efficiente e veloce.
- **Svantaggi**: **Spreco di risorse e costo elevato**. Si paga per il tempo di prenotazione del canale, anche se non si trasmette nulla per l'80% del tempo. Il setup iniziale introduce un ritardo.

Commutazione di Pacchetto (es. Internet)

- **Funzionamento**: L'informazione viene spezzettata in **pacchetti autonomi**, ognuno con indirizzo di sorgente e destinazione. I pacchetti di utenti diversi **condividono** gli stessi canali.
- **Vantaggi**: **Uso efficiente delle risorse condivise**, che si traduce in costi molto più bassi (modello "flat rate"). La rete è più resiliente, perché i pacchetti possono seguire percorsi alternativi se un collegamento si guasta.
- **Svantaggi**: **Nessuna garanzia di prestazioni**. La banda è condivisa, quindi le prestazioni dipendono dal carico della rete. I pacchetti possono essere persi, arrivare in disordine o essere duplicati.

2 Architettura di Rete e il Protocol Stack

2.1 Struttura a Livelli (Stack)

La complessità della comunicazione è gestita suddividendo i compiti in **livelli (layer)** astratti. Ogni livello risolve un problema specifico, offre servizi a quello superiore e usa i servizi di quello inferiore.

> **Analogia del Prof: Gli Innamorati.** Un innamorato italiano vuole comunicare con una innamorata giapponese. Non parlano la stessa lingua e usano sistemi postali diversi.
>
> - **Livello 3 (Traduzione):** Risolve il problema della lingua.
> - **Livello 2 (Formato/Posta):** Risolve il problema del sistema postale.
> - **Livello 1 (Fisico):** L'aereo che trasporta la lettera.
>
> Se l'innamorata fosse stata finlandese, sarebbe bastato cambiare solo il traduttore (livello 3), senza toccare gli altri livelli. Questa è la potenza della modularità.

2.1.1 Il Modello TCP/IP (Internet Stack)

1. **Livello Applicazione**: Protocolli vicini all'utente finale (HTTP per il web, SMTP per l'email, DNS per la risoluzione dei nomi).
2. **Livello Trasporto**: Fornisce la comunicazione logica end-to-end tra processi. I due protocolli principali sono TCP e UDP.
3. **Livello Rete (o Internet)**: Gestisce l'instradamento dei pacchetti attraverso reti eterogenee (internetworking). Il suo protocollo è l'IP.
4. **Livello Collegamento (o Link)**: Gestisce la comunicazione tra nodi adiacenti sulla stessa rete locale. Aggiunge indirizzi fisici (MAC).
5. **Livello Fisico**: Gestisce la trasmissione dei bit sul mezzo fisico (cavi, fibra ottica, onde radio), definendo voltaggi, frequenze, etc.

2.2 Incapsulamento e De-incapsulamento dei Dati

Quando un dato scende lungo lo stack, ogni livello aggiunge le proprie informazioni di controllo (**header** e a volte **trailer**), "incapsulando" i dati del livello superiore.

- I **dati** dell'applicazione...
- ...diventano un **segmento** a livello Trasporto (header TCP/UDP).
- ...diventano un **pacchetto (o datagramma)** a livello Rete (header IP).
- ...diventano un **frame (o trama)** a livello Link (header MAC e trailer FCS).
- ...vengono trasmessi come **bit** a livello Fisico.

Il processo inverso, il de-incapsulamento, avviene in ricezione, dove ogni livello "scarta" il proprio header e passa i dati al livello superiore.

3 Livello 2: Collegamento Dati (MAC)

Gestisce la comunicazione tra nodi fisicamente connessi sulla stessa rete locale (LAN).

3.1 Indirizzo MAC (o Fisico)

Ogni scheda di rete (NIC) ha un indirizzo univoco di $48\,\mathrm{bit}$ ($6\,\mathrm{byte}$), scritto in esadecimale (es. `0A:1B:2C:3D:4E:5F`). Viene usato per identificare sorgente e destinazione all'interno di una LAN. L'indirizzo di broadcast MAC è `FF:FF:FF:FF:FF:FF`.

3.2 Dispositivi di Livello 1 e 2

- **Hub/Repeater (Livello 1)**: È un dispositivo "stupido" che opera a livello fisico. Riceve un segnale su una porta e lo ripete, rigenerandolo, su tutte le altre porte.
  - **Dominio di collisione unico**: Tutta la rete connessa all'hub è un unico dominio di collisione. Se due nodi parlano contemporaneamente, i segnali si scontrano e i dati vengono persi.
- **Switch (o Commutatore, Livello 2)**: È un dispositivo "intelligente" che opera a livello di collegamento.
  - **Separa i domini di collisione**: Ogni porta dello switch è un dominio di collisione separato. Questo permette a più coppie di host di comunicare contemporaneamente senza interferire tra loro.
  - **Apprendimento MAC**: Lo switch impara quali indirizzi MAC sono raggiungibili attraverso quali porte. Costruisce dinamicamente una **tabella MAC** e inoltra i frame solo sulla porta di destinazione, invece di fare flooding su tutte le porte (come farebbe un hub).

3.3 Protocollo ARP (Address Resolution Protocol)

Domanda da esame: Come si trova l'indirizzo MAC di un host conoscendo il suo indirizzo IP all'interno di una LAN? **Risposta**: Con il protocollo ARP.

1. L'host A vuole inviare un pacchetto all'IP di B, ma non conosce il suo indirizzo MAC.
2. A invia un **broadcast ARP** sulla LAN. Il frame ha come MAC di destinazione `FF:FF:FF:FF:FF:FF` e contiene la domanda: "Chi ha l'indirizzo IP `IP_B`?".
3. Tutti gli host sulla LAN ricevono ed esaminano la richiesta. Solo l'host B, riconoscendo il proprio IP, risponde.
4. B invia una **risposta unicast** ad A, dicendo: "Io ho l'IP `IP_B`, e il mio MAC address è `MAC_B`".
5. A riceve la risposta e memorizza l'associazione (`IP_B`, `MAC_B`) nella sua **tabella ARP** (o cache ARP) per non dover ripetere il processo la volta successiva.

```
# Esempio di output del comando arp -a
> arp -a
Interfaccia: 137.204.103.100 --- 0x12
  Indirizzo Internet    Indirizzo fisico      Tipo
  137.204.103.1         70-4c-a5-5f-8e-8f     dinamico
  137.204.103.254       70-4c-a5-5f-8e-8f     dinamico <-- Router
  137.204.103.255       ff-ff-ff-ff-ff-ff     statico  <-- Broadcast IP
```

4 Livello 3: Rete (IP)

Gestisce l'instradamento dei pacchetti attraverso reti diverse (Internetworking). Il suo compito è far arrivare un pacchetto dalla sorgente originale alla destinazione finale, anche se si trovano su reti fisiche diverse e lontane.

4.1 Il Router

È il dispositivo chiave del livello 3. Connette reti diverse e prende decisioni di instradamento basate sugli indirizzi IP.

- **Forwarding (Inoltro)**: L'azione di spostare un pacchetto da un'interfaccia di input a un'interfaccia di output appropriata. Questa decisione viene presa consultando la **tabella di routing**. Questo è il **Data Plane**.
- **Routing**: Il processo con cui i router costruiscono e mantengono le loro tabelle di routing. Lo fanno scambiando informazioni con altri router tramite **protocolli di routing** (es. RIP, OSPF, BGP). Questo è il **Control Plane**.
- **Default Gateway (o Router di Default)**: L'indirizzo IP del router a cui un host invia tutti i pacchetti destinati a reti esterne alla propria LAN. È una configurazione fondamentale per ogni dispositivo in rete.

4.2 Indirizzamento IPv4, Subnetting e Supernetting

4.2.1 Indirizzi IPv4 e Classi

- Un indirizzo IPv4 è di $32\,\mathrm{bit}$, suddiviso in una parte di rete (**Net-ID**) e una parte di host (**Host-ID**).
- Gli indirizzi sono raggruppati in classi (A, B, C) che definiscono la lunghezza predefinita del Net-ID.
- **Indirizzi riservati** in ogni rete/sottorete:
  - Indirizzo con **Host-ID tutto a 0**: identifica la rete stessa (es. `192.168.1.0`).
  - Indirizzo con **Host-ID tutto a 1**: è l'indirizzo di **broadcast** per quella rete (es. `192.168.1.255`).

4.2.2 Metodologia per il Subnetting

Il subnetting permette di suddividere una rete grande (es. una classe B) in sottoreti più piccole e gestibili.

1. **Determinare il numero di bit da "rubare"**: Per creare N sottoreti, servono k bit, dove $2^k \geq N$. Questi k bit vengono presi dalla parte di Host-ID e aggiunti al Net-ID.
2. **Calcolare la nuova maschera di rete**: La maschera originale viene estesa di k bit. Ad esempio, una `/16` (Classe B) divisa in 8 sottoreti (2^3) diventa una `/19` ($16 + 3$).
3. **Elencare le sottoreti**: Si elencano tutte le combinazioni possibili dei k bit rubati.
4. **Per ogni sottorete, calcolare**:
   - **Indirizzo di Sottorete**: La parte di host rimanente è tutta a 0.

- **Primo Host Utilizzabile**: Indirizzo di sottorete + 1.
- **Ultimo Host Utilizzabile**: Indirizzo di broadcast - 1.
- **Indirizzo di Broadcast**: La parte di host rimanente è tutta a 1.
- **Router (convenzione)**: L'ultimo host utilizzabile.

4.2.3 Supernetting

Il processo inverso: si fondono più reti adiacenti in un'unica super-rete. Si "restituiscono" bit dal Net-ID all'Host-ID, accorciando la maschera.

> **Attenzione del Prof**: Il supernetting è possibile solo se le reti da unire sono **contigue a livello binario**. Ad esempio, le reti `/24` `...136.0` e `...137.0` possono essere unite in una `/23`, perché in binario `136` è `10001000` e `137` è `10001001`. Differiscono solo per l'ultimo bit, che può essere "inglobato" nella parte host. Le reti `...137.0` e `...138.0` non possono essere unite in una `/23` perché non sono binaramente contigue in quel modo.

4.3 Protocollo DHCP (Dynamic Host Configuration Protocol)

Automatizza l'assegnazione della configurazione di rete (IP, netmask, default gateway, DNS server) ai client.

1. **Discover**: Il client appena connesso invia un broadcast per cercare un server DHCP.
2. **Offer**: I server DHCP che ricevono la richiesta rispondono con un'offerta di configurazione.
3. **Request**: Il client sceglie un'offerta e invia una richiesta formale.
4. **Acknowledge (ACK)**: Il server conferma e assegna la configurazione per un certo periodo (*lease time*).

> **Riflessione sulla Sicurezza (DHCP)**: Un aggressore potrebbe creare un **server DHCP "rogue"** (malevolo) sulla rete. Potrebbe rispondere alle richieste prima del server legittimo, assegnando ai client una configurazione fasulla (es. un router e un DNS controllati dall'aggressore) per intercettare e manipolare tutto il loro traffico (attacco Man-in-the-Middle).

5 Livello 4: Trasporto (TCP e UDP)

Fornisce la comunicazione logica end-to-end tra processi applicativi in esecuzione su host diversi.

5.1 Multiplexing e Demultiplexing con le Porte

Per distinguere tra le varie applicazioni (es. browser, email), il livello trasporto usa i **numeri di porta** (un numero a 16 bit).

- Un socket è identificato dalla quintupla: (protocollo, IP sorgente, porta sorgente, IP destinazione, porta destinazione).
- **Porte note (Well-known ports)**: 0-1023, riservate per servizi standard (es. 80 per HTTP, 443 per HTTPS, 25 per SMTP).

5.2 Confronto Approfondito: TCP vs. UDP

Caratteristica	TCP (Transmission Control Protocol)	UDP (User Datagram Protocol)
Tipo di servizio	**Orientato alla connessione**. Stabilisce una connessione con un *handshake a 3 vie* (SYN, SYN-ACK, ACK) prima di inviare dati.	**Non orientato alla connessione**. Invia datagrammi senza preavviso ("spara e dimentica").
Affidabilità	**Altamente affidabile**. Usa **numeri di sequenza** per rilevare pacchetti persi o fuori ordine. Usa **acknowledgement (ACK)** per confermare la ricezione. Se un ACK non arriva entro un **timeout**, il segmento viene **ritrasmesso**.	**Inaffidabile**. Nessuna garanzia di consegna, ordine o unicità. È compito dell'applicazione gestire eventuali perdite.
Controllo di Flusso	**Sì**. Il ricevitore comunica al mittente quanto spazio ha nel suo buffer tramite il campo *receive window*. Il mittente si adatta per non "inondare" il ricevitore.	**No**. Il mittente può inviare dati a una velocità superiore a quella che il ricevente può gestire.
Controllo di Congestione	**Sì**. Il mittente monitora la rete (perdite, ritardi degli ACK) e riduce la sua velocità di invio (*congestion window*) per evitare di sovraccaricare i router intermedi.	**No**. Non si preoccupa della congestione e può contribuire a peggiorarla.
Velocità e Overhead	Più lento a causa della gestione della connessione, degli ACK e dei controlli. Header di $20\,\mathrm{byte}$.	Molto più veloce e leggero. Header di soli $8\,\mathrm{byte}$.
Casi d'uso	Applicazioni dove ogni singolo bit è fondamentale: Web (HTTP), Email (SMTP), Trasferimento file (FTP).	Applicazioni dove la velocità e il basso ritardo sono più importanti della perdita di qualche pacchetto: Streaming video, gaming online, DNS, VoIP.

Tabella 1: Confronto dettagliato tra TCP e UDP.

Domanda del Prof: Perché non l'ACK dell'ACK? TCP non richiede un ACK per un ACK perché si ricadrebbe nel *problema dei due generali*: ogni conferma richiederebbe un'ulteriore conferma all'infinito, senza mai raggiungere la certezza assoluta. Il **timeout** è la soluzione pragmatica: se non ricevo l'ACK entro un tempo ragionevole, assumo che qualcosa sia andato storto (il pacchetto o l'ACK si è perso) e ritrasmetto. La "fregatura" è che questo può generare duplicati se l'ACK era solo in ritardo.

6 Sicurezza nelle Reti

6.1 Principi di Crittografia

- **Crittografia Simmetrica (es. AES)**: Usa la **stessa chiave** (K_S) per cifrare e decifrare. È molto veloce, ideale per grandi quantità di dati, ma ha il problema della distribuzione sicura della chiave.
- **Crittografia Asimmetrica (es. RSA)**: Usa una coppia di chiavi (pubblica K^+ e privata K^-). Lenta, ma risolve il problema della distribuzione delle chiavi.

6.2 Funzioni di Hash e Firma Digitale

Una funzione di hash (es. SHA-256) crea un'impronta digitale (digest) a lunghezza fissa di un messaggio.

- **Proprietà ideali**: Difficile da invertire e resistente alle collisioni.

Firma Digitale combina hash e crittografia asimmetrica per garantire **autenticità** e **integrità**.

1. **Mittente (Alice)**:
   (a) Calcola $h = H(M)$.
   (b) Cifra l'hash con la sua **chiave privata**: $firma = K_A^-(h)$.
   (c) Invia $(M, firma)$.
2. **Destinatario (Bob)**:
   (a) Calcola $h' = H(M_{\text{ricevuto}})$.
   (b) Decifra la firma con la **chiave pubblica di Alice**: $h_{\text{decifrato}} = K_A^+(firma)$.
   (c) Confronta h' e $h_{\text{decifrato}}$. Se sono uguali, il messaggio è autentico e integro.

   Nota del Prof: Si firma l'hash del messaggio, non l'intero messaggio. Cifrare con RSA un messaggio di grandi dimensioni sarebbe estremamente lento e inefficiente. L'hash è piccolo e a dimensione fissa, rendendo la firma veloce.

7 Fondamenti di Comunicazione Wireless

7.1 Potenza e Decibel (dB, dBm, dBi)

Per gestire le enormi variazioni di potenza nei sistemi wireless, si usa la scala logaritmica dei decibel.

- **Decibel (dB)**: Misura **relativa** di un guadagno/perdita.
  - **+3 dB** $\approx$ Raddoppio della potenza ($\times 2$).
  - **-3 dB** $\approx$ Dimezzamento della potenza ($\div 2$).
  - **+10 dB** = Aumento di 10 volte ($\times 10$).
  - **-10 dB** = Riduzione a 1/10 ($\div 10$).
- **Decibel-milliwatt (dBm)**: Misura **assoluta** di potenza, con il punto di riferimento fisso: **0 dBm =** $1\,\mathrm{mW}$. Permette di usare l'aritmetica dei dB per calcolare potenze assolute.
- **Decibel isotropico (dBi)**: Misura il **guadagno di un'antenna** rispetto a un'antenna isotropica teorica (che irradia uniformemente in tutte le direzioni).

7.2 Progettazione di un Link Wireless: Il Link Budget

Il calcolo del link budget è fondamentale per verificare se una connessione wireless funzionerà a una data distanza e con determinati componenti.

7.2.1 Componenti del Calcolo

- **Potenza del Trasmettitore (P_{TX})**: La potenza emessa dal dispositivo, espressa in .
- **Guadagni**:
  - Guadagno antenna trasmittente (G_{TX} in).
  - Guadagno antenna ricevente (G_{RX} in).
- **Perdite**:
  - Perdite dei cavi e connettori (L_{cavi} in dB).

  - **Free Space Path Loss (FSPL)**: La perdita di segnale dovuta alla propagazione nello spazio. Aumenta con la distanza e la frequenza.

- **Receiver Sensitivity (RS)**: La potenza minima (es. -90) che il ricevitore necessita per funzionare a un certo bitrate.
- **Fade Margin**: Un margine di sicurezza (es. $10\,\mathrm{dB}$ a $20\,\mathrm{dB}$) per garantire l'affidabilità del link in condizioni non ideali.

7.2.2 Metodologia di Calcolo

1. **Calcolare la Potenza Ricevuta (P_{RX})**:

$$P_{RX}[] = P_{TX}[] + G_{TX}[] + G_{RX}[] - L_{\text{cavi}}[\mathrm{dB}] - \text{FSPL}[\mathrm{dB}]$$

2. **Calcolare il Link Budget Effettivo**:

$$\text{Link Budget} = P_{RX} - \text{RS}$$

3. **Verificare la Condizione di Successo**:

$$\textbf{Link Budget} \geq \textbf{Fade Margin}$$

   Se questa condizione è verificata, il link è considerato robusto e affidabile.

   La Regola dei 6 dB (scorciatoia da esame): In spazio libero, ogni volta che si **raddoppia la distanza**, la potenza del segnale ricevuto si riduce di circa $6\,\mathrm{dB}$ (cioè diventa 1/4 della potenza precedente). Viceversa, **dimezzando la distanza**, si guadagnano $6\,\mathrm{dB}$ di segnale.

8 Guida alla Risoluzione degli Esercizi

8.1 Esercizio Tipo: Subnetting

Progettare le sottoreti per il dominio `X.Y.Z.W/M` con le richieste R1, R2, ...

1. **Analizzare il dominio principale**: Calcolare lo spazio totale di indirizzi ($2^{(32-M)}$) e l'intervallo di indirizzi IP.
2. **Ordinare le richieste**: Ordinare le sottoreti richieste dalla più grande (più host) alla più piccola. Questo è il passo più importante per evitare la frammentazione dello spazio.
3. **Per ogni richiesta (dalla più grande)**:
   (a) **Calcolare lo spazio necessario**: Trovare la potenza di 2 (2^k) immediatamente superiore o uguale al numero di host richiesti. Questo darà k bit per la parte host.
   (b) **Calcolare la maschera della sottorete**: Sarà una `/(32-k)`.
   (c) **Assegnare un blocco di indirizzi**: Prendere il primo blocco libero disponibile e assegnarlo alla sottorete.
   (d) **Calcolare gli indirizzi chiave**:
      - **Indirizzo di rete**: Il primo indirizzo del blocco (host-id a 0).
      - **Indirizzo di broadcast**: L'ultimo indirizzo del blocco (host-id a 1).
      - **Router (convenzione)**: L'ultimo indirizzo utilizzabile (broadcast - 1).
      - **Default Gateway**: Sarà il router della rete "madre" (quella di livello superiore).
4. **Ripetere** il punto 3 per tutte le sottoreti.

8.2 Esercizio Tipo: Throughput Hub vs Switch

N host inviano N file. Tempo con Hub? Tempo con Switch?

1. **Calcolare il tempo di trasmissione di un singolo file**: $T_{\text{file}} = (\text{Dimensione File in bit})/(\text{Velocità Link in bit/s})$.
2. **Analisi con Hub (Dominio di collisione unico)**:
   - I trasferimenti sono **necessariamente sequenziali**.
   - $T_{\text{totale}} = N \times T_{\text{file}}$. Il tempo cresce linearmente con il numero di host.
3. **Analisi con Switch (Domini di collisione separati)**:
   - Assumendo destinazioni diverse per ogni file, i trasferimenti possono avvenire **in parallelo**.
   - Si ragiona a fasi:
     (a) Tutti gli N host inviano contemporaneamente allo switch. Tempo: T_{file}.
     (b) Lo switch inoltra contemporaneamente i file alle N destinazioni. Tempo: T_{file}.
   - $T_{\text{totale}} = T_{\text{file}} + T_{\text{file}} = 2 \times T_{\text{file}}$. Il tempo è **costante** e non dipende da N.

8.3 Esercizio Tipo: Link Budget Wireless

Un link a distanza D funziona con link budget 0. A quale distanza D' si ottiene un fade margin di X dB?

1. **Identificare il guadagno necessario**: Il guadagno richiesto è pari al fade margin desiderato (X dB).
2. **Usare la Regola dei 6 dB**: Ricordare che ogni **dimezzamento della distanza** fornisce un guadagno di $6\,\text{dB}$.
3. **Scomporre il guadagno**: Scomporre X dB in multipli di $6\,\text{dB}$.
4. **Applicare i dimezzamenti**: Per ogni $6\,\text{dB}$ richiesto, dimezzare la distanza di partenza.
5. **Esempio**: Per ottenere $18\,\text{dB}$ (che è $3 \times 6\,\text{dB}$), si deve dimezzare la distanza per 3 volte.

$$D' = D/2/2/2 = D/8$$

Alternativa: a distanza D, di quanto aumentare la potenza di trasmissione P_{TX} per avere X dB di margine?

1. L'aumento di potenza richiesto è esattamente pari al fade margin desiderato (X dB).
2. Calcolare la potenza di partenza in .
3. Sommare il fade margin: $P'_{\text{TX}}[] = P_{\text{TX}}[] + X[\text{dB}]$.
4. Se richiesto, convertire il risultato finale da a mW.