

Sommario

Tipologie di rete	4
Tecnologie di comunicazione	5
Topologie di rete	5
Rete a bus	6
Rete ad anello	6
Rete a stella	7
Rete a maglia	7
Tipologie di trasferimento dati	7
Modello ISO/OSI	8
Modello TCP/IP	10
Tipologie di connessione	11
Affidabilità del servizio	12
Segnali	12
Come stabilire quando un bit è 1 o 0 al livello fisico	12
Banda di frequenza	12
Teorema di Nyquist e bit/baud rate	13
Rapporto segnale/rumore	14
Teorema di Shannon	14
Mezzi trasmissivi	14
Trasmissione tramite mezzi elettrici	14
Doppino intrecciato	14
Unshielded Twisted Pair (UTP)	14
Shielded Twisted Pair (STP)	15
Cavo coassiale	15
Fibra ottica	15
Trasmissione wireless	17
Onde radio	18
Microonde	18
Onde infrarosse e millimetriche	18
Trasmissione onde luminose	18
Sistema telefonico	19
Trasmissione sul local loop	19
Capacità e velocità di trasmissione del canale telefonico (e come aumentarle)	20
Frequency Division Multiplexing (FDM)	21
Wavelength Division Multiplexing (WDM)	21
Time Division Multiplexing (TDM)	21

Commutazione di circuito e di pacchetto	22
Rilevamento errori nei livelli fisici	23
Operazioni del livello 2	23
Struttura dei frame	23
Conteggio dei caratteri	24
Character stuffing	24
Bit stuffing	24
Violazioni della codifica del livello fisico	25
Manchester encoding (e differential Manchester encoding)	25
Gestione degli errori	26
Distanza di Hamming, codeword e codice	27
Codice di parità	27
Esempio di distanza di Hamming	28
Correzione degli errori	28
Rilevamento e correzione di burst di errori	29
Rilevamento e correzione di errori nella pratica	29
Cyclic Redundancy Code (CRC)	30
Struttura di un frame	31
Protocolli di comunicazione	31
Protocollo Heaven	31
Protocollo stop and wait	32
Piggybacking	32
Protocolli sliding window (finestra scorrevole)	33
Protocolli go-back-n e selective repeat	34
Protocolli data link	35
High Level Data Link Control (HDLC)	35
Serial Line IP (SLIP)	36
Point to Point Protocol (PPP)	36
MAC (Medium Access Control) e modulazione	37
Allocazione statica	37
Allocazione dinamica	38
Protocolli Carries Sense Multiple Access (CSMA)	38
Protocolli Carries Sense Multiple Access with Collision Detection (CSMA/CD)	39
Fast ethernet	40
Token ring	42
Logical Link Control (LLC – IEEE 802.2)	43
Bridge	44
Switch	45

Algoritmi di routing	46
Algoritmi statici di routing	46
Algoritmi dinamici di routing	48
Routing gerarchico	51
Congestione	51
Traffic shaping (open loop)	52
Leaky bucket (secchio che perde)	52
Token bucket (secchio di gettoni)	53
Flow specification	53
Choke packet (closed loop)	54
Internetworking	54
Reti di router multiprotocollo	54
Tunneling	55
Internet Protocol (IP)	56
Classi di indirizzi IP	56
Indirizzi speciali	57
Indirizzi privati	57
Subnet mask	57
Protocolli ARP e RARP	58
Variable length subnet mask (VLSM)	59
Internet Control Message Protocol (ICMP)	60
TCP e UDP	60
Funzionalità del TCP	61
Apertura di una connessione	61
UDP	63
DNS	63
DNS client/server	65
SMTP (Simple Mail Transfer Protocol)	65
Computer senza mail servers – POP (Post Office Protocol)	65
File Transfer Protocol (FTP)	66
Hypertext Transfer Protocol (HTTP)	66
Telefonia cellulare (Reti cellulari)	67
NAT (Network Address Translation)	68
Streaming audio/video stored	70
Voice/video-over-IP (VoIP)	72
Perdita di pacchetti	72
Ritardo end-to-end	72

Packet jitter	73
Ritardo di riproduzione	73
Correzioni anticipate degli errori	74
Modello Peer-to-Peer (P2P)	75
Overlay networks	76
Classificazioni dei sistemi P2P	77
Hybrid decentralized P2P	78
Purely decentralized P2P	78
Partially centralized P2P	79
Unstructured P2P	79
Structured P2P	80
Loosely structured P2P	80
Esempio di classificazione di applicazioni P2P	81
Condivisione di file P2P (file sharing)	81

Tipologie di rete

Le reti si dividono, come prima cosa, in **cablate**, quindi in **rame o fibra ottica**, e **wireless**, cioè tramite **radiofrequenze o infrarossi**. Per quelle **cablate** abbiamo:

- **Local Area Network (LAN)**, che è una rete locale/aziendale ed ha un raggio compreso tra 10 m e 1 km;
- **Metropolitan Area Network (MAN)**, che è una rete metropolitana con raggio compreso tra 100 m e 10 km;
- **Wide Area Network (WAN)**, che è una rete geografica di raggio compreso tra 10 e 1000 km.

Per quanto riguarda invece le **reti wireless** abbiamo:

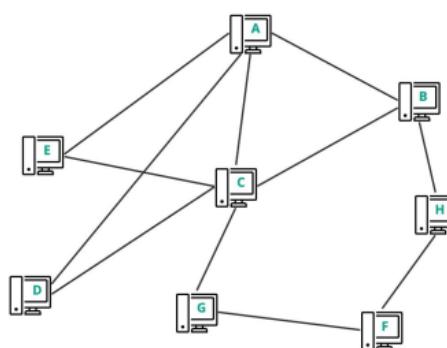
- **Wireless Personal Area Network (WPAN)**, che permette di coprire piccolissime distanze, misurate in cm. Un esempio di questo tipo di reti sono l'NFC e il bluetooth;
- **Wireless LAN (WLAN)**, che in pratica rappresenta la controparte wireless della LAN;
- **Wireless MAN (WMAN)**, che in pratica rappresenta la controparte wireless della MAN;
- **Wireless WAN (WWAN)**, che in pratica rappresenta la controparte wireless della WAN.

Tutte queste diverse tipologie di rete possono essere interconnesse per formare reti sempre più grandi (questo è il principio alla base di Internet).

Tecnologie di comunicazione

Le reti vengono distinte anche in base alle diverse tecnologie di comunicazione.

Abbiamo infatti le **reti punto a punto**, nelle quali ogni calcolatore deve connettersi direttamente ad un altro, ma se non vi è un collegamento diretto tra i due è necessario creare un instradamento (**routing**) tramite altri calcolatori. Questa tecnologia, caratterizzata da un **alto numero di connessioni dedicate**, risulta però chiaramente **costosa**.



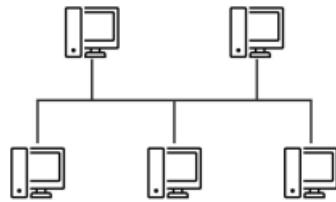
Abbiamo poi le **reti broadcast**, nelle quali tutti i calcolatori sono connessi tramite un unico canale di comunicazione condiviso. In questo caso i messaggi sono ricevuti da tutti i calcolatori, ma letti solamente dal destinatario, il quale viene identificato in maniera univoca tramite il suo indirizzo di rete.



Topologie di rete

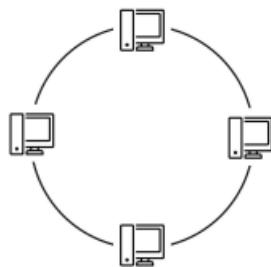
Le topologie di rete definiscono come gli apparati di rete sono collegati tra loro. Ogni elemento connesso alla rete è detto **nodo**, mentre l'informazione scambiata da essi è detta **pacchetto**. Le topologie si dividono in **logiche**, se riguardano il come vengono scambiati i dati tra i nodi, e **fisiche**, se invece riguardano la dislocazione fisica dei nodi. Abbiamo, ovviamente, **diversi tipi di topologie fisiche**: rete a bus, ad anello, a stella e a maglia.

Rete a bus



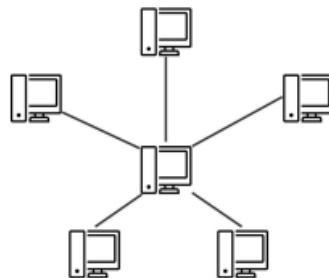
Nella rete a bus le informazioni viaggiano su un **unico canale**, di conseguenza tutti i nodi possono leggere le informazioni in viaggio. Solo il nodo destinatario però legge il pacchetto, mentre tutti gli altri, dopo averlo ricevuto, lo scartano. Esistono inoltre **nodi terminali** per i messaggi senza un destinatario. Questo tipo di rete è **semplice da realizzare e da estendere** ma ha **velocità ridotte**.

Rete ad anello



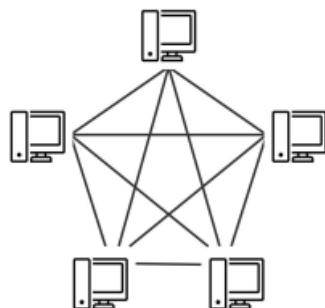
Come nella rete a bus, anche nella rete ad anello le informazioni viaggiano su un **unico canale**. In questo caso abbiamo però **due modalità di trasmissione: unidirezionale**, quando i pacchetti sono trasmessi in un unico senso, orario o antiorario che sia, o **bidirezionale**, nel caso in cui i pacchetti vengano trasmessi in entrambe le direzioni. Per quanto riguarda invece la trasmissione e l'invio dei pacchetti, un nodo non destinatario lo inoltra al successivo. Questa azione si ripete fino a quando non viene raggiunto il nodo destinatario, che legge il pacchetto e blocca l'inoltro, oppure quando il pacchetto torna al mittente, ed in questo caso la comunicazione si interrompe poiché il destinatario non è stato trovato. Questo tipo di rete è **veloce e semplice da estendere** ma ha un **problema di bassa tolleranza ai guasti**.

Rete a stella



Nella rete a stella esiste un nodo centrale che gestisce la comunicazione tra i nodi. Tutte le informazioni vengono quindi inviate ad esso, il quale si occupa di indirizzare il pacchetto verso il destinatario. Questo tipo di rete è **semplice da realizzare** ed offre **buone velocità**, ma ha una **tolleranza parziale ai guasti**.

Rete a maglia



Nella rete a maglia ogni nodo è collegato a tutti gli altri. Di conseguenza tutti i nodi possono leggere le informazioni in viaggio, ma i nodi non destinatari, una volta ricevuto il pacchetto, procedono a scartarlo, mentre l'unico a leggerlo (il pacchetto) sarà il nodo destinatario. Questo tipo di rete ha molta **più tolleranza ai guasti** ed è **semplice da realizzare per pochi calcolatori**, ma diventa più costosa e complicata all'aumentare di essi (calcolatori).

Tipologie di trasferimento dati

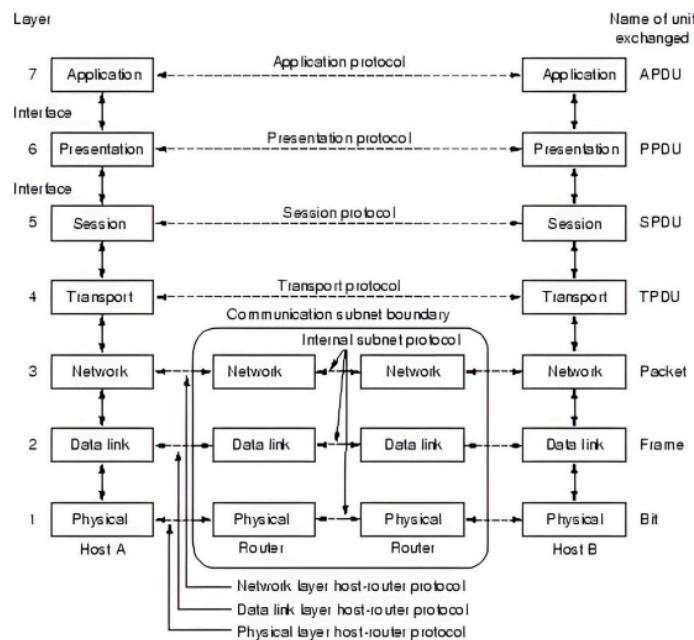
Esistono diverse tipologie di trasferimento dati:

- **simplex**, ossia un trasferimento monodirezionale (A->B);
- **half-duplex**, cioè un trasferimento bidirezionale ma alternato (A->B o B->A);
- **full-duplex**, ossia un trasferimento bidirezionale che può avvenire anche contemporaneamente (come l'half-duplex, ma i trasferimenti nelle due direzioni possono anche avvenire contemporaneamente).

Il **multiplexing**, invece, è una tecnica che permette di inviare più messaggi in un unico segnale, utilizzando un solo canale di comunicazione.

Modello ISO/OSI

Uno dei modelli di riferimento per le architetture di rete è il modello ISO/OSI. Questo definisce il numero e le caratteristiche funzionali dei livelli, ma anche le relazioni tra loro. Il modello ISO/OSI è composto da **7 livelli**, di cui **3 fisici** (livelli 1-3) e **4 applicativi** (livelli 4-7).



Procediamo ora alla descrizione dei livelli (da 1 a 3 livelli fisici, dopodichè livelli applicativi):

- **Fisico (1)**, si occupa della **trasmissione dei dati grezzi (bit) su un canale di comunicazione**, specificando le caratteristiche meccaniche, elettriche e procedurali dell'apparato di connessione. (come possono essere durata di un singolo bit e tipo di trasmissione ad es.);
- **Data link (2)**, si occupa della **definizione del frame e dell'indirizzamento in funzione del mezzo fisico**. In particolare finalizza il framing dei dati, **prepara ed invia i frame in sequenza**, ed infine si assicura che il frame sia giunto a destinazione mediante un **segnale di acknowledgement (ack)**. Questo livello si occupa anche della **regolazione del traffico sulla rete**, in modo da evitare che il ricevente sia sommerso di dati. I segnali ack possono essere inviati sia come **frame separati**, ma così facendo vanno in competizione con il traffico di rete, che con la tecnica del **piggybacking**, che consente di includere gli ack all'interno di un pacchetto dati già in transito. Il livello data link permette di evitare la presenza di errori ed ha, **nelle reti broadcast**, un sottolivello chiamato **Media Access Control (MAC)**, che **controlla e gestisce l'accesso al canale di trasmissione**;
- **Network (3)**, si occupa della **creazione dei pacchetti**, ma anche dell'**indirizzamento e dell'instradamento degli stessi ad alto livello** (astrazione). In particolare si occupa di **specificare e controllare il**

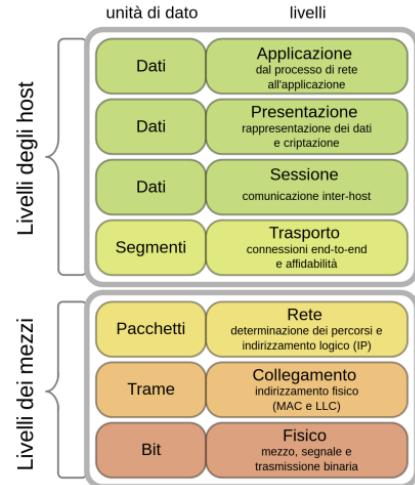
funzionamento della sottorete di comunicazione, effettua il **routing**, che può essere **statico o dinamico**, registra il traffico generato dalla rete (**accounting**) ed effettua la **conversione di dati**, nel caso in cui si stia **comunicando con reti differenti**. Questo livello, inoltre, è anche quello deputato alla **selezione dei pacchetti da frammentare**;

- **Trasporto** (4), si occupa dell'**invio e della ricezione dei dati**, con anche una **fase di controllo e, se possibile, correzione degli errori**. Questo livello viene chiamato **end-to-end** poiché isola i livelli superiori dal mezzo fisico. **Altri compiti** del livello trasporto sono la **divisione dei dati in pacchetti** e, se richiesto, il **controllo che essi (i pacchetti) giungano a destinazione**. Si occupa anche del **tipo di connessione da creare**:
 - una connessione network per ciascuna connessione transport;
 - una singola connessione network per molte connessioni transport (**multiplexing**);
 - molte connessioni network per una singola connessione transport.

Offre, inoltre, **due tipologie di connessione al livello superiore**:

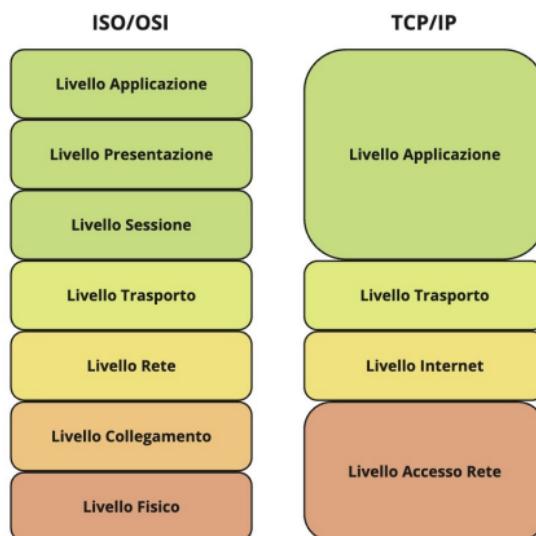
connection oriented, ossia un **canale punto a punto affidabile**, nel quale i dati vengono consegnati in ordine e senza errori, e **connectionless**, dove non solo l'**invio è senza garanzia di consegna**, ma nel caso in cui i dati arrivassero, è quasi certo che non lo farebbero in ordine;

- **Session** (5), si occupa delle **sessioni di comunicazione**, dell'inizializzazione alla chiusura. In particolare **consente ad utenti su sistemi diversi di stabilire una sessione di comunicazione**. Questo livello utilizza la tecnica del **token management** per regolare la trasmissione tra le parti coinvolte dalla comunicazione, in modo da **evitare sovrapposizioni**. Un'altra tecnica utilizzata è quella del **checkpointing in fase di download**, la quale permette di salvare periodicamente lo stato del download in corso, in modo che in caso di interruzioni o problemi l'utente possa riprendere il processo dall'ultimo checkpoint e non dall'inizio;
- **Presentazione** (6), si occupa di **formattare e trasformare i dati in base alla loro rappresentazione locale**, fornendo inoltre anche la loro (dei dati) cifratura/decifratura. In particolare si occupa della **conversione dei tipi standard** (ad es. caratteri o interi), attua i **meccanismi di cifratura/decifratura** (come già detto in precedenza) e utilizza la **tecnica del checkpointing**;
- **Applicazione** (7), che è l'**interfaccia tra il sistema di comunicazione e le applicazioni**, di conseguenza **offre servizi all'utente**, come possono essere trasferimento dei file, posta elettronica e terminale virtuale.



Modello TCP/IP

Nel tempo c'è stata una crescita della rete che ha portato alla nascita di protocolli complessi, in modo da connettere diverse reti in modo più semplice e aumentare sia l'affidabilità che la tolleranza ai guasti. E' stato quindi sviluppato un modello di riferimento semplificato che contiene protocolli multipli, oltre a quelli da cui prende il nome, ossia TCP e IP. Nel modello TCP/IP sono previsti **4 livelli** invece dei 7 del modello ISO/OSI, infatti il livello applicazione comprende anche i livelli presentazione e session, mentre il livello accesso rete comprende sia il livello data link che il livello fisico. Per quanto riguarda invece il livello internet, questo corrisponde al livello network del modello ISO/OSI.



Passiamo ora alla descrizione dei livelli:

- **Accesso rete (1), lascia la possibilità di utilizzare i dispositivi di rete con propri protocolli**, non specificando appunto come l'accesso alla rete debba

funzionare nel dettaglio. Nel caso di comunicazione con dispositivi differenti, inoltre, l'apparato di rete provvederà alla conversione multiprotocollo;

- **Internet** (2), si occupa del **routing** (instradamento) dei pacchetti e del **controllo della congestione della rete**, ossia regola la velocità del flusso di dati, in modo da rendere le prestazioni il più efficienti possibile. Questo livello, inoltre, **utilizza il protocollo IP (Internet Protocol)**;
- **Trasporto** (3), che ha due protocolli fondamentali, ossia il **Transmission Control Protocol (TCP)** e lo **User Datagram Protocol (UDP)**. Il primo è **reliable connection-oriented** ed in particolare il mittente frammenta il messaggio in pacchetti, che vengono poi riuniti dal destinatario per ricostruire il messaggio. Il protocollo **UDP invece è unreliable connectionless** e, di conseguenza, è molto probabile la perdita di pacchetti;
- **Applicazione** (4), in questo livello **si possono utilizzare sia vari protocolli standard**, come possono essere FTP per il trasferimento file, DNS per il mapping degli indirizzi IP (ossia l'associazione di indirizzi IP con nomi di dominio, come ad es. www.example.com) e HTTP per la navigazione web, ma vi è anche la **possibilità di implementare i propri protocolli**.

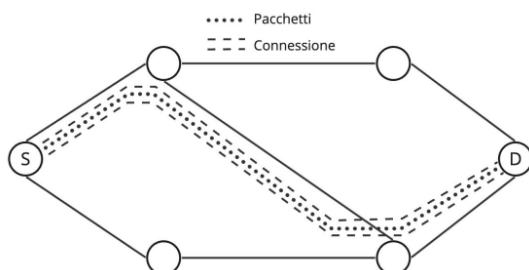
Il modello TCP/IP nasce appunto come modello di riferimento, ma in realtà descrive nel dettaglio il suo funzionamento ed i suoi protocolli. Tra l'altro risulta difficile valutare quando un protocollo sia tale, visto che non si fa troppa distinzione tra protocolli, servizi ed interfacce.

Tipologie di connessione

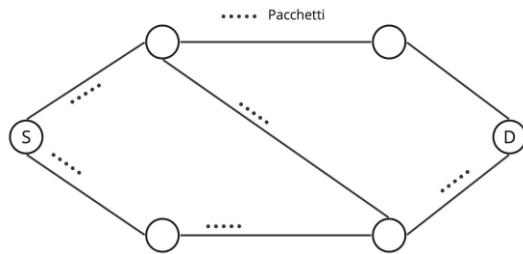
Esistono due tipologie di connessioni: **connection oriented** e **connectionless**.

Nella prima viene stabilita la connessione ed un percorso

(instradamento/routing) attraverso la rete, dopodiché si effettua la **comunicazione**, ossia l'invio dei pacchetti, e si **rilascia la connessione**.



Per quanto riguarda le **connectionless** invece, le **informazioni vengono inviate sulla rete senza un percorso predefinito**, di conseguenza i pacchetti **potrebbero arrivare in ordine sparso o addirittura non arrivare proprio**.



Affidabilità del servizio

Esistono due livelli di affidabilità dei servizi: **reliable** e **unreliable**. Nel primo livello i dati devono essere tutti consegnati al destinatario e, per ogni pacchetto ricevuto, viene inviato un **ack al mittente**, con ack sta per acknowledgement, che è un **sistema di notifica di azioni tra hosts**. Questo livello è più affidabile ma allo stesso tempo più lento del secondo livello, ossia l'**unreliable**. Quest'ultimo infatti non porta nessuna garanzia per quanto riguarda la consegna dei dati, quindi risulta più veloce, ma ovviamente non è affidabile.

Segnali

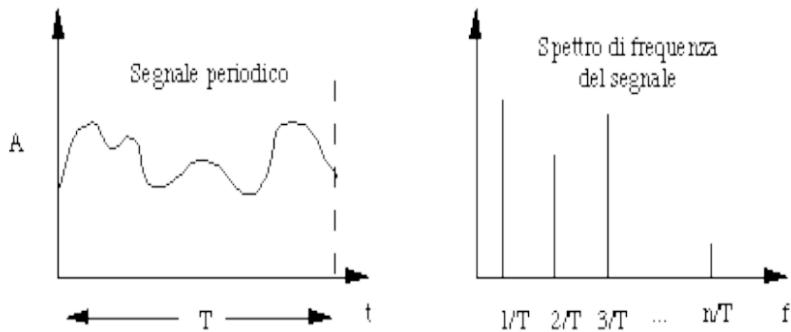
I segnali si dividono in **analogici** e **digitali**. Nei primi il valore del segnale può variare gradualmente in un intervallo costituito da un numero infinito di possibili valori. Nei segnali digitali, invece, il segnale varia bruscamente assumendo in ogni istante solo uno di un insieme finito di valori. La natura dei fenomeni fisici è di tipo analogico e non digitale, in quanto un segnale non varierà mai istantaneamente ma impiegherà sempre un certo intervallo di tempo. A causa dell'interazione con il mezzo trasmissivo, inoltre, la **forma del segnale**, una volta raggiunta la destinazione, non sarà mai esattamente quella di partenza.

Come stabilire quando un bit è 1 o 0 al livello fisico

La rappresentazione dei valori 1 e 0 può avvenire in diversi modi, come ad esempio attraverso segnali elettrici (doppino e cavo coassiale) o anche ottici (fibra ottica). In generale se il segnale è alto avrà valore 1, altrimenti 0.

Banda di frequenza

Un qualunque segnale $g(t)$, di durata T , può essere rappresentato dal suo **spettro di frequenza**, ossia la sua **scomposizione in sinusoidi**.



La **banda di frequenza** è un intervallo comprendente le frequenze di tutte le sinusoidi che descrivono il segnale e viene **influenzata da due fattori**:

- più è breve la durata T del segnale e più è alto il valore della frequenza fondamentale;
- Più velocemente varia $g(t)$ e più numerose sono le armoniche necessarie a descriverla.

I canali attenuano le armoniche mediante **frequenza di taglio** (f_c) o **distorsione dei segnali**. La **banda passante** (di un mezzo fisico) è un intervallo di frequenze che il mezzo fisico è in grado di trasmettere senza alterare oltre certi limiti.

L'**attenuazione** e il **ritardo** sono le **principali alterazioni del segnale** e variano al variare delle frequenze trasmesse. Per quanto riguarda l'**ampiezza di banda**, invece, questa dipende dalle **caratteristiche fisiche del mezzo trasmittivo** e dall'**eventuale presenza di filtri**, come possono essere il **passa-basso**, che permette il passaggio di segnali con frequenze al di sotto di una determinata frequenza di taglio, mentre attenua o blocca le frequenze al di sopra di questa, o anche il **passa-banda**, che consente il passaggio di un intervallo specifico di frequenze (banda passante) mentre attenua le frequenze al di fuori di questa banda (intervallo). Nella trasmissione in un mezzo trasmittivo l'attenuazione subita dal segnale dipende dalla frequenza ed è proporzionale alla distanza percorsa. Se la **banda passante è inferiore alla banda di frequenza**, allora il **segnale viene distorto**, ossia privato di alcune armoniche, ma se un numero sufficiente di armoniche arriva a destinazione, il segnale è comunque utilizzabile.

Teorema di Nyquist e bit/baud rate

Secondo Nyquist un segnale analogico di banda h può essere completamente ricostruito mediante una campionatura effettuata $2 * h$ volte al secondo. In ambito binario, se ciascun campione può assumere uno di n valori distinti, il segnale risulta completamente rappresentato con $2 * h * \lg(n)$ bit al secondo. Il **bit rate**, ossia la **velocità di trasmissione in bit/sec**, di un canale di comunicazione con banda passante di h Hz, su cui è trasmesso un segnale che può assumere V livelli discreti, che prende il nome di **massimo data rate** (bit/sec), è $2 * h * \lg(V)$. Il **baud rate**, invece, è la velocità di **segnalazione di una linea**, ossia quante volte

al secondo essa è in grado di cambiare valore. Il teorema di Nyquist è però valido per **canali totalmente privi di disturbo**, il che **non è realistico**.

Rapporto segnale/rumore

Il rapporto segnale/rumore è il **rapporto tra la potenza del segnale e quella del rumore**, misurata in decibel. Il decibel equivale a **10 volte il logaritmo in base 10 del rapporto segnale/rumore (S/N)**.

Teorema di Shannon

Nel teorema di Shannon, invece, il **massimo data rate di un canale rumoroso**, con banda passante di h Hz e rapporto segnale/rumore calcolato (S/N) come visto in precedenza, è dato da $h * \lg_2(1 + S/N)$.

Mezzi trasmissivi

Trasmissione tramite mezzi elettrici

Doppino intrecciato

Il doppino intrecciato è formato da una **coppia di conduttori di rame intrecciati in forma elicoidale**. Questa particolare forma intrecciata permette di **ridurre i disturbi causati dalle interferenze**. Il doppino intrecciato viene utilizzato in particolare per le **connessioni terminali del sistema telefonico**, ossia quelle dalla casa dell'utente alla centrale più vicina.

Unshielded Twisted Pair (UTP)

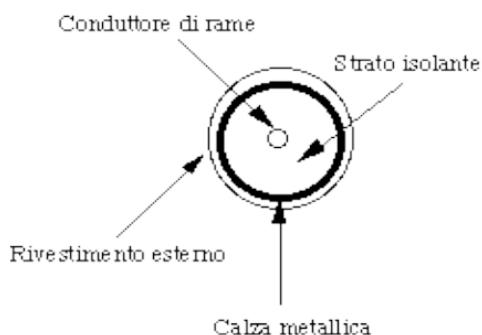
L'UTP è caratterizzato da **coppie di conduttori intrecciati senza uno schermo esterno**. Esistono **diverse categorie di UTP**: in particolare più si sale di categoria e più si avranno **prestazioni migliori in termini di velocità e distanza di trasmissione**. Ad esempio, la categoria 3 è formata da **otto fili isolati, leggermente attorcigliati a coppie, contenuti in una guaina di plastica**. Questo tipo di UTP viene comunemente utilizzata nei cablaggi telefonici interni agli edifici. Un altro esempio è la categoria 5, la quale, rispetto alla categoria 3, presenta un più fitto avvolgimento ed un migliore isolamento. Questa categoria viene **utilizzata soprattutto nei collegamenti in ambito LAN**.

Shielded Twisted Pair (STP)

Il contrario dell'UTP è lo STP, il quale a differenza del primo è **schermato**, quindi più ingombrante, ma offre **migliori prestazioni**. Nonostante ciò, rimane comunque poco utilizzato.

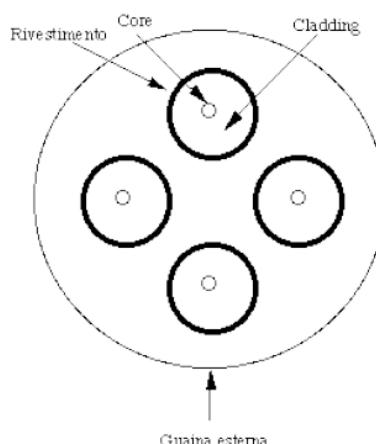
Cavo coassiale

Il cavo coassiale è formato da un **conduttore centrale in rame** circondato da uno **strato isolante** all'esterno del quale vi è una **calza metallica**, il tutto **rivestito in plastica**. Il **migliore isolamento** del cavo coassiale rispetto al doppino consente **maggiori velocità di trasmissione e distanze superiori**. Veniva utilizzato molto in passato nell'ambito dei sistemi telefonici, per quanto riguarda le tratte a lunga distanza. Ormai è sostituito quasi ovunque dalla fibra ottica e viene utilizzato quasi esclusivamente per la TV via cavo ed in vecchie LAN.

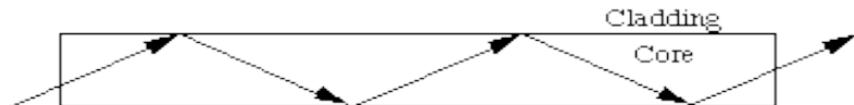


Fibra ottica

La fibra ottica è formata da un **sottilissimo cilindro centrale in vetro (core)**, circondato da uno **strato esterno (cladding)** di vetro avente un **diverso indice di rifrazione** e da una **guaina protettiva**. In genere più coppie sono contenute insieme in una stessa guaina esterna.



Quando un raggio di luce attraversa il confine tra il core ed il cladding subisce una **deviazione**, la quale dipende dagli indici di rifrazione dei due materiali ed è tale che **per certi angoli di incidenza, il raggio resta intrappolato all'interno del core.**



Esistono **due diversi tipi di fibre ottiche**: le monomodali e le multimodali.

Nelle **monomodali** (core di 8-10 micron) la luce di un **singolo raggio** avanza nella fibra, la quale si comporta come una guaina d'onda. Rispetto alle multimodali sono **più costose ma garantiscono distanze maggiori** (fino a 30 km).

Nelle **multimodali** (core di 50 micron), invece, **raggi diversi con diversi angoli possono contemporaneamente propagarsi nella stessa fibra**. Nelle fibre ottiche un **impulso luminoso rappresenta un 1 mentre la sua assenza uno 0**. Le attuali fibre consentirebbero velocità di trasmissione di 50 Tbps ad un bassissimo tasso d'errore, ma mancando sistemi di conversione luminoso/elettrico in grado di operare a tali velocità, **la pratica corrente limita l'uso delle fibre a qualche Gbps**. Nella trasmissione in fibra si ha una **bassa attenuazione**, la quale è dovuta alla particolare trasparenza delle fibre ottiche e all'utilizzo di **tre particolari bande** per la trasmissione:

- la prima, etichettata **850 nm**, con **sorgente LED**, viene utilizzata per le **fibre multimodali in applicazione LAN con distanze medio-piccole**;
- la seconda, etichettata **1300 nm**, con **sorgente LED**, è utilizzata sempre con le **fibre multimodali in applicazione LAN**, ma stavolta con **distanze medie**. Questa banda viene anche portata a **1310 nm**, con **sorgente laser**, costituendo la **prima banda per la fibra monomodale**, la quale viene utilizzata per **applicazioni LAN e trasmissione dati su distanze medio-lunghe**;
- la terza, etichettata **1550 nm**, con **sorgente laser**, viene utilizzata con **fibre monomodali per la trasmissione dati su distanze lunghe**.

Un **sistema di trasmissione ottico ha tre componenti**:

- **sorgente luminosa**, la quale converte un segnale elettrico in impulsi luminosi, e può essere un LED o un laser;
- **mezzo di trasmissione**, ossia la fibra ottica;
- **fotodiodo ricevitore**, che si occupa della conversione di impulsi luminosi in segnali elettrici.

Per quanto riguarda le fibre ottiche, vengono principalmente utilizzate **due topologie**:

- **anello**, dove l'idea è quella di **concatenare più fibre ottiche per creare un anello**. L'interfaccia del singolo sistema può essere **passiva**, nel caso in cui

faccia passare l'impulso luminoso nell'anello, o **attiva**, se converte l'impulso luminoso in elettrico, lo amplifica e poi lo riconverte in luce;

- **stella passiva**, dove l'impulso, inviato da un trasmettitore, arriva in un **cilindro di vetro al quale sono attaccate tutte le fibre ottiche**, realizzando così una **rete broadcast**.

Rispetto al rame, le fibre ottiche hanno diversi **vantaggi**:

- **banda**, infatti due fibre sono più capaci di 1000 doppini;
- **peso**, 100 kg/km contro 8000 kg/km;
- **totale insensibilità a disturbi elettromagnetici**;
- **difficili intrusioni**.

Hanno però anche alcuni **svantaggi**, come il **costo delle giunzioni** e la **comunicazione unidirezionale**.

Trasmissione wireless

Alla base della trasmissione wireless vi sono le **onde elettromagnetiche**, le quali **viaggiano nello spazio** (anche vuoto) **alla velocità della luce** ($3 * 10^8 \text{ m/sec}$) e **possono indurre una corrente in un dispositivo** (antenna) **che le riceve**.

Un'onda elettromagnetica monocromatica, ossia con una ben definita frequenza e lunghezza d'onda, è **costituita da un campo elettrico e da un campo magnetico**, perpendicolari tra loro, che oscillano perpendicolarmente alla direzione di propagazione. Un'onda elettromagnetica consiste in realtà di **due componenti accoppiate**, appunto **una elettrica ed una magnetica**. Un'onda di questo tipo è detta **onda polarizzata piana** e il piano di polarizzazione è il piano in cui oscilla il campo elettrico. Nell'ambito delle onde elettromagnetiche esiste una **relazione fondamentale** per cui $f * \lambda = c$ (nel vuoto), che significa che nel vuoto, quale che sia la frequenza f, le onde elettromagnetiche viaggiano tutte alla stessa velocità c. Nel rame e nelle fibre ottiche la velocità si riduce ai 2/3 di c. Ricordiamo che la **frequenza** è il **numero di oscillazioni al secondo** e si misura in Hertz (Hz), mentre la **lunghezza d'onda** (λ) è la **distanza tra due minimi, o massimi, consecutivi**. Quest'ultima ovviamente decresce al crescere della frequenza $\lambda = c/f$. Per quanto riguarda la trasmissione dati, possono essere utilizzate solo alcune **porzioni dello spettro elettromagnetico**. Infatti i **raggi X** e i **raggi gamma**, ad esempio, non possono essere utilizzati poiché **difficili sia da produrre che da modulare**, non si propagano bene attraverso gli edifici, ed inoltre sono anche **pericolosi per gli esseri viventi**. Procediamo, nei seguenti paragrafi, con la descrizione delle porzioni dello spettro elettromagnetico utilizzabili per la trasmissione dati.

Onde radio

Le onde radio vengono largamente utilizzate sia per **comunicazioni interne che esterne**, in quanto **facili da generare**. **Si propagano in tutte le direzioni (omnidirezionali)**, possono viaggiare per **lunghe distanze e penetrano negli edifici**. Si dividono in quelle a **bassa frequenza** (VLF, LF, MF) e quelle ad **alta frequenza** (HF, VHF). Le **prime** hanno le stesse caratteristiche descritte in precedenza (omnidirezionali, lunghe distanze e penetrano negli edifici), mentre nelle **seconde** si inizia ad avere una **propagazione in linea retta, vengono fermate**, o meglio rimbalzate, dagli ostacoli e tendono ad essere assorbite dal suolo. Quando utilizzate per trasmissioni direzionali, le onde radio ad alta frequenza che colpiscono la ionosfera vengono rifratte e rispedite a terra. Sia le onde radio a bassa frequenza che quella ad alta frequenza sono soggette ad **interferenze elettromagnetiche**.

Microonde

Le microonde vengono utilizzate per **trasmissioni a grande distanza** e come **bande per applicazioni industriali, scientifiche e mediche** (ad es. cellulari). Queste **non attraversano gli edifici** e a causa della dispersione nello spazio **alcune onde possono essere rifratte, arrivando quindi in ritardo e fuori fase** (ossia non contemporaneamente). Il fenomeno della distruzione del segnale causata dall'arrivo fuori fase di onde rifratte prende il nome di **multipath fading**. Per bande fino a circa 8 Ghz, inoltre, le onde vengono assorbite dall'acqua (pioggia).

Onde infrarosse e millimetriche

Le onde infrarosse vengono utilizzate per **comunicazioni su piccole distanze** (telecomandi tv), sono **relativamente direzionali e non passano attraverso corpi solidi**. Quando vengono utilizzate in ambienti chiusi, inoltre, **non interferiscono con altre onde infrarosse impiegate nei locali vicini**.

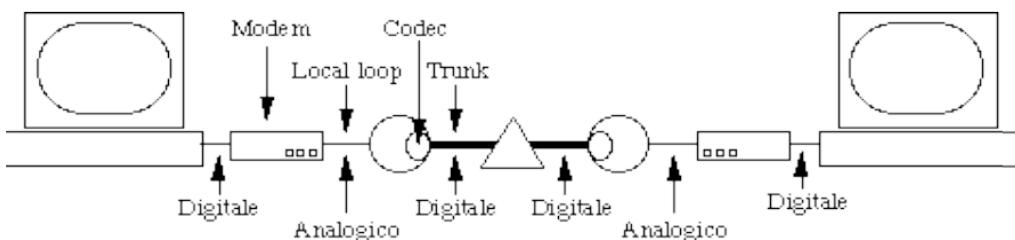
Trasmissione onde luminose

Le onde luminose sono una **soluzione economica** ed in grado di fornire una **grande larghezza di banda**. Tramite segnalazione ottica coerente, basata su laser montati su tetti, si possono trasferire informazioni anche a diverse centinaia di metri. La trasmissione mediante onde luminose è però un **sistema molto delicato**, in quanto un raggio laser inferiore ad 1 mm deve puntare un bersaglio a centinaia di metri. Il raggio laser, inoltre, **non passa attraverso la pioggia e/o nebbia fitta** ed anche **piccole turbolenze d'aria possono portarlo fuori bersaglio**.

Sistema telefonico

Il sistema telefonico, ossia la rete pubblica telefonica commutata, è nato ed evolutosi per la fonìa, di conseguenza poco adatto, almeno fino a qualche tempo fa, per la trasmissione dati. Senza il sistema telefonico la connessione di apparecchiature distanti centinaia di km risulterebbe praticamente impossibile. I sistemi telefonici attuali sono organizzati in una gerarchia multilivello con elevata ridondanza. Queste gerarchie comprendono **vari elementi**:

- **centrale di commutazione**, un'apparecchiatura o infrastruttura che serve a **connettere due apparecchi telefonici**. Di solito queste centrali sono digitali poiché, oltre a **data rate** (velocità con cui i dati vengono trasmessi) più alti, è **più facile ricostruire il segnale** senza introdurre errori e mescolare voce, dati e video;
- **local loop**, il quale **collega il telefono alla più vicina centrale di commutazione**, ad es. attraverso un doppino che trasporta un segnale analogico che occupa una banda di 3 kHz;
- **trunk**, che si occupano di **collegare le centrali di commutazione** anche mediante l'utilizzo di **cavi coassiali**, microonde o fibre ottiche;
- **modem** (modulator-demodulator), un **dispositivo che effettua la trasformazione digitale/analogico in trasmissione e analogico/digitale in ricezione**. Queste trasformazioni sono necessarie per trasmettere dati sul **local loop** (attraverso i doppini ad es.);
- **codec** (coder-decoder), il quale si occupa di effettuare le **trasformazioni analogico/digitale nella prima centrale di commutazione e digitale/analogico nell'ultima**.



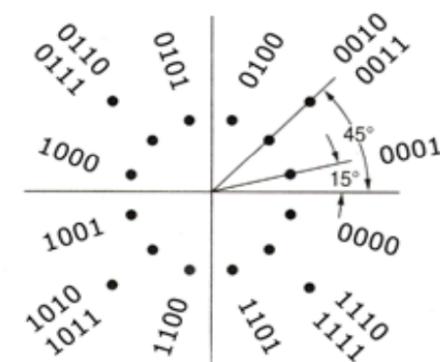
Trasmissione sul local loop

Le trasmissioni sul local loop subiscono **attenuazione e distorsione**. In particolare la **trasmissione digitale genera onde quadre**, le quali hanno un **ampio spettro di frequenze** che però deve confrontarsi con la **banda ridotta del local loop** (3 kHz). Per questo motivo la **trasmissione digitale su local loop può avvenire solo a bassissime velocità di trasmissione**. Per trasmettere un segnale digitale sul local loop **si modula un segnale sinusoidale**, detto **portante**, la cui frequenza è compresa tra 1 e 2 kHz, visto i 3 kHz di banda del local loop. La **modulazione può essere di 3 tipi**:

- modulazione di **ampiezza**, nella quale si varia l'ampiezza;
- modulazione di **frequenza**, dove si varia la frequenza;
- modulazione di **fase**, nella quale si varia la fase, ossia il “ritardo” rispetto al segnale originale.

Capacità e velocità di trasmissione del canale telefonico (e come aumentarle)

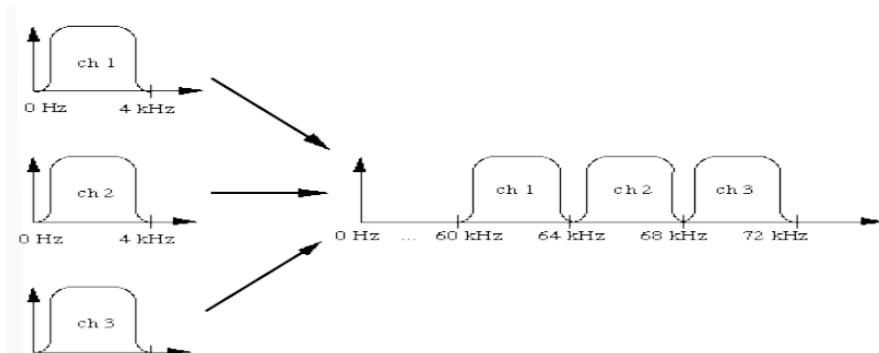
La figura sottostante rappresenta un **constellation pattern**, ossia un diagramma che definisce i punti corrispondenti a valori validi del segnale da trasmettere. Questi punti nello spazio sono dati dalle coordinate polari ampiezza-fase.



modulazione a 4 bit per baud

Frequency Division Multiplexing (FDM)

Il FDM è un tipo di multiplexing adatto alla **gestione di segnali analogici**, nel quale lo **spettro di frequenza è suddiviso in bande più piccole ed ogni comunicazione ha l'utilizzo esclusivo di una di esse**. Ad esempio nel canale telefonico, come sappiamo, abbiamo una banda di 4 kHz centrata su un'apposita frequenza.



Gli **standard CCITT** per il FDM hanno consentito di **organizzare e standardizzare il FDM** a diversi livelli per i sistemi telefonici. Abbiamo ad esempio:

- i **group**, che permettono di convogliare 12 canali;
- i **supergroup**, che equivalgono a 5 group, quindi 60 canali;
- i **mastergroup**, cioè cinque supergroup e quindi 300 canali.

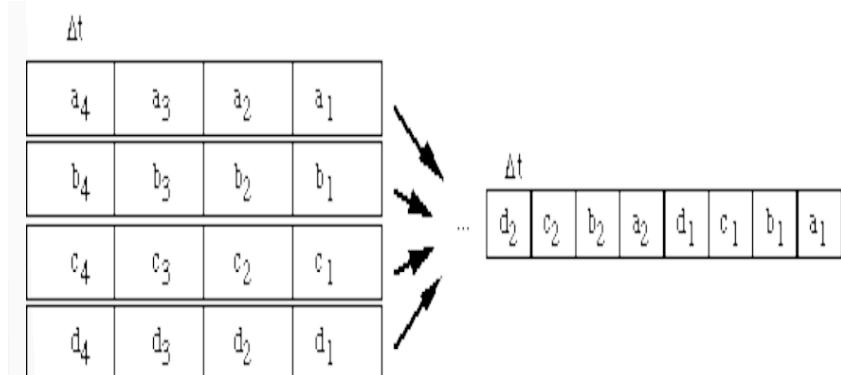
Abbiamo visto quindi alcuni standard, ma ne esistono, per FDM, fino a 230000 canali.

Wavelength Division Multiplexing (WDM)

Il WDM è una variante del FDM **utilizzata per le fibre ottiche**. In particolare, i **raggi di bande differenti, provenienti da più fibre, convergono in un prisma** che li combina inoltrandoli in un'unica fibra. All'altro capo, attraverso un prisma o un sintonizzatore ottico, il raggio viene diviso tra le fibre destinate, **selezionando, per ciascuna delle fibre, la lunghezza d'onda del raggio che in essa si intende inoltrare**.

Time Division Multiplexing (TDM)

Il TDM è un tipo di multiplexing adatto per la **gestione di dati in forma digitale**, nel quale i **bit provenienti da diverse connessioni vengono a turno prelevati ed inviati su un'unica connessione ad alta velocità**.



Nell'ambito del TDM viene utilizzata la tecnica del **Pulse Code Modulation (PCM)**, nella quale il **codec trasforma il segnale analogico**, proveniente dal local loop, in **digitale**. Visto che il **ritmo di funzionamento di tutti i sistemi telefonici** è di 125 μs , si avranno 8000 campioni al secondo. Nell'ambito del TDM **non esiste uno standard internazionale**, infatti in **America e in Giappone** si ha il **T1 Carrier** (anche T2, T3 e T4), dove ogni campione viene rappresentato da **7 bit**, mentre in **Europa** si ha l'**E1 Carrier** (anche E2, E3, E4 e E5), dove ogni campione viene rappresentato da **8 bit**.

Commutazione di circuito e di pacchetto

Nei **sistemi commutati**, in base alle richieste, le linee in ingresso vengono di volta in volta connesse a differenti linee in uscita tramite la **commutazione di circuito (circuit switching)**. Quest'ultima, in base al **sender**, al **receiver** e alla **disponibilità del momento**, stabilisce una connessione fisica temporanea attraverso cui far fluire la comunicazione (**connection setup**). Nei sistemi telefonici, la connessione è data da un local loop più una successione di canali a 64Kb ricavati all'interno di flussi T1 o T3. Con l'avvento delle tecnologie digitali la commutazione di circuito è stata spesso sostituita dalla **commutazione di pacchetto (packet switching)**, nella quale i **dati, suddivisi appunto in pacchetti, vengono inviati indipendentemente gli uni dagli altri ed una volta giunti a destinazione vengono poi raccolti e riordinati**. La commutazione di pacchetto **evita lo spreco di risorse di rete in assenza di traffico**, in quanto non viene creata alcuna connessione fisica dedicata tra mittente e destinatario, la quale porterebbe ad un spreco di risorse quando non vi è alcuna attività di comunicazione tra i due. Tramite la commutazione di pacchetto, inoltre, si **evita di perdere tempo per la fase di connection setup**, ma **sono probabili i problemi di congestione della rete**, ossia quando la domanda di larghezza di banda o risorse di rete supera la capacità disponibile della rete stessa.

Rilevamento errori nei livelli fisici

Il livello fisico [data link](#) **consente comunicazioni affidabili ed efficienti tra due macchine adiacenti**, ossia connesse da un [cavo coassiale](#), da un [doppino](#), o altri mezzi di trasmissione. I protocolli di questo livello devono tener conto del fatto che ci sono **errori e disturbi occasionali**, ma anche che il canale ha un **data rate finito** e che c'è un **ritardo nella propagazione**. Quando attraverso un canale (ad es. una linea fisica) arrivano dei bit alla scheda di rete di un router, l'hw della scheda di rete, preposto alle operazioni del livello 1 (fisico), li rileva e li passa alla sezione hw/sw preposta alle operazioni del livello 2 (data link), che si trova sempre sulla scheda di rete. La sezione hw/sw del livello 2 effettua i **controlli di framing, errori di trasmissione e numero di frame** e nel caso in cui non ci siano errori genera un interrupt alla CPU del router. L'interrupt attiva e passa il frame arrivato ad un **processo di sistema**, di cui si occupa la sezione sw del livello 3. Il processo ricostruisce il pacchetto ed in base all'indirizzo del destinatario, contenuto nel pacchetto, determina su quale linea rimetterlo in uscita. Dopo aver fatto ciò, restituisce il pacchetto alla sezione hw/sw del livello 2, la quale lo mette in un nuovo frame e lo consegna al sottostante livello fisico, che provvederà ad inoltrarlo sulla linea prescelta. Il livello 1 accetta un flusso di **bit grezzi** e cerca di farli arrivare a destinazione, ma purtroppo **il flusso non è esente da errori**, infatti possono arrivare più o meno bit di quelli effettivamente inviati. E' compito del livello 2 rilevare e, se possibile correggere, tali errori.

Operazioni del livello 2

Le operazioni del livello 2, per quanto riguarda la **trasmissione**, sono:

- suddividere il flusso di bit che arriva dal livello 3 in una serie di frame;
- calcolare una funzione checksum per ciascun frame, che viene utilizzata per verificare l'integrità dei dati trasmessi;
- inserire il checksum nel frame;
- consegnare il frame al livello 1, il quale lo spedirà come sequenza di bit.

Per quanto riguarda invece la **ricezione**, le operazioni del livello 2 sono:

- ricevere una sequenza di bit dal livello 1;
- ricostruire da questa un frame dopo l'altro;
- per ciascun frame ricalcolare il checksum;
- controlla che il checksum ricalcolato sia uguale a quello contenuto nel frame e in caso negativo scarta il frame.

Struttura dei frame

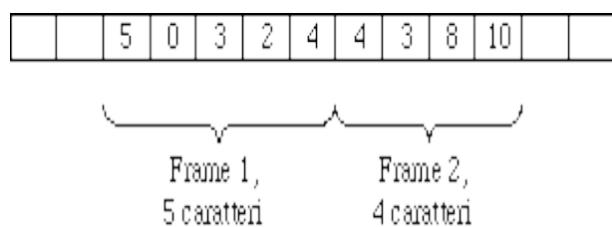
Per delimitare i frame è rischioso utilizzare lo spazio temporale che li separa, questo perché di solito nelle reti è difficile garantire una perfetta temporizzazione.

Per indicare l'inizio e la fine del frame si possono utilizzare diversi metodi
come:

- conteggio dei caratteri;
- caratteri di inizio e fine (character stuffing);
- bit pattern di inizio e fine (bit stuffing);
- violazioni della codifica del livello fisico.

Conteggio dei caratteri

Un campo dell'header indica quanti sono i caratteri del frame.



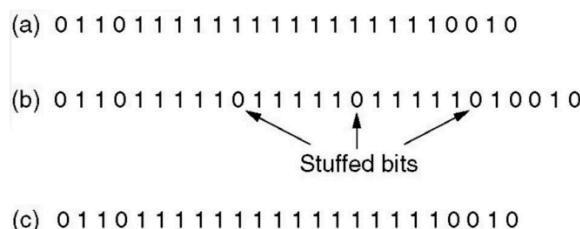
Eventuali errori o perdita di tale campo rendono però impossibile l'individuazione dell'inizio del prossimo frame e dei successivi, di conseguenza questo metodo viene utilizzato poco a causa della scarsa affidabilità.

Character stuffing

Ogni frame inizia e finisce con una particolare sequenza di caratteri ASCII detta **Data Link Escape (DLE)**. La **sequenza di inizio frame è DLE STX (Start of TeXt)**, mentre la **sequenza di fine è DLE ETX (End of TeXt)**. Nel caso in cui si perda traccia dei confini di un frame, la si riacquista all'arrivo della prossima coppia di DLE STX – DLE ETX. Poiché il byte corrispondente alla codifica dei DLE potrebbe coincidere con byte del frame da trasmettere, il data link aggiunge, per ciascuno di tali byte, **un altro DLE**, di conseguenza solo i singoli DLE segnano i confini dei frame. Una volta arrivati a destinazione, il **data link rimuove tutti i DLE** prima di consegnare i frame al livello 3.

Bit stuffing

Ogni frame inizia e finisce con una specifica sequenza di bit, chiamata **flag byte**. Poiché il flag byte può far parte dei dati che devono essere trasmessi, il data link provvede ad inserire specifici bit che saranno poi rimossi all'arrivo. Se prendiamo come esempio la figura sottostante, vedremo che quando il data link incontra 5 bit 1 consecutivi inserisce uno 0 aggiuntivo; in questo modo il flag byte può apparire solo all'inizio ed alla fine dei frame. Coerentemente, quando a destinazione compaiono 5 bit uguali ad 1, il data link rimuove lo 0 che li segue.



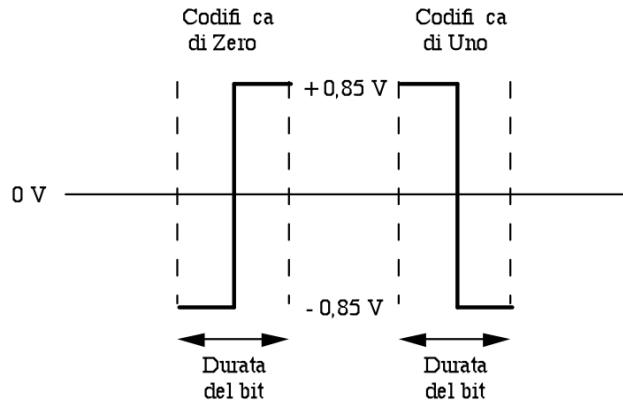
- a) Dati originali
- b) Dati trasmessi sul canale (stuffed)
- c) Dati ricevuti dopo il de-stuffing

Violazioni della codifica del livello fisico

Per motivi fisici in molte reti, soprattutto LAN, i **bit si codificano con una certa ridondanza**. Ne è un esempio la **Manchester encoding** (IEEE 802.3), dove il **bit dati 1 viene rappresentato come coppia di bit fisici high/low**, mentre il **bit dati 0 viene rappresentato come coppia di bit fisici low/high**. Visto che le **coppie low/low e high/high** non sono utilizzate, possono essere **impiegate per delimitare i frame**. La Manchester encoding prevede **una transizione** (da high a low o da low a high) **per ogni bit dato**.

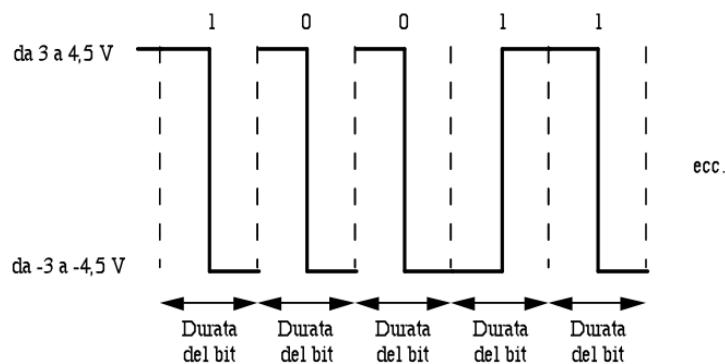
Manchester encoding (e differential Manchester encoding)

La codifica Manchester prevede una **transizione del valore del segnale nel mezzo di ogni bit**, qualsiasi sia il valore (0 o 1) che assume.

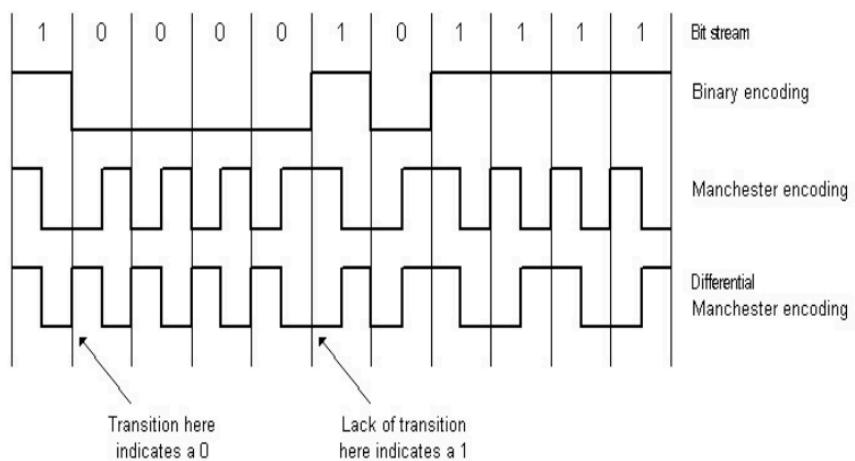


Livelli standard IEEE 802.3: **-0,85, +0,85**

Esiste poi una variante della Manchester encoding, la **differential Manchester encoding** che prevede una transizione all'inizio del bit per valore del segnale 0, nessuna transizione all'inizio del bit per valore del segnale 1 e transizione a metà del bit sia per valore 0 che per valore 1.



Data Link Bit Encoding



Nei modelli carrier sensitive che fanno uso della Manchester encoding, l'assenza della **portante** (la portante è il segnale principale che trasporta i dati digitali) può

essere codificata con segnale nullo. In generale, con la Manchester encoding, si ha una **facile sincronizzazione tra mittente e destinatario** ed è **semplice anche rilevare le collisioni**. Il codice è bilanciato, cioè vi è uguale energia per 0 e 1, e quindi la trasmissione dati, anche se ne genera diverse quantità (di 0 e 1), non produce componenti in corrente continua, poiché dannose nella trasmissione dei segnali. A parità di velocità di trasmissione, la **codifica Manchester richiede una banda doppia rispetto alla codifica diretta**, in quanto ogni bit richiede la trasmissione di due valori distinti.

Gestione degli errori

Molti fattori possono provocare errori, soprattutto sul local loop, nelle trasmissioni wireless ed in generale nei mezzi di trasmissione meno moderni. Gli errori sono dovuti solitamente a **rumori di fondo, disturbi improvvisi** (es. fulmini) o **interferenze** (es. motori elettrici). Esistono **due approcci al trattamento degli errori**:

- **rilevazione dell'errore**, che consiste nell'includere informazione aggiuntiva, in modo da accorgersi che c'è stato un errore;
- **correzione dell'errore**, che consiste nell'includere abbastanza informazione aggiuntiva in modo da poter ricostruire, in caso di errori, il messaggio originario.

Distanza di Hamming, codeword e codice

Un frame, a parte i delimitatori, consiste di $n = m + r$ bit, con m **il numero di bit di messaggio vero e proprio** e r **il numero di bit ridondanti (redundant bit o check bit)**. La sequenza di $m + r$ bit prende il nome di **codeword**, ossia parola codice. Attraverso una **XOR bit a bit** tra due codeword è possibile determinare **la loro distanza di Hamming**, ossia il numero di bit in cui esse differiscono.

Se due codeword hanno una distanza di Hamming uguale a d , allora ci vogliono esattamente d errori su singoli bit per trasformare l'una nell'altra. Un **insieme prefissato di codeword prende il nome di codice (code)** e **la distanza di Hamming di un codice è il minimo delle distanze di Hamming tra tutte le possibili coppie di codeword del codice**. Per rilevare d errori serve un codice con distanza di Hamming $d + 1$. Qualunque combinazione di d errori non riesce a trasformare una codeword valida in un'altra codeword valida. Per correggere d errori serve un codice con distanza di Hamming $2d + 1$. Una codeword contenente fino a d errori è più vicina a quella originaria rispetto a qualunque altra codeword valida. A seconda degli scopi, ed in funzione di m , si progetta un **algoritmo per il calcolo degli r check bit**, in modo che le codeword risultanti di $n = m + r$ bit costituiscano un codice con la desiderata distanza di Hamming.

Codice di parità

$\leftarrow m \rightarrow r$	Si contano il numero di bit 1 nella codeword!
1011 0101 1	Dispari: 1
1000 0111 0	Pari: 0

Il codice di parità (parity code) ha distanza di Hamming uguale a 2, in quanto per ottenere una codeword valida devono cambiare 2 bit. Questo perché ogni singolo errore produce un numero dispari di 1 e quindi una parola codice non valida.

Esempio di distanza di Hamming

Si considerino le seguenti codeword a distanza 5:

00000	00000
00000	11111
11111	00000
11111	11111

Se si invia la codeword **00000 11111**
ma arriva: **00000 00111** (2 errori)
la codeword più vicina è: **00000 11111**

Se invece arriva: **00000 00011** (tre errori)

la codeword più vicina è: **00000 00000**

Correzione degli errori

Per correggere un singolo errore su m bit si devono impiegare almeno r check bit, con r tale che $2^r \geq (m + r + 1)$. Se le codeword legali sono 2^m , poiché per ogni codeword di $n = m + r$ bit abbiamo $n + 1$ codeword a distanza 1, il numero di bit n deve essere tale che

$2^n = 2^{m+r} > 2^m * (n + 1) \Rightarrow 2^r > (n + 1) = (m + r + 1)$. Per correggere un singolo errore su m bit, si devono impiegare almeno r check bit, con r tale che $2^r \geq (m + r + 1)$, di conseguenza occorrono circa n check bit. Se le informazioni che intendiamo trasmettere sono caratteri ASCII, quindi $m = 7$, poiché il numero r di redundant bit deve essere tale che

$$2^r \geq (m + r + 1) \geq (7 + r + 1)$$

segue $r = 4 \rightarrow n = m + r = 7 + 4 = 11$. Occorre un algoritmo che, in base alla posizione ed al valore dei 4 bit di controllo, consenta la correzione del bit errato (ad es. si dispongono gli r redundant bit nelle posizioni 2^i con $0 \leq i < r$).

Consideriamo un carattere ASCII ed ipotizziamo che nella trasmissione si verifichi un errore. Cioè inviamo (A): 1000001, ed arriva (Q) : 1010001

$$\begin{array}{ll} 11=8+2+1; & 10=8+2; \\ 7=4+2+1; & 6=4+2; \\ 3=2+1; & \end{array} \quad \begin{array}{ll} 9=8+1; & 8^*=8; \\ 5=4+1; & 4^*=4; \\ 2^*=2; & 1^*=1; \end{array}$$

il bit di parità **2*** è calcolato sui bit di posto: 11, 10, 7, 6, 3, 2

1*	2*	3	4*	5	6	7	8*	9	10	11
0	0	1	0	0	0	0	1	0	0	1

+ -> Addizione senza riporto == XOR

Inviato (A) è arrivato (Q)

1	2	3	4	5	6	7	8	9	10	11
*	*		*				*			
0	0	1	0	0	0	0	1	0	0	1

0	0	1	0	0	0	0	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---

$$\begin{array}{ll} 11=8+2+1; & 10=8+2; \\ 7=4+2+1; & 6=4+2; \\ 3=2+1; & \end{array} \quad \begin{array}{ll} 9=8+1; & 8=8; \\ 5=4+1; & 4=4; \\ 2=2; & 1=1; \end{array}$$

Risultano errati i bit di parità di posto **2*** e **4***, quindi il bit errato è quello di posto 6=2+4

Rilevamento e correzione di burst di errori

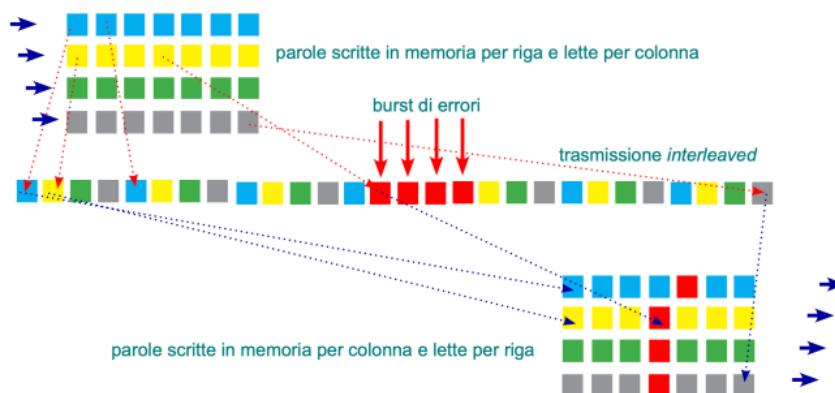
Per rilevare burst di errori, ossia gruppi contigui di errori, di lunghezza massima k si utilizzano i **seguenti passi**:

- si accumulano k codeword di n bit secondo una matrice di k righe e n colonne;
- si aggiunge il bit di parità per ciascuna riga, formando una colonna aggiuntiva;
- i dati vengono trasmessi per colonne;
- ricevuto l'intero blocco si controllano tutti i bit di parità e, in caso di errori, si richiede la **ritrasmissione del blocco**.

Per correggere burst di errori di lunghezza massima prefissata k si utilizzano i **seguenti passi**:

- si accumulano k codeword, riga per riga;
- le codeword si trasmettono per colonne;
- quando arrivano vengono riassemblate per righe.

Poiché un burst di k errori comporta un singolo errore in ciascuna delle k codeword, correggendo ciascuna codeword ricostruiamo l'intero burst.



Rilevamento e correzione di errori nella pratica

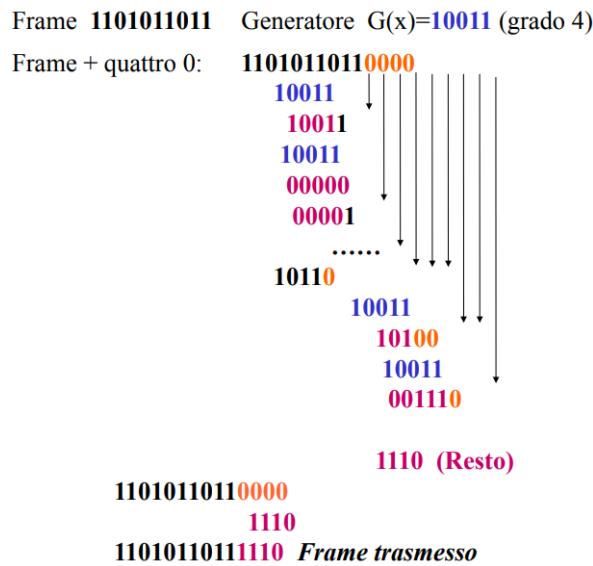
I codici di correzione di errore, ad eccezione delle trasmissioni simplex dove non è possibile inviare al mittente una richiesta di ritrasmissione, sono **utilizzati raramente**. E' infatti **più efficiente limitarsi a rilevare gli errori, ritrasmettendo saltuariamente i dati errati**, piuttosto che impiegare un codice per la correzione degli errori o anche i bit di parità, i quali risultano decisamente più dispendiosi in termini di ridondanza.

Cyclic Redundancy Code (CRC)

Nella pratica viene quasi sempre utilizzato il **Cyclic Redundancy Code (CRC)** o **polynomial code**, dove un numero di m bit corrisponde ad un polinomio di grado $m - 1$. Un esempio di questa tecnica è il polinomio $x^3 + x^2 + x^0$ che corrisponde alla stringa di bit 1101. L'**aritmetica polinomiale utilizzata per il calcolo del CRC è sviluppata modulo 2**, ed in particolare abbiamo addizione senza riporto e sottrazione senza prestito (equivalenti allo **XOR**) ma anche divisione calcolata attraverso la sottrazione modulo 2. Nel calcolo del CRC **mittente e destinatario adottano lo stesso polinomio generatore $G(x)$** , dove **il bit più significativo e quello meno significativo devono essere entrambi 1**. Se r è il grado di $G(x)$ ed m è il numero di bit del frame $M(x)$, m deve essere maggiore di r , di conseguenza $M(x)$ deve essere più lungo di $G(x)$. **In coda al frame si aggiunge un checksum tale da originare un polinomio formato da frame + checksum di grado $m + r$** che sia divisibile per $G(x)$. Quando il ricevitore riceve il polinomio, lo divide per $G(x)$ e **se il risultato è 0 significa che l'operazione è andata a buon fine**, altrimenti significa che c'è stato un errore. Per effettuare il **calcolo del checksum si seguono alcuni passi**:

- si aggiungono r bit 0 a destra del frame, ottenendo $M(x)x^r$ di $m + r$ bit;
- si divide $M(x)x^r$ per $G(x)$;
- si ottiene il frame da trasmettere sottraendo ad $M(x)x^r$ il resto della divisione.

La sottrazione (XOR) fatta sugli r bit meno significativi non modifica il frame, il quale risulta comunque divisibile per $G(x)$ vista la costruzione.

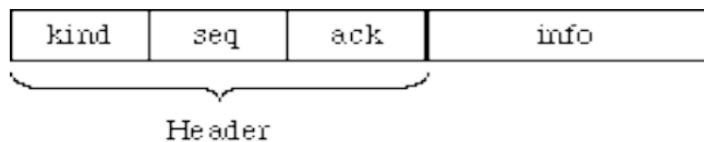


Un codice polinomiale con r bit rileva tutti gli errori singoli e doppi, tutti gli errori di x bit, con x dispari, e tutti i burst di errori di lunghezza $\leq r$. Ci sono alcuni polinomi divenuti addirittura standard internazionali:

- CRC-12, $x^{12} + x^{11} + x^3 + x^2 + x^1 + 1$;
- CRC-16, $x^{16} + x^{15} + x^2 + 1$;
- CRC-CCITT, $x^{16} + x^{12} + x^5 + 1$.

Un checksum a 16 bit rileva errori singoli e doppi, errori di numero dispari di bit, burst di errori di lunghezza ≤ 16 , 99.997% di burst di errori di lunghezza 17 ed il 99.998% di burst di errori di lunghezza 18. Tutto ciò è vero nel caso in cui gli m bit del messaggio siano distribuiti casualmente, il che è poco probabile e quindi i burst di lunghezza 17 e 18 sfuggono più spesso di quanto si creda.

Struttura di un frame



Un frame è caratterizzato da un **header**, diviso nelle sezioni kind, seq e ack, e dal **pacchetto vero e proprio**, memorizzato nel campo info. Il campo **kind** specifica il tipo del frame, ossia se si tratta di frame dati o frame di controllo ad es., il campo **seq** contiene il numero, progressivo, del frame, il campo **ack** ovviamente riguarda informazioni legate all'acknowledgement, mentre il campo **info** contiene il pacchetto di livello network completo.

Protocolli di comunicazione

Protocollo Heaven

Alla base del protocollo Heaven vi sono **alcune ipotesi**, non molto realistiche, a partire dal fatto che si utilizzano i **canali simplex**, e quindi di conseguenza i **frame vengono trasmessi in una sola direzione**. Secondo il protocollo i **livelli network non devono mai attendere per inviare e/o ricevere al/dal livello data link**, il **tempo di elaborazione del livello data link è nullo**, il **ricevitore ha un buffer infinito** e il **canale fisico è esente da errori**.

Mittente (loop infinito):

- 1) attende un pacchetto dal livello network;
- 2) costruisce un frame dati;
- 3) passa il frame al livello fisico;
- 4) torna ad 1).

Destinatario (loop infinito):

- 1) attende l'arrivo di un frame da livello fisico;
- 2) estrae il pacchetto;
- 3) lo passa al livello network;
- 4) torna ad 1).

Protocollo stop and wait

Il protocollo stop and wait eredita la maggior parte delle ipotesi del protocollo Heaven, con la differenza che si considera un **ricevitore senza buffer infinito**. In questo caso, quindi, il **mittente deve essere opportunamente rallentato**. Una possibilità sarebbe quella di **ritardare le trasmissioni**, effettuandole ogni Δt , con Δt prefissato e calibrato sul caso peggiore, oppure **far inviare al mittente il frame successivo solo se esplicitamente autorizzato dal destinatario**. Visto che la prima soluzione sarebbe troppo gravosa, **si può utilizzare la seconda**, che è alla **base del protocollo stop and wait**.

Mittente (loop infinito):

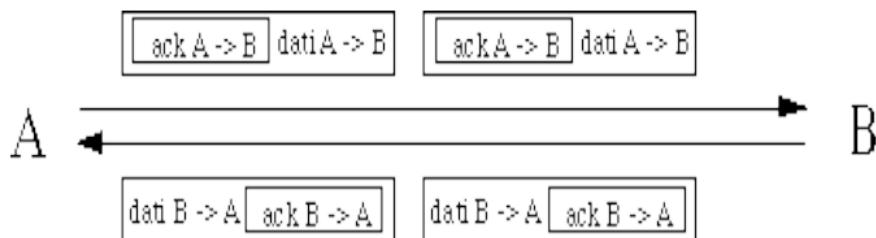
- 1) attende l'arrivo di un pacchetto dal livello network;
- 2) costruisce un frame dati;
- 3) passa il frame al livello fisico;
- 4) attende l'ack;
- 5) torna ad 1).

Destinatario (loop infinito):

- 1) attende l'arrivo di un frame dati da livello fisico;
- 2) estraе il pacchetto;
- 3) consegna il pacchetto al livello network;
- 4) invia un frame di ack al mittente;
- 5) torna ad 1).

Piggybacking

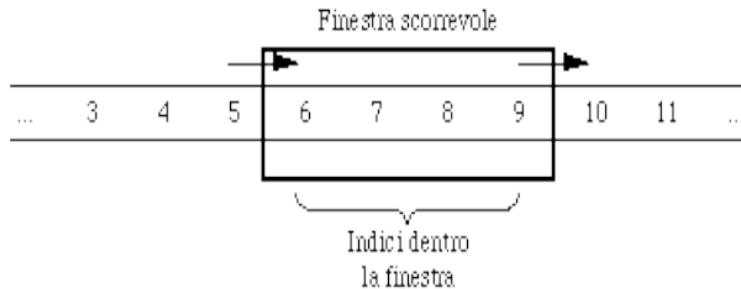
Tramite la tecnica del piggybacking, quando B deve inviare un ack ad A, lo inserisce (in un apposito campo) nel prossimo frame dati che B deve inviare ad A. Quindi la **tecnica del piggybacking permette di includere informazioni aggiuntive in un pacchetto, sfruttandone uno già esistente.**



Per inviare un ack si attende quindi un frame su cui trasportarlo ed in particolare si stabilisce un **tempo di attesa limite**, alla fine del quale viene creato un apposito frame di ack.

Protocolli sliding window (finestra scorrevole)

In questo tipo di protocolli **ad ogni frame viene assegnato un numero di sequenza** compreso tra 0 e $2^n - 1$, in quanto il campo seq è di n bit. Il mittente fa scorrere una **finestra sulla sequenza di indici**, con questi ultimi (indici) che corrispondono ai frame da spedire, o spediti, ma non ancora confermati. **Quando arriva un ack**, il corrispondente indice esce dalla finestra, mentre ad ogni pacchetto in arrivo dal livello network vengono assegnati nuovi indici. **I frame spediti ma non confermati** devono essere mantenuti in memoria per la possibile ritrasmissione.

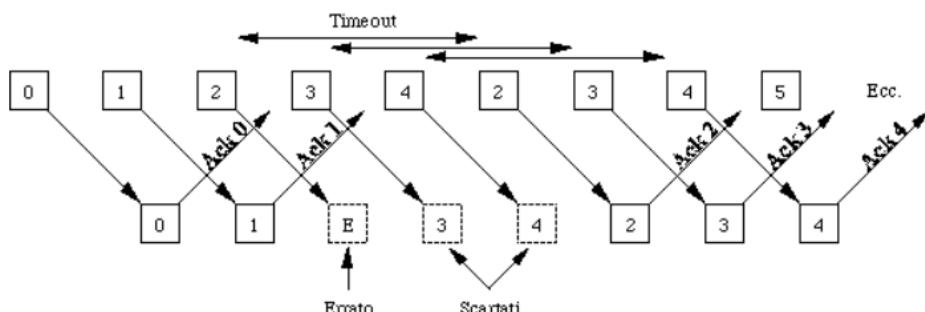


Se il buffer è pieno il livello data link deve costringere il livello network a sospendere consegna di pacchetti. **Il destinatario mantiene una finestra** corrispondente agli indici dei frame che possono essere accettati. **Quando arriva un frame il cui indice è fuori** dalla finestra viene scartato, di conseguenza non invia il relativo ack, mentre **se invece arriva un frame il cui indice è all'interno** della finestra, il frame viene accettato, viene spedito il relativo ack e la finestra viene spostata in avanti. **La finestra del destinatario rimane sempre della stessa dimensione**: se è pari a 1 il destinatario accetta un solo frame e quindi gli stessi (frame) devono arrivare nell'ordine giusto. Le finestre di mittente e destinatario non devono necessariamente avere uguali dimensioni né uguali limiti inferiori o superiori.

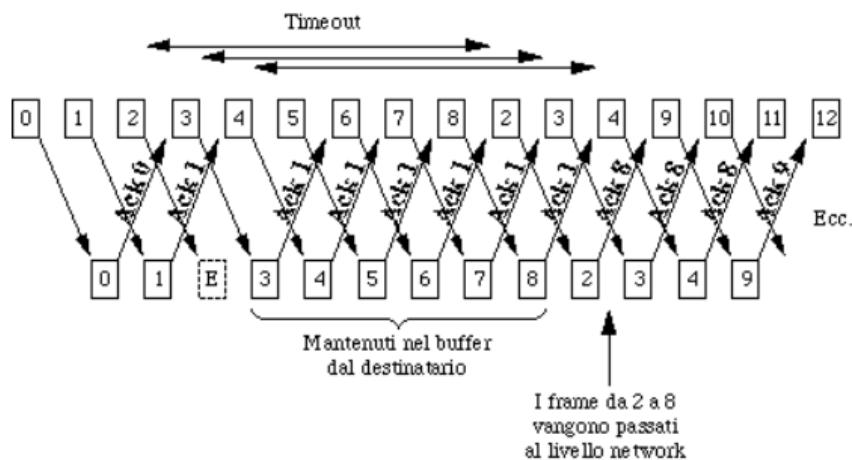
Protocolli go-back-n e selective repeat

Se il **round-trip time** (tempo di andata e ritorno) del segnale è alto, come ad esempio nei canali satellitari, i protocolli stop-and-wait risultano altamente inefficienti, in quanto viene sprecato molto tempo in attesa degli ack. Per migliorare le cose si può consentire l'invio di un certo numero di frame anche senza aver ricevuto l'ack del primo. Questa tecnica prende il nome di **pipelining**.

Questa soluzione però porta un problema, infatti se un frame nel mezzo della sequenza subisce alterazioni, quindi se si perde o va scartato, il mittente invia molti altri frame prima di riceverne notizia. Il primo approccio al problema è quello del **protocollo go-back-n**, dove **se arriva un frame danneggiato o con un numero di sequenza non progressivo**, il destinatario ignora tale frame e tutti i successivi, non inviando i relativi ack. In pratica vengono accettati i frame solo nell'ordine giusto. Quando il mittente va in time-out sul frame sbagliato, e su tutti quelli successivi, provvederà a ritrasmettere la sequenza di frame a partire da quello per il quale si è verificato il time-out.



In questo approccio il mittente deve mantenere in un apposito buffer tutti i frame non confermati. Se il buffer si riempie, il mittente deve bloccare il livello network fino a che non vi sarà di nuovo spazio. Quindi se il tasso d'errore è alto e/o il time-out è lungo, vi sarà **spreco di banda**. Un altro approccio, più efficiente, è il **protocollo selective repeat**, dove per ogni frame ricevuto correttamente, il destinatario invia un ack con il numero più alto della sequenza completa arrivata fino a quel momento. In caso di frame errato, il destinatario sospende la trasmissione di ack ma continua la ricezione dei frame successivi, che mantiene nel proprio buffer, e appena riceve nuovamente quel frame (ovviamente senza errori) lo consegna, insieme a tutti i frame mantenuti nel buffer, al livello network. Quando si verifica un timeout il mittente rispedisce il frame corrispondente all'indice su cui è andato in timeout.



In questo protocollo, **mittente e destinatario devono entrambi gestire un buffer**: il primo per contenere i frame non confermati ed il secondo per contenere i frame successivi ad un errore. In questo modo si ha un **basso spreco di banda**, il quale può ulteriormente diminuire attraverso l'utilizzo di nack (negative ack), da impiegare quando arriva un frame danneggiato o diverso da quello atteso. Sia per il protocollo go-back-n che per il selective repeat è necessaria la gestione di **timer multipli**, cioè uno per ogni frame inviato e non confermato, ed inoltre il ricevente, per inviare gli ack, utilizza la tecnica del piggybacking, ove possibile, altrimenti invia un apposito frame.

Protocolli data link

I protocolli data link attualmente più diffusi sono:

- **HDLC** (standard ISO);
- **SLIP** (architettura TCP/IP);
- **PPP** (successore dello SLIP).

Tutti e tre **discendono dal protocollo SDLC** (Synchronous Data Link Control), nato nell'ambito dell'architettura SNA (IBM).

High Level Data Link Control (HDLC)

L'HDLC è un protocollo **bit oriented**, quindi **utilizza la tecnica del bit stuffing**.

Bit:	8	8	8	≥ 0	16	8
	01111110	Address	Control	Dati	Checksum	01111110

Figura 3-12: Frame HDLC

In un frame HDLC abbiamo **diversi campi**:

- **Address**, utilizzato nelle linee multipunto, dove identifica i diversi terminali, (necessario per il dialogo tra un concentratore e diversi terminali);
- **Control**, contiene i **numeri di sequenza, di ack, ecc.**;
- **Dati**, contiene i dati da trasportare;
- **Checksum**, calcolata con lo standard [CRC-CCITT](#).

Questo protocollo utilizza una **finestra scorrevole con numeri di sequenza a 3 bit**, contenuti nel campo **Seq**, a sua volta contenuto nel campo Control ed utilizza il campo **Next**, anch'esso contenuto nel campo Control, per la tecnica del piggybacking. **L'HDLC prevede tre tipi di frame, identificati dai primi due bit del campo Control**:

- **Information**, per la trasmissione dati;
- **Supervisory**, per comandare diverse modalità di ritrasmissione;
- **Unnumbered**, dove, come suggerisce il nome, manca il numero di sequenza, ed infatti **vengono utilizzati con finalità di controllo o per trasportare il traffico di connessioni non affidabili**.

Serial Line IP (SLIP)

Il SLIP fu presentato nel 1984, infatti è il **primo protocollo di livello data link**, e fu proposto per collegare via modem macchine Sun, che utilizzavano TCP/IP, ad Internet. Questo protocollo **utilizza character stuffing ma ha diverse limitazioni**, infatti **non vi è controllo degli errori, supporta solo IP statici e non è uno standard ufficiale di Internet**.

Point to Point Protocol (PPP)

Il PPP è un **protocollo adatto sia a connessioni telefoniche che a linee router-router e fornisce diverse funzionalità**:

- framing;
- rilevamento degli errori;
- protocollo di controllo per attivare, testare e disattivare la linea (**Link Control Protocol – LCP**);
- supporto di molteplici protocolli di livello network;

- una famiglia di protocolli per negoziare opzioni di livello network (**Network Control Protocol – NCP**).

Per quanto riguarda l'ultimo punto, per ogni livello network supportato vi è un differente NCP, ad esempio, nel caso di IP, il NCP viene utilizzato per negoziare un indirizzo IP dinamico. Il traffico prodotto, nelle fasi iniziali e finali, dai protocolli LCP e NCP viene trasportato all'interno dei frame PPP. Il protocollo è **modellato sull'HDLC**, ma essendo character-oriented utilizza il **character stuffing** (quindi i frame sono costituiti da un numero intero di byte) ed inoltre c'è un campo apposito per il supporto multiprotocollo offerto al livello network.

Byte:	1	1	1	1	Variabile	2 oppure 4	1
	Flag 01111110	Address 11111111	Control 00000011	Protocol	Dati	Checksum	Flag 01111110

Figura 3-13: Frame PPP

In un frame PPP abbiamo **diversi campi**:

- **flag**, come in HDLC (dove non aveva nome ma erano il campo iniziale e finale anche in quel caso);
- **address**, che sarà sempre 11111111, poiché di fatto non ci sono indirizzi, in quanto non c'è più l'idea di gestire le linee multipunto;
- **control**, il cui default 00000011 indica un unnumbered frame, quindi relativo ad un servizio non affidabile;
- **protocol**, indica il protocollo relativo al pacchetto che si trova nel payload (LCP, NCP, IP, ecc.);
- **payload** (corrispettivo del campo dati dell'HDLC), è di lunghezza variabile e negoziabile, ma il default è di 1500 byte;
- **checksum**, solitamente di 2 byte, oppure di 4.

MAC (Medium Access Control) e modulazione

Il MAC è un sottolivello del livello data link. Nelle **reti broadcast** il problema principale è decidere a quale elaboratore (stazione) assegnare il mezzo trasmisivo in caso di più richieste simultanee. In particolare è compito dei protocolli del sottolivello MAC del livello data link decidere **chi deve essere il prossimo a trasmettere su un canale broadcast**. I protocolli MAC vengono utilizzati soprattutto nelle reti broadcast, come le LAN e le WAN basate su satellite, ed assegnano i canali attraverso un'**allocazione statica**, quindi decisa in anticipo, o **dinamica**, nel caso in cui l'allocazione venga stabilita in base alle esigenze del momento.

Allocazione statica

L'allocazione statica prevede la **suddivisione del canale tra gli N utenti**, ciascuno dei quali riceve una **frazione della banda totale**. Si può utilizzare questo tipo di allocazione in tecniche come il [FDM](#), allocando a ciascun utente una banda di frequenze diversa da quella degli altri utenti. Ciò va bene nel caso in cui il numero di utenti non varia rapidamente e se tutti trasmettono con [data rate](#) più o meno costante. Nella realtà dei fatti, però, spesso questa tecnica di allocazione porta a **spreco di banda** quando uno o più utenti non trasmettono, e poiché il **traffico in generale è molto bursty** (flusso di dati caratterizzato da brevi periodi di intensa attività, ossia bursty, contrapposti a periodi di inattività) non si riescono a gestire in modo adeguato i picchi che si verificano.

Allocazione dinamica

L'allocazione dinamica, come abbiamo detto, **cerca di adeguarsi alle esigenze trasmissive**, in modo da soddisfarle al meglio. L'allocazione dinamica **si basa su un modello**:

- **a stazioni**, ossia ci sono N stazioni indipendenti che generano frame da trasmettere. In particolare una stazione che ha generato un frame si blocca finché non viene trasmesso;
- **a singolo canale**, cioè tutte le stazioni possono trasmettere e ricevere frame da un singolo canale;
- **con collisioni**, ossia se due frame vengono trasmessi contemporaneamente, il segnale risultante è rovinato, poiché appunto si ha una collisione, ed i frame devono essere ritrasmessi.

Abbiamo poi due concetti, il tempo e l'ascolto del canale, che possono essere gestiti in vari modi. Partendo dal **tempo**, si può avere **continuous time**, dove la trasmissione di un frame può iniziare in qualsiasi istante, o uno **slotted time**, dove il tempo è diviso in intervalli discreti (slot) e la trasmissione può iniziare solo all'inizio di uno slot. In un slot si possono avere zero (slot vuoto), uno (slot con frame) o più frame, ed in quest'ultimo caso si verificano collisioni. Per quanto riguarda il concetto dell'**ascolto del canale**, invece, abbiamo **carrier sense** (tipico delle LAN), dove le stazioni, prima di trasmettere, ascoltano il canale e la trasmissione ha inizio solo se il canale non è occupato, o **no carrier sense** (tipico dei canali via satellite nei quali vi è un elevato [round-trip time](#)), dove le stazioni trasmettono comunque, senza ascoltare, e solo dopo si preoccupano di rilevare eventuali collisioni.

Protocolli Carries Sense Multiple Access (CSMA)

Per **migliorare il throughput**, nelle reti locali, le stazioni possono ascoltare il canale e regolarsi di conseguenza, ottenendo un'efficienza molto più alta.

I protocolli nei quali le stazioni ascoltano il canale prima di iniziare a trasmettere si dicono carrier sense. Ci sono vari tipi di **protocolli carrier sense**:

- **1-persistent:** quando una stazione deve trasmettere, ascolta il canale e, **se è occupato**, aspetta finchè si libera per poi trasmettere. Nel caso in cui invece il canale sia **libero** trasmette con probabilità 1 (ossia trasmette senza alcun problema). **Se avviene una collisione** la stazione aspetta un tempo random e riprova tutto da capo. Si ha una collisione se una stazione A trasmette e prima che il suo segnale arrivi a B, anche quest'ultimo inizia a trasmettere. Si ha una collisione anche se A e B ascoltano contemporaneamente durante la trasmissione di C e non appena quest'ultima termina, iniziano entrambe a trasmettere. Più è alto il tempo di propagazione tra A e B, più è alta la probabilità di collisioni;
- **Non-persistent:** quando una stazione deve trasmettere, ascolta il canale e, **se è occupato**, attende che si liberi, aspettando un tempo random prima di ripetere tutto il procedimento da capo (a differenza del 1-persistent che trasmetteva appena si liberava il canale). Nel caso in cui invece sia **libero**, trasmette. Ci si aspettano **meno collisioni**, rispetto al 1-persistent, ma **maggior ritardi**, prima di riuscire a trasmettere un frame;
- **P-persistent (per canali slotted):** quando una stazione deve trasmettere, ascolta il canale e, **se occupato**, aspetta lo slot successivo e ricomincia da capo. Nel caso in cui invece sia **libero**, con probabilità p trasmette subito, mentre con probabilità $1 - p$ aspetta il prossimo slot, riapplicando tale procedimento. **In caso di collisione** la stazione aspetta un tempo random e ricomincia da capo. Al diminuire di p ci si aspettano crescenti ritardi ed una progressiva diminuzione delle collisioni prima di riuscire a trasmettere un frame.

Protocolli Carries Sense Multiple Access with Collision Detection (CSMA/CD)

Un **ulteriore miglioramento del throughput** si ha se le stazioni interrompono la loro trasmissione, non appena rilevano una collisione, invece di portarla a termine. **Per rilevare le collisioni** le stazioni ascoltano il canale durante le proprie trasmissioni. Se la potenza del segnale ricevuto (analogico) è superiore a quella trasmessa, **significa che si è verificata una collisione**. Quando se ne verifica una (collisione), la stazione ritenta la trasmissione dopo un intervallo di tempo casuale. Posto uguale a T il tempo di propagazione del segnale da un capo all'altro, è necessario che trascorra un tempo pari a $2T$ perché una stazione possa essere sicura di rilevare una collisione. Infatti, se una stazione A posta ad un'estremità della rete inizia a trasmettere al tempo t_0 , il suo segnale arriva a B (che si trova all'altra estremità della rete) solo al tempo $t_0 + T$. Se B inizia a trasmettere un attimo prima dell'istante in cui gli arriva il segnale, la collisione conseguente viene

rilevata da B quasi immediatamente, ma impiega un'ulteriore quantità T di tempo per giungere ad A, che quindi può rilevarla solo un attimo prima dell'istante $t_0 + 2T$.

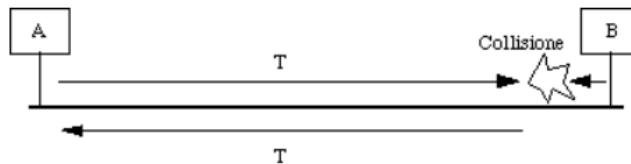
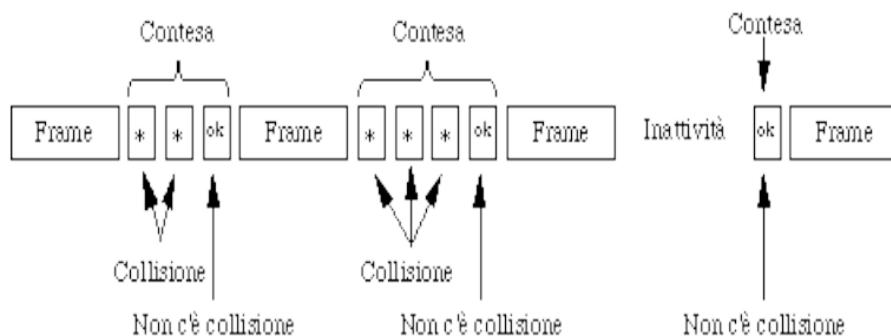


Figura 4-4: Rilevazione di una collisione

Si utilizza un **modello concettuale** per descrivere il comportamento dei protocolli CSMA/CD:



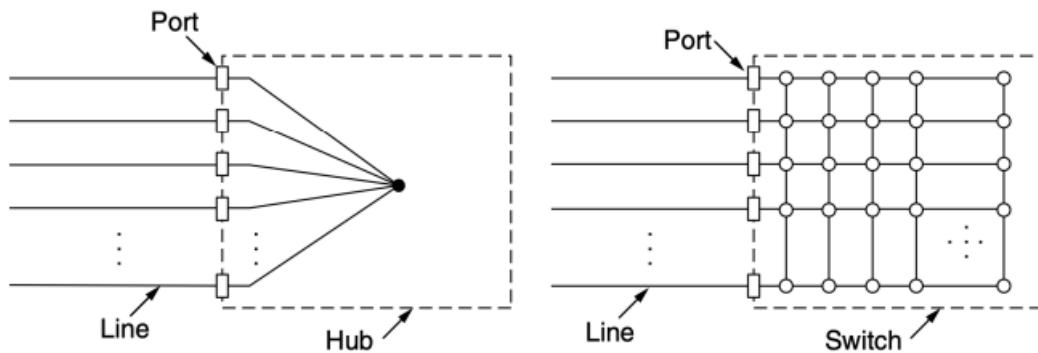
In pratica vi è un'alternanza di periodi di **contesa**, **trasmissione** e **inattività**, con il primo che è modellato come uno slotted con slot di durata $2T$.

Fast ethernet

Ethernet nasce all'inizio degli anni 80, previsto con topologia a bus su cavo coassiale, con velocità 10 Mbps e utilizzo del protocollo di livello MAC CSMA/CD. Successivamente si è evoluto in una topologia a stella, con cablaggio strutturato, doppini in rame e fibra ottica a 100 Mbps (802.3u), 1 Gbs (802.3z) e 10 Gbps (802.3ae). Il **fast ethernet** è lo standard 802.3u, ossia quello che prevede l'aumento di velocità da 10 Mbps a 100 Mbps. In base al supporto fisico utilizzato, i problemi del minimo tempo di trasmissione e della massima lunghezza di rete vengono risolti in modi diversi:

- nel caso in cui si scelga come supporto fisico il **doppino di cat.3 (100BaseT4)**, né vengono utilizzati 4 tra l'hub ed ogni stazione. Uno per il traffico dall'hub alla stazione, uno per il traffico dalla stazione all'hub e 2 usati di volta in volta nella direzione della trasmissione in corso. Lo standard ethernet 100BaseT4 **non** utilizza la codifica Manchester ma utilizza una **codifica 8B6T** (8 bit codificati con 6 trit, con questi ultimi che sono simboli ternari i cui possibili valori sono [+,-,0]). La **velocità di segnalazione** è 25 Mhz, contro i 20 dello standard 802.3, e si inviano 3 trit per volta sui 3 doppini. Poiché 3 trit convogliano 4 bit, si ottiene $4 \text{ bit} * 25 \text{ Mhz} = 100 \text{ Mbps}$;

- nel caso in cui si scelga come supporto fisico il **doppino di cat.5** (**100BaseT**), si avrà come **codifica la 4B5B** (4 bit codificati con 5 bit) e come **velocità di segnalazione** 125 Mhz. Se viene utilizzato un **hub tradizionale**, la **lunghezza massima di un ramo** sarà 100 metri, e di conseguenza il **diametro della rete** sarà 200 metri, mentre se si utilizza uno **switched hub**, ogni ramo è un **dominio di collisione separato** e quindi non esiste problema delle collisioni poiché su ogni ramo vi sarà un'unica stazione. Rimane, invece, il limite dei 200 metri.



Per quanto riguarda la **codifica 4B5B**, i 4 bit della sorgente sono mappati su **pattern fisso** di 5 bit sul mezzo fisico. Questi pattern fissi di 5 bit prendono il nome di **codeword** e non ne esistono, in questa codifica, con una sequenza di valori 0 maggiori di tre.

Sorgente	Mezzo fisico
0000	11110
0001	01001
0010	10100
0011	10101
0100	01010
0101	01011
0110	01110
0111	01111
1000	10010
1001	10011
1010	10110
1011	10111
1100	11010
1101	11011
1110	11100
1111	11101

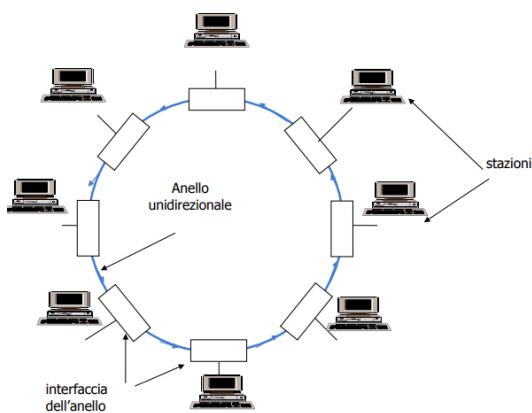
Sorgente	Mezzo fisico
0000	11110
0001	01001
0010	10100
0011	10101
0100	01010
0101	01011
0110	01110
0111	01111
1000	10010
1001	10011
1010	10110
1011	10111
1100	11010
1101	11011
1110	11100
1111	11101

;

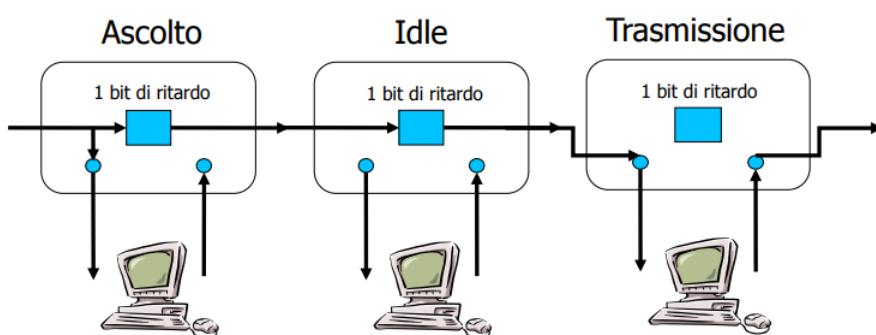
- nel caso in cui si scelga come supporto fisico la **fibra ottica (100BaseFX)**, si avrà una **velocità di segnalazione** di 125 Mhz e si utilizzeranno la **codifica 4B5B** e, in modo obbligatorio, uno **switched hub**. La **lunghezza dei rami** potrà arrivare fino a 2 km in quanto con uno switched hub non c'è il problema delle collisioni ed inoltre la fibra regge velocità anche dell'ordine dei Gbps a distanze anche superiori.

Token ring

Il token ring è un tipo di topologia di rete, ad anello, per collegare dispositivi in una rete LAN e prende il nome dall'utilizzo di un [token](#), appunto, il quale circola tra i dispositivi connessi per controllare l'accesso alla rete.



Il token ring utilizza un **insieme di collegamenti da punto a punto**. Ogni bit che raggiunge l'interfaccia viene copiato in un buffer di 1 bit, con quest'ultimo che viene ritrasmesso sull'anello dopo un eventuale controllo (ascolto) o modifica (trasmissione), e si ha un **ritardo di 1 bit per ogni interfaccia**.



Poiché l'anello deve contenere tutto il token, oltre al problema della lunghezza, **sono necessarie interfacce di rete che restino attive anche a terminale spento**. La soluzione consiste nel realizzare un **anello virtuale (wire center)**, accentratamente in un singolo apparato di controllo e comunicazione; **una specie di hub ma più complesso**. Nonostante il cablaggio wire center, l'assenza della corrente su un **lodo** causa la chiusura del corrispondente **relais** e, di conseguenza, l'esclusione di tale lodo.

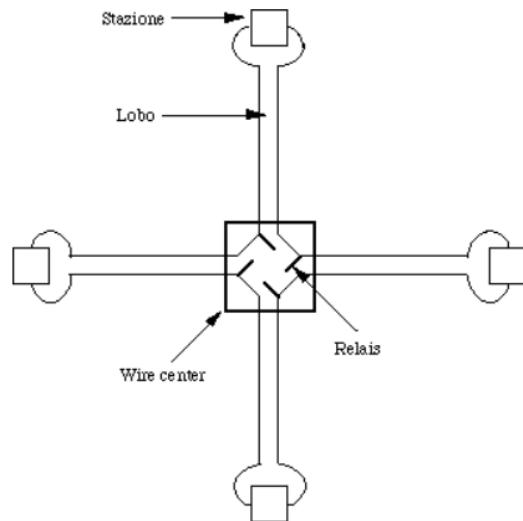


Figura 4-14: Cablaggio con wire center

I lobi hanno una lunghezza massima variabile a seconda del cablaggio utilizzato: se si utilizzano UTP di cat.4 la lunghezza massima sarà 150m, se si utilizzano UTP di cat.5 la lunghezza massima sarà 195m, mentre se si utilizzano STP la lunghezza massima sarà 340m.

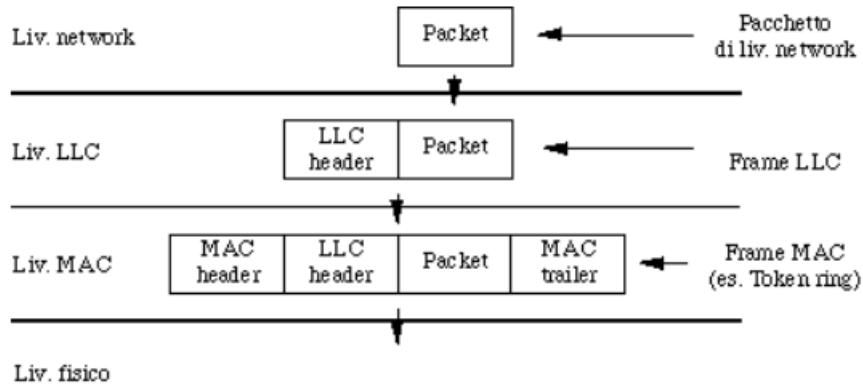
Logical Link Control (LLC – IEEE 802.2)

Il Logical Link Control definisce la parte superiore del livello data link, fornendo al livello network un'interfaccia unica, nascondendo le differenze tra i vari sottolivelli MAC. A differenza del sottolivello MAC che fornisce solo servizi datagram, il LLC, a seconda della richiesta, **fornisce servizi datagram, servizi datagram confermati o servizi affidabili orientati alla connessione**.

Frame LLC:

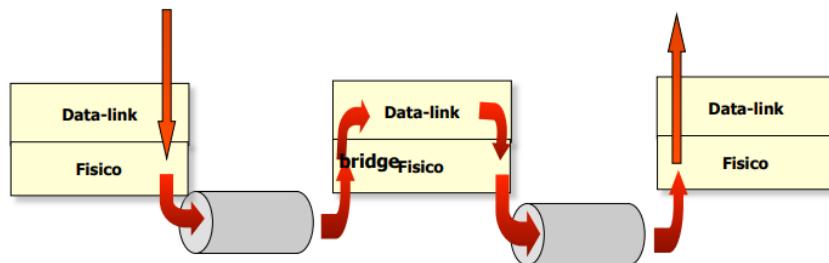
Destinazione	Mittente	Controllo	Informazione
1 Ottetto	1 Ottetto	1 o 2 Ottetti	m Ottetti

Gli indirizzi LLC, lunghi un byte (SSAP e DSAP, Source e Destination Service Access Point), indicano il protocollo di livello superiore che deve ricevere quel pacchetto; in questo modo il LLC offre un **supporto multiprotocollo**. In trasmissione il frame LLC viene imbustato nel frame del sottolivello MAC impiegato, con il processo inverso che avviene in fase di ricezione.

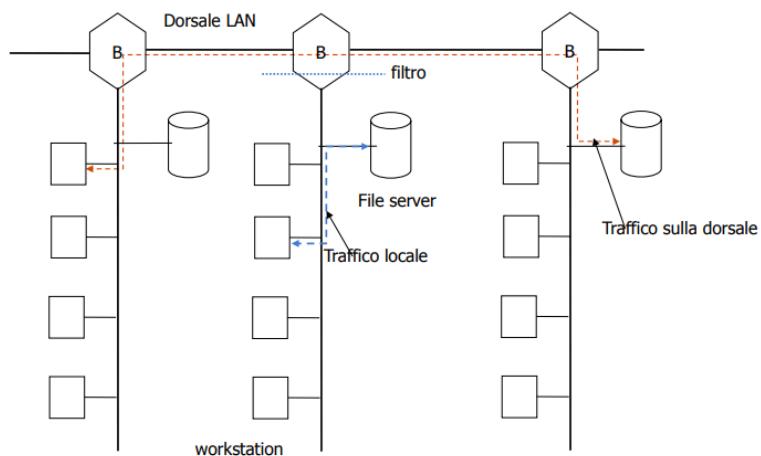


Bridge

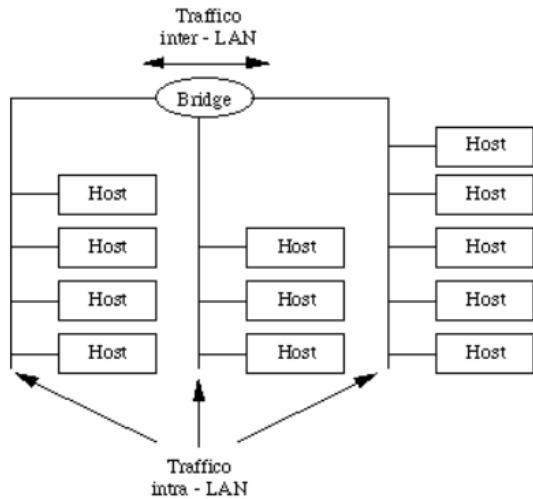
I bridge sono **dispositivi che operano a livello data link e permettono di connettere più segmenti di una stessa LAN**, mantenendo la suddivisione a livello data link. I bridge **possono collegare segmenti operanti con protocolli diversi, frazionano la rete in segmenti creando domini di collisione separati** (il traffico locale rimane confinato nella sottorete), **confinano i malfunzionamenti dovuti a stazioni difettose e aumentano la sicurezza dei dati**.



Come detto, i traffici locali rimangono confinati nelle sottoreti e consentono il passaggio solo alle comunicazioni che hanno mittente e destinatario su segmenti distinti.



Un bridge ha tante interfacce di rete quanti sono i segmenti ad esso collegati.

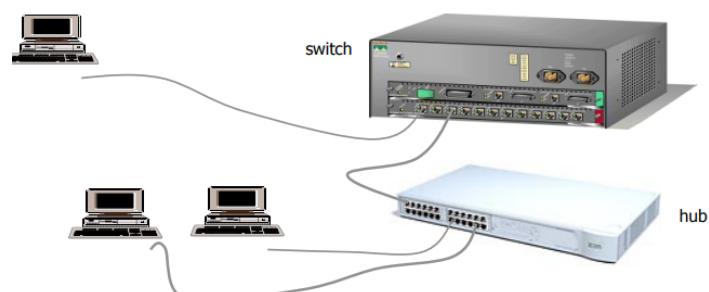


Quando una delle interfacce riceve un frame, lo passa al relativo software di livello MAC, il quale lo deimbusta e lo passa al software di livello LLC del bridge.

Sulla base dell'indirizzo MAC di destinazione, il bridge decide a quale segmento inviarlo: se destinatario e mittente si trovano sullo stesso segmento, il frame viene scartato, altrimenti il frame LLC viene passato al livello MAC dell'interfaccia collegata al segmento di destinazione, il quale lo imbusta (frame LLC) in un frame MAC e provvede ad inviarlo su tale LAN. Il bridge è **diverso da un ripetitore** in quanto quest'ultimo copia tutto ciò che riceve da una linea su tutte le altre, mentre il bridge, acquisito un frame, lo analizza, lo **ricostruisce** e lo inoltra, quindi **può anche essere configurato in modo da filtrare** (non far passare) traffico di uno specifico protocollo. Il filtraggio avviene in funzione dei due campi SSAP e DSAP del LLC. I bridge progettati per l'interconnessione di **segmenti di tipo diverso** (ad es. token ring ed ethernet) **devono risolvere diversi problemi**, tra cui i diversi formati dei frame, differenti data rate e diverse lunghezze massime, ma anche funzioni supportate o meno dalle differenti LAN (ad es. il concetto di priorità ed i bit A e C presenti in 802.5 ma non in 802.3).

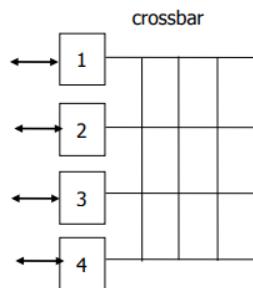
Switch

Gli switch sono **dispositivi che operano con le stesse modalità del bridge**, hanno un numero di porte maggiore di 2 e ogni porta può essere collegata ad un segmento della rete o ad una stazione singola.



Gli switch utilizzano **alcune tecnologie**:

- **shared memory**, che consiste nella **memorizzazione dei pacchetti in una memoria comune a tutte le porte**, per poi inviarli alla porta destinazione;
- **matrix**, ossia una **matrice di commutazione** che permette, **in base all'indirizzo e al contenuto della tabella, di attivare la connessione necessaria**;



- **bus-architecture**, cioè ha un **bus interno condiviso e ad alta velocità**, che per la comunicazione interna utilizza **TDMA**, ossia **Time Division Multiple Access**, (tecnica di gestione dell'accesso concorrente ad una risorsa condivisa).

Algoritmi di routing

Esistono due tipi di algoritmi di routing, quelli **statici**, o **non adattivi**, e quelli **dinamici**, o **adattivi**. Gli algoritmi di routing statici vengono preferiti quando si tratta di reti di dimensioni ridotte, mentre si preferiscono gli algoritmi di routing dinamico quando sono possibili frequenti cambi di topologia, magari dovuti a guasti del sistema o dei percorsi, oppure quando vi sono variazioni di prestazioni in dipendenza del carico di lavoro di router o linee.

Algoritmi statici di routing

Gli algoritmi statici di routing **sono eseguiti solamente all'avvio della rete e le decisioni di routing vengono applicate senza essere più modificate**. Un esempio di algoritmo statico di routing è lo **shortest path routing**, il quale calcola il minimo cammino tra una coppia di router. **All'avvio della rete**, o quando ci sono variazioni nella topologia, l'host impiegato per la gestione della rete **costruisce**, o ricostruisce, **un grafo della subnet**, individuando i router e le linee punto a punto tra essi. **Dopodiché applica un algoritmo**, come ad es. Dijkstra, **che calcola il cammino minimo tra ogni coppia di nodi e invia tali informazioni a tutti i router**. Il cammino minimo cambia in funzione delle grandezze che si desidera minimizzare, come il numero di **hop** (salti da un router ad un altro attraverso le reti intermedie), la lunghezza dei collegamenti, ecc. Come abbiamo detto, può essere

utilizzato l'**algoritmo di Dijkstra per il calcolo del cammino minimo tra ogni coppia di nodi**. Questo algoritmo lavora su **grafi orientati** che hanno **pesi non negativi** sui collegamenti. Ogni nodo del grafo rappresenta un router ed ogni arco una linea di comunicazione (canale). **Tra tutti i possibili cammini** tra il router mittente ed il router destinatario, **l'algoritmo sceglie quello con peso minimo**.

Come sappiamo, la **tecnica del flooding** consiste nell'inviare ogni pacchetto su tutte le linee eccetto quella da cui è arrivato. In linea di principio il flooding **può essere usato come algoritmo statico di routing** ma presenta l'inconveniente di **generare un numero enorme di pacchetti**, addirittura teoricamente infinito.

Si possono comunque utilizzare alcune **tecniche per limitare il traffico generato**:

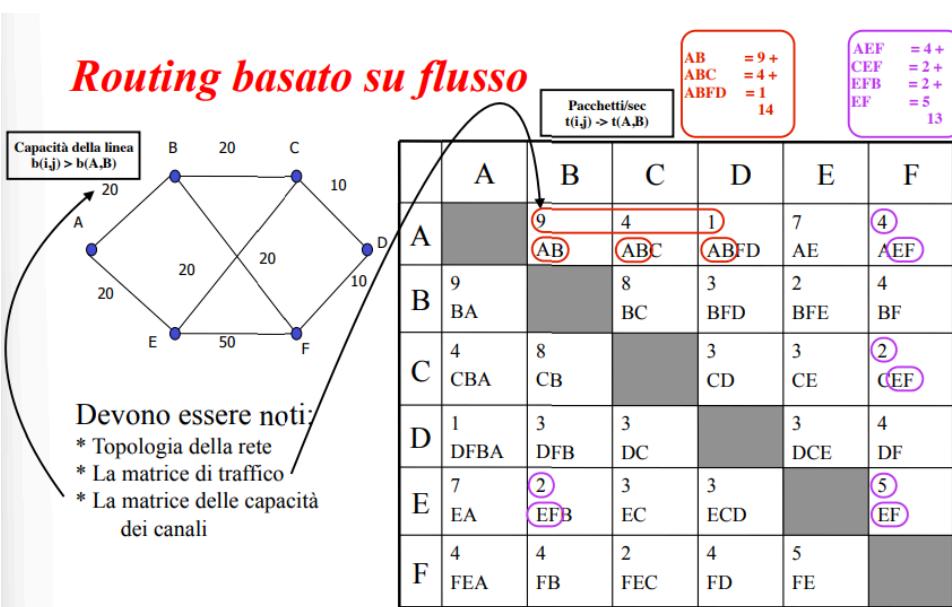
- si inserisce in ogni pacchetto un **contatore** che viene decrementato ad ogni hop e, quando il contatore arriva a 0, il pacchetto viene scartato.
Un appropriato valore iniziale per il contatore può essere il diametro della subnet;
- si inserisce in ogni pacchetto la **coppia (source router ID, sequence number)**, dopodichè un router prende nota ed accetta il pacchetto solo la prima volta, scartando arrivi successivi;
- si **duplicano i pacchetti solo sulle linee che vanno all'incirca nella giusta direzione** e, per questo, si devono mantenere apposite tabelle.
Questa tecnica prende il nome di **selective flooding**.

Come ultimo algoritmo di routing statico abbiamo il **flow-based routing**, il quale **sceglie il cammino che minimizza il ritardo medio**. Questo algoritmo calcola in anticipo il traffico atteso su ogni linea, in modo da **stimare il ritardo medio atteso per ciascuna linea**. Per applicare l'algoritmo è necessario conoscere la **topologia della rete**, la **matrice del traffico** $T(i,j)$, i cui elementi $t(i,j)$ indicano il **traffico stimato** tra il router i ed il router j , e la **matrice delle capacità** $B(i,j)$, i cui elementi $b(i,j)$ indicano la **capacità della linea punto a punto** che collega il router i al router j . Il flow-based routing calcola il **ritardo medio dell'intera rete** effettuando la **somma pesata dei ritardi delle singole linee**, con il **peso di ogni linea** che è dato dal traffico su quella linea diviso il traffico totale sulla rete.

Fissato un percorso tra mittente e destinatario **si seguono determinati passi**:

- si calcola il traffico che incide su ogni linea, dato dalla somma di tutti i $t(i,j)$ instradati su quella linea;
- si calcola il ritardo di ogni linea;
- si calcola il ritardo medio dell'intero percorso;
- si ripete il procedimento per tutti i possibili percorsi, scegliendo alla fine quello che ha il minimo ritardo medio.

Routing basato su flusso



La Formula di Little ci dice che :

$$\text{Ritardo medio} = \frac{1}{\text{Pacchetti/ s} - \text{Traffic}}$$

Routing basato su flusso

i	Linea	Traffico (p/s)	Capacità(kbps)	Pacchetti/s	Ritardo(ms)	Peso
1	AB	14	20	25	91	0.17
2	BC	12	20	25	77	0.14
3	CD	6	10	12.5	154	0.07
4	AE	11	20	25	71	0.13
5	EF	13	50	62.5	20	0.15
6	FD	8	10	12.5	222	0.09
7	BF	10	20	25	67	0.12
8	EC	8	20	25	59	0.09

Dimensione media del pacchetto 800 bit e Traffico totale 82 pacchetti/s

Con riferimento alla tabella precedente, se consideriamo una dimensione media del pacchetto di 800 bit ed un traffico totale di 82 pacchetti/s distribuiti come in tabella, per la linea AB, avendo ipotizzato una capacità di 20 Kbps si ha:

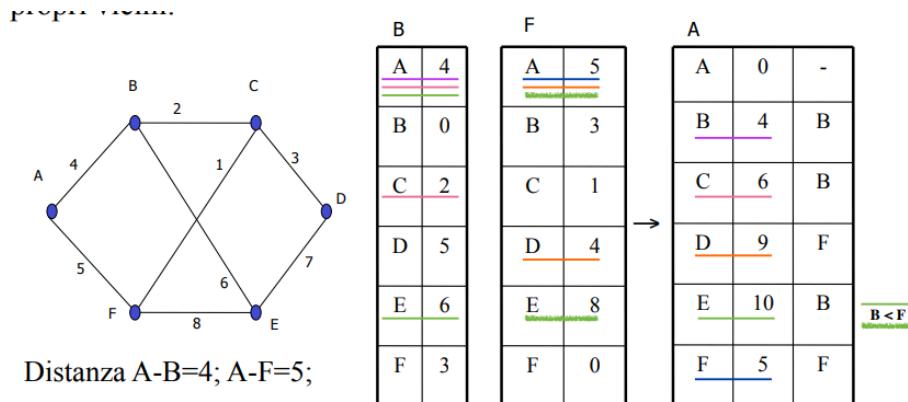
$$\text{Pacchetti/s}=20 \text{ Kbps} / 800 \text{ bit} = 20000 / 800 = 25$$

$$\text{Ritardo(ms)} = 1 / (25 - 14) \text{ p/s} = 0.091 \text{ s} = 91 \text{ ms}$$

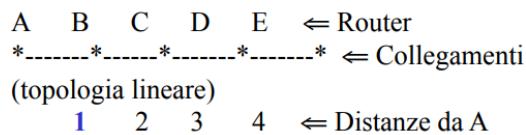
$$\text{Peso} = 14 / 82 = 0.17$$

Algoritmi dinamici di routing

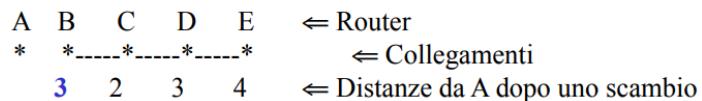
Nelle moderni reti vengono utilizzati gli algoritmi dinamici di routing, in quanto si adattano automaticamente ai cambiamenti della rete. Uno dei principali algoritmi dinamici di routing è il **distance vector routing** (routing dinamico **basato sulla distanza**), nel quale ogni router mantiene nella propria **tabella (vector)** di routing la distanza (numero di hop, ritardo, ecc.) che lo separa da ogni altro router e la linea in uscita da usare per arrivarci. La **stima della distanza** viene fatta misurando il tempo di risposta a **speciali pacchetti ECHO** che il router invia agli altri router ad esso fisicamente connessi. Nel distance vector routing **ogni router invia periodicamente la propria tabella a tutti i router vicini**, che a loro volta inviano la propria. In base a tali tabelle, per ogni destinazione, vengono ricalcolati e registrati nella propria tabella (vector) i migliori percorsi. Per valutare il singolo percorso, ogni router rileva dal proprio vector le distanze con i router confinanti e, dalle tabelle ricevute, quelle (le distanze) tra i vicini ed i router remoti. Come abbiamo detto, **nelle tabelle mantenute da ogni router viene memorizzata la più piccola distanza conosciuta per ogni destinazione e quale linea utilizzare per raggiungerla.**



L'algoritmo distance vector routing è **lento a reagire quando un collegamento cade**:

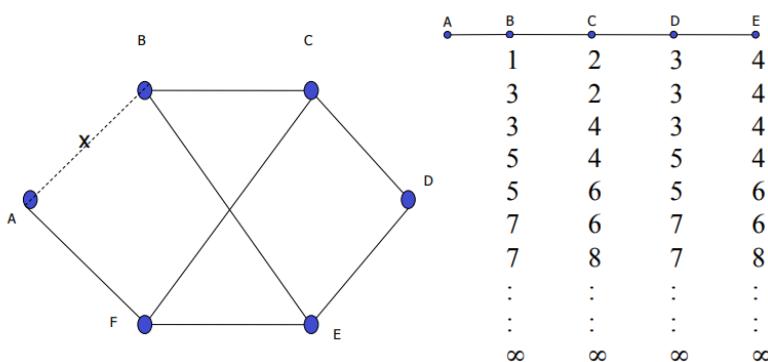


Se cade la linea fra A e B, dopo uno scambio:



Non ricevendo risposta da A, B crede di poterci arrivare via C, la quale ha distanza 2 da A. B cambia quindi la propria distanza da A a 3, considerando che si trova a distanza 1 da C più 2 che è la distanza da C ad A. A lungo andare tutti i router vedono lentamente aumentare sempre di più la distanza per arrivare ad A e questo

è il problema cosiddetto **count-to-infinity**, il quale non tiene conto della capacità delle linee e converge con tempi molto lunghi.



Se la distanza rappresenta il numero di hop si può porre come limite il diametro della rete, ma se essa (la distanza) rappresentasse il tempo, il limite dovrebbe essere molto alto poiché, altrimenti, i cammini con un ritardo alto (magari a causa della congestione) verrebbero considerati interrotti. Nonostante ciò, il distance vector routing **era l'algoritmo di routing di ARPANET ed era usato anche in Internet con il nome di Routing Internet Protocol (RIP)**.

Un altro algoritmo dinamico di routing è il **link state routing** (routing dinamico basato sullo stato dei collegamenti), dove **ogni router controlla lo stato dei collegamenti, misurando il ritardo di ogni linea**, con i suoi vicini immediati, e **distribuisce tali informazioni a tutti gli altri**. Ciascun router, sulla base delle informazioni ricevute, **costruisce localmente la topologia dell'intera rete e calcola il cammino tra sé e tutti gli altri router**. Quando il router si avvia, invia un **pacchetto HELLO** su tutte le linee in uscita e riceve, in risposta, gli indirizzi dei propri vicini. Successivamente invia vari **pacchetti ECHO**, misura il tempo di arrivo della risposta e, mediando su pacchetti di varia lunghezza, **calcola il ritardo della linea**. A questo punto **costruisce un pacchetto con identità del mittente, numero di sequenza del pacchetto, età del pacchetto e lista dei vicini con i relativi ritardi**. La costruzione e l'invio di tali pacchetti si verifica ad intervalli regolari o quando accade un evento significativo, come ad esempio quando un collegamento cade. **Errori nella distribuzione dei pacchetti** possono indurre i router in errore sull'effettiva topologia, con conseguenti malfunzionamenti. Proprio per quanto riguarda la distribuzione, di base viene utilizzata la tecnica del [flooding](#), inserendo nei pacchetti le coppie (source router ID, sequence number) per eliminare, in fase di ricezione, i duplicati. **Per evitare che ci siano pacchetti vaganti**, per qualche errore, viene utilizzata l'**età del pacchetto**. Questa (età del pacchetto) viene decrementata nel tempo e, quando arriva a 0, il pacchetto viene scartato. Il link state routing **venne attivato nel 1979 su ARPANET ed i passi che segue**, nel suo funzionamento, sono:

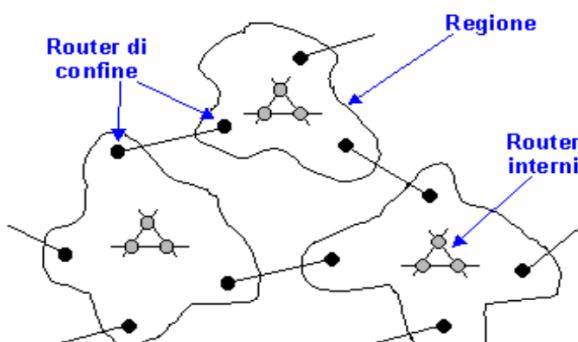
- scoprire i propri vicini, ed i loro indirizzi di rete, spedendo uno speciale pacchetto HELLO su ogni linea;

- misurare il ritardo o il costo per ognuno dei suoi vicini mediante speciali pacchetti ECHO;
- costruire un pacchetto contenente le informazioni appena scoperte, più l'identità del router ed un numero di sequenza a 32 bit;
- spedire questo pacchetto a tutti i router mediante la tecnica del flooding;
- Calcolare il cammino minimo per ogni altro router utilizzando l'algoritmo di Dijkstra.

Il link state routing è molto utilizzato attualmente, infatti l'**Open Shortest Path First (OSPF)**, che **si sta avviando ad essere l'algoritmo più utilizzato in Internet**, è basato su tale principio.

Routing gerarchico

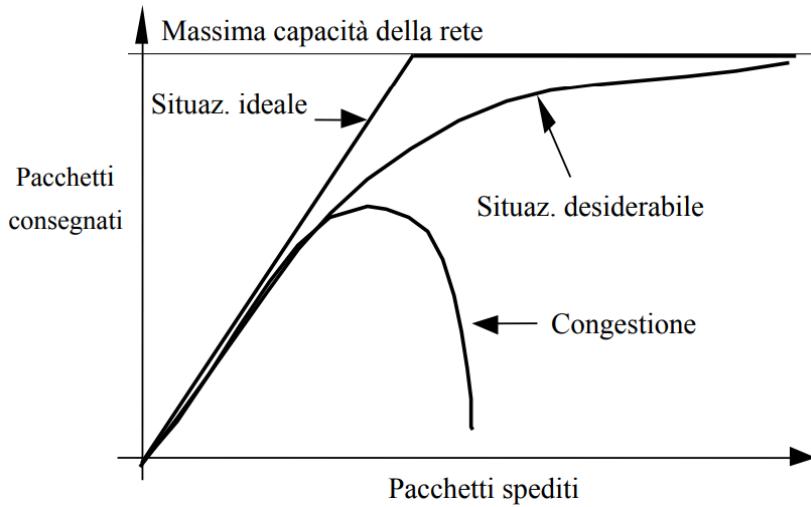
Quando la rete cresce fino a contenere decine di migliaia di nodi, diventa **troppo gravoso mantenere in ogni router la topologia completa**. Di conseguenza il **routing viene impostato in modo gerarchico, suddividendo la rete in zone**, anche dette **regioni**.



All'interno di una regione vale quanto visto finora, cioè i router, detti interni, utilizzano gli algoritmi visti. Quando, invece, un **router interno deve spedire qualcosa ad un router di un'altra regione**, deve passare attraverso un particolare router, detto **di confine**. E' poi compito del router di confine stabilire a quale altro router di confine, ovviamente di diversa regione, inviare i dati. Nel caso in cui questi due livelli non bastassero, l'organizzazione viene ripetuta su più livelli. Per questo tipo di routing le tabelle vengono gestite in modi diversi in base al tipo di router di cui si tratta. Nelle **tabelle dei router interni**, viene utilizzata **una riga per ogni altro router interno**, con la relativa linea da utilizzare per arrivarcì, e **una riga per ogni regione**, con l'indicazione del relativo router di confine e della linea da utilizzare per arrivarcì. Per quanto riguarda invece le **tabelle dei router di confine**, viene usata **una riga per ogni regione**, con l'indicazione del router di confine da contattare e della linea da utilizzare per arrivarcì.

Congestione

Quando vi è **degrado delle prestazioni dovuto alla presenza di troppi pacchetti** in una parte della subnet si sta in realtà parlando del concetto di **congestione**.



La congestione può essere determinata dalla presenza di **troppi pochi buffer nel router**, dal **processore del router che risulta essere troppo lento**, oppure da una **linea di trasmissione troppo lenta**. La congestione in un router **tende a propagarsi ai suoi vicini** poiché, quando tale router è costretto a scartare i pacchetti che riceve, non li conferma più, e quindi i router che li hanno spediti devono mantenerli nei propri buffer, aggravando così anche la propria situazione. Il **controllo della congestione è un problema di tutta la rete** e, per gestirlo, vi sono **due possibili approcci**: l'**approccio open loop (senza controllazione)** e l'**approccio closed loop (con controllazione)**. Nel primo si fissano parametri per la rete, come ad es. la velocità di trasmissione e l'ampiezza dei buffer, in modo che la congestione non si verifichi, ma poi non vengono effettuate azioni correttive. Nel secondo, invece, viene controllata la situazione della rete, intraprendendo azioni opportune quando necessario.

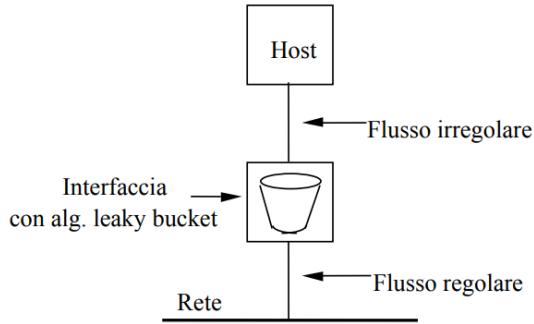
Traffic shaping (open loop)

Un tipo di approccio **open loop** è il **traffic shaping**, nel quale l'idea è di **forzare la trasmissione dei pacchetti ad un ritmo piuttosto regolare**, in modo da **limitare la possibilità di congestioni**. Esistono **tre tecniche** per implementare il traffic shaping:

- **leaky bucket**;
- **token bucket**;
- **flow specification**.

Leaky bucket (secchio che perde)

Dal nome di questo algoritmo possiamo trarre un'**analogia reale** che ci permette di spiegare meglio il funzionamento del leaky bucket. Consideriamo un secchio, riempito da un rubinetto, di cui si può regolare l'apertura. Il secchio riversa l'acqua che contiene attraverso un forellino sul fondo, a ritmo costante. Se viene immessa troppa acqua, essa fuoriesce dal bordo superiore del secchio e si perde.



Allo stesso tempo, tramite questo algoritmo, il **router riversa sulla rete pacchetti con un data rate fissato**, diciamo b bps, accodando nei propri buffer quelli ricevuti ma non ancora trasmessi. **Se l'host genera più pacchetti di quelli che possono essere contenuti nei buffer, essi si perdono**. In questo modo, l'host può anche produrre un traffico bursty senza creare problemi sulla rete, in quanto finché il data rate medio non supera i b bps, tutto funziona regolarmente; in caso di superamento, invece, si cominciano a perdere pacchetti.

Token bucket (secchio di gettoni)

L'algoritmo token bucket **consente un grado di irregolarità controllato anche nel flusso che esce sulla rete**. Il bucket in questo caso contiene dei **token**, i quali **si creano con una cadenza prefissata fino ad un valore massimo prefissato** che equivale al riempimento del secchio. In base a tale algoritmo, un **host che non trasmette, accumula un credito trasmissivo** con un certo data rate, fino ad un massimo consentito, mentre **quando invece ha dati da trasmettere**, sfruttando se necessario tutto il credito disponibile, trasmette alla massima velocità consentita dalla linea.

Flow specification

Il **traffic shaping è molto efficace se sorgente, subnet e destinazione si accordano riguardo alle caratteristiche del traffico che si vuole inviare** (data rate, grado di burstiness, ecc.) e **alla qualità del servizio** (ritardo massimo, frazione di pacchetti che si può perdere, ecc.). Questo accordo, preso prima di trasmettere, può essere fatto sia in subnet connection-oriented che in subnet connectionless. Mentre **nelle subnet connesse** l'accordo si riferisce ai **circuiti virtuali** (connessione temporanea, alla base delle connessioni connection-oriented,

tra due dispositivi di una rete, prima che i dati vengano effettivamente trasmessi), **in quelle non connesse si riferisce alla sequenza di pacchetti che sarà trasmessa**. Nelle reti basate su **circuiti virtuali**, per evitare la congestione, **si nega l'attivazione di nuovi circuiti virtuali** ove non vi fossero sufficienti risorse per gestirli; questa tecnica prende il nome di **admission control**.

Choke packet (closed loop)

Nell'approccio **choke packet**, di tipo **closed loop**, è previsto che un **router controlli il grado di utilizzo delle sue linee d'uscita**. In particolare quando il **grado di utilizzo di una linea in uscita si avvicina ad una soglia prefissata**, il router con un **choke packet** invita l'host d'origine a **diminuire il flusso**. Dopo aver ricevuto un choke packet, l'host, dopo aver diminuito il flusso, continua a trasmettere ignorando, per un tempo prefissato, i successivi choke packet, in quanto ne arrivano molti in sequenza. Trascorso tale tempo, l'host si rimette in attesa di choke packet e, se ne arrivano, riduce ulteriormente il flusso di trasmissione. Il **problema di questa tecnica** è che intercorre un certo tempo tra quando l'host riceve i choke packet e quando inizia a diminuire il ritmo della trasmissione. Per migliorare questa situazione viene utilizzata una variante dei choke packet, chiamata **hop-by-hop choke packet**, nel quale i router che li ricevono (i choke packet) rallentano immediatamente il ritmo di trasmissione, mantenendo parte dei pacchetti nei propri buffer. Questa tecnica richiede, però, più spazio di buffer nei router sul percorso dall'host d'origine a quel router.

Internetworking

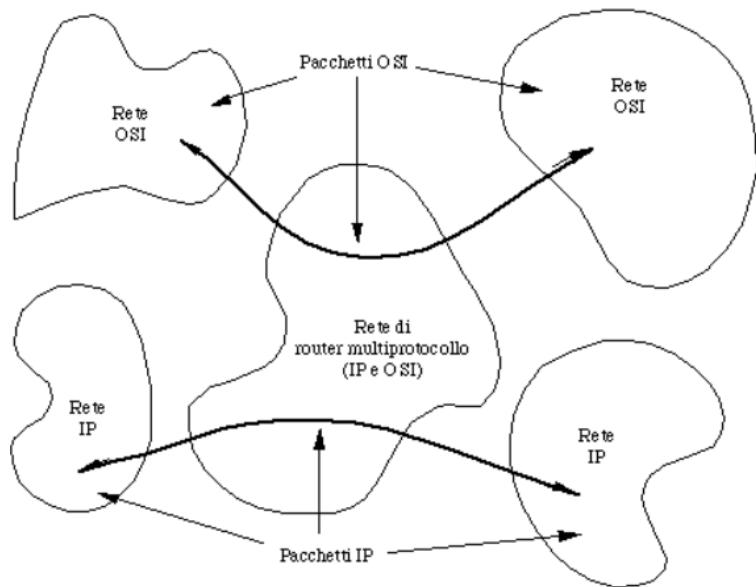
Per connettere reti eterogenee si devono **risolvere problemi di diversità in vari campi**:

- servizi offerti (connected-oriented o connectionless);
- formati dei pacchetti e degli indirizzi;
- meccanismi di controllo dell'errore e della congestione;
- dimensioni massime dei pacchetti.

Per superare questi problemi vengono **utilizzate diverse tecniche**, come ad esempio i **router multiprotocollo** o il **tunneling**.

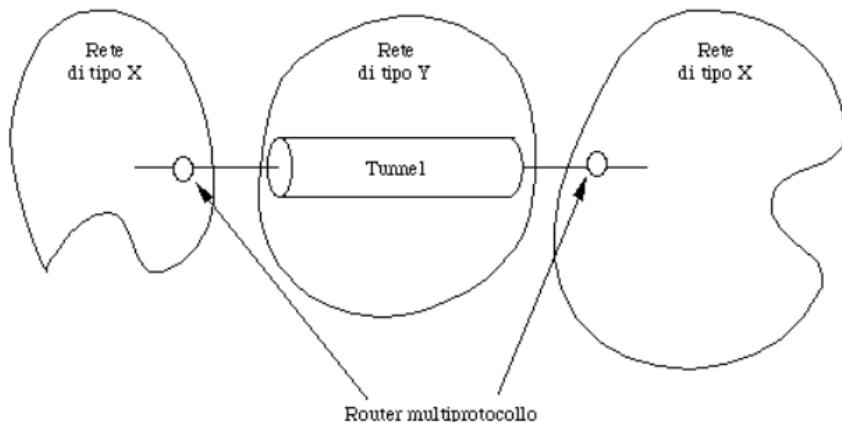
Reti di router multiprotocollo

Tramite questa tecnica, **ogni rete può comunicare con le altre reti a lei conformi attraverso una porzione di rete costituita di router multiprotocollo**, che si occupano di instradare i pacchetti secondo le regole di competenza dell'architettura di cui fanno parte (ad es. OSI, IP, ecc.). Nell'esempio sottostante vediamo che due reti OSI possono comunicare tra loro, come anche le due reti IP.

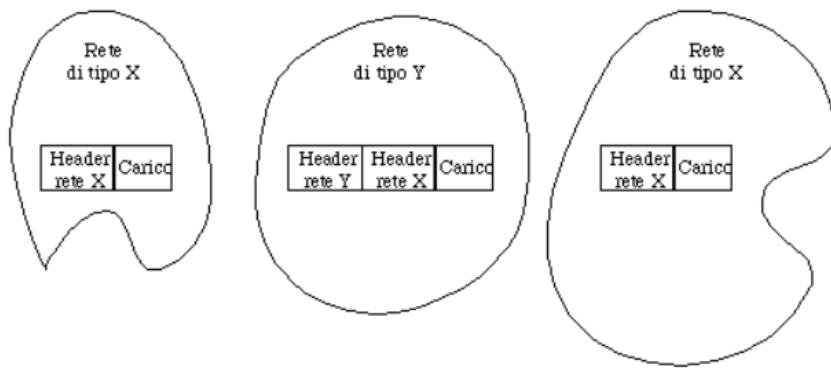


Tunneling

La tecnica del tunneling **permette di mettere in comunicazione due reti uguali per mezzo di una rete diversa.**

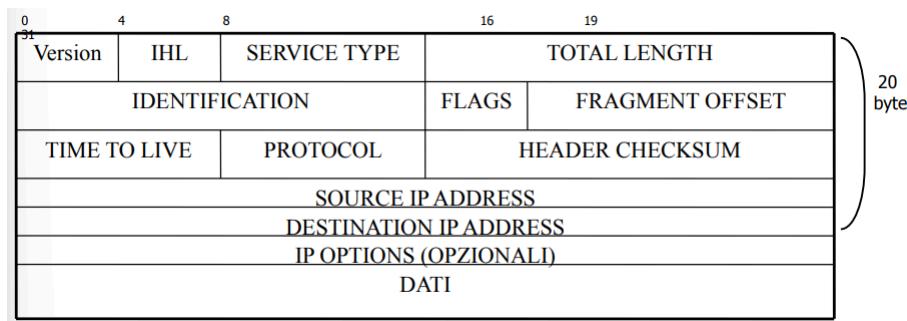


Nell'esempio soprastante la rete di tipo Y non è dotata di router multiprotocollo, che invece esiste in ciascuna delle due reti di tipo X. Questi router encapsulano i pacchetti delle reti di tipo X all'interno di pacchetti di tipo Y, consegnandoli alla rete di tipo Y.



Internet Protocol (IP)

L'IP è un protocollo datagram, quindi non connesso e non affidabile, che riceve i dati dal livello transport e li incapsula in pacchetti (solitamente di circa 1500 byte), instradando questi ultimi sulla subnet, eventualmente frammentandoli lungo il viaggio. Una volta arrivati a destinazione, riassembra, se necessario, i frammenti in pacchetti, estrae da questi i dati del livello transport e consegna a quest'ultimo (livello transport) i dati arrivati. L'IP è un protocollo senza connessione, come detto, standardizzato in **RFC 791** e rappresenta la base di Internet. Come abbiamo detto, l'IP utilizza i datagrammi, la cui **consegna non è garantita**, che vengono spediti/gestiti indipendentemente e contengono l'indirizzo del destinatario.



I datagrammi IP seguono una **politica di best-effort**, ossia il protocollo IP fa del suo meglio per consegnare i pacchetti, ma senza garantire la consegna o la qualità del servizio. I datagrammi IP possono essere ritardati, duplicati, distribuiti fuori ordine, persi e addirittura pacchetti diversi dello stesso messaggio possono seguire percorsi diversi. Tutte queste situazioni sono dirette conseguenze del fatto che IP è un protocollo senza connessione e non affidabile.

Classi di indirizzi IP

	bits	0 1 2 3 4	8	16	24	31	
Class A	0	prefix		suffix			1.0.0.0 127.255.255.255
Class B	10		prefix		suffix		128.0.0.0 191.255.255.255
Class C	110		prefix		suffix		192.0.0.0 223.255.255.255
Class D	1110			multicast address			224.0.0.0 239.255.255.255
Class E	1111			reserved for future use			240.0.0.0 247.255.255.255

classe	bit nel prefisso	numero massimo di reti	bit nel suffisso	numero massimo di host per rete
A	7	128	24	16777216
B	14	16384	16	65536
C	21	2097152	8	256

Indirizzi speciali

prefisso	suffisso	indirizzo	scopo
tutti 0	tutti 0	questo computer	utilizzato in fase di boot
rete	tutti 0	la rete	identificare una rete
rete	tutti 1	broadcast diretto	broadcast su una rete remota
tutti 1	tutti 1	broadcast locale	broadcast sulla propria rete
127	qualsiasi	loopback	Test

Quando si parla di **broadcast locale**, o limitato, i pacchetti sono confinati alla sottorete, quindi il router non li propaga all'esterno. Per quanto riguarda il **loopback**, invece, il pacchetto viene smistato localmente e solo a livello rete (non viene passato al livello fisico). Altri indirizzi speciali sono **0.0.0.0** che indica **questo host su questa rete** e **255.255.255.255** (ossia tutti 1) che indica **broadcast sulla rete locale**.

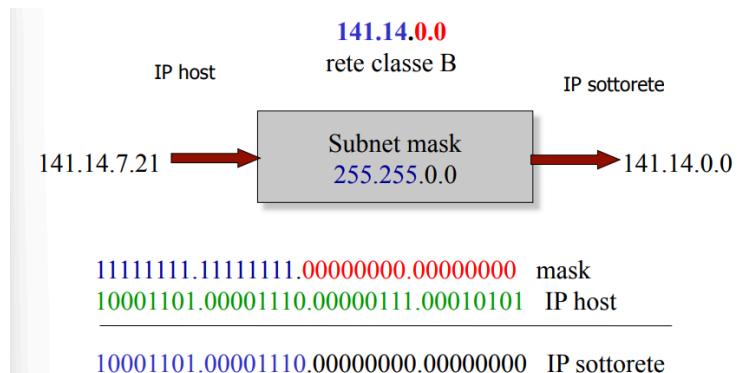
Indirizzi privati

Gli indirizzi privati sono **indirizzi riservati per reti**, appunto private, non connesse ad internet. Abbiamo **diversi indirizzi privati nelle varie classi**:

classe	rete	numero reti
A	10.0.0	1
B	Da 172.16 a 172.31	16
C	Da 192.168.0 a 192.168.255	256

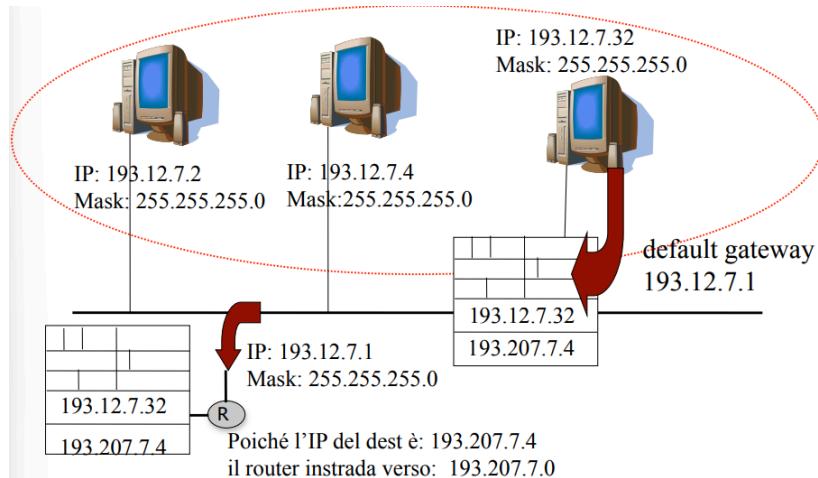
Subnet mask

Ogni rete può essere divisa in varie sottoreti, a loro volta contenenti degli host. Come sappiamo, l'indirizzo IP è composto dall'indirizzo di rete (net_id) e poi dall'host_id, relativo all'indirizzo dell'host all'interno della rete. Per individuare l'indirizzo di rete dall'indirizzo IP completo, viene utilizzata una maschera, chiamata **subnet mask**, formata da **bit 1 in corrispondenza del net_id** e da **bit 0 in corrispondenza dell'host_id**.



Indirizzo di rete: AND bit a bit fra l'IP e la maschera

La subnet mask **permette di verificare se l'indirizzo IP del destinatario appartiene o meno alla rete del mittente**. Se l'host destinatario non è nella stessa sottorete, il pacchetto verrà instradato verso il router (**gateway**), che si occupa dell'instradamento all'esterno della sottorete.



Protocolli ARP e RARP

Il protocollo **Address Resolution Protocol (ARP)** permette di **associare all'indirizzo IP del destinatario il suo indirizzo fisico**. Questa **associazione può essere**:

- **statica**, dove le associazioni vengono registrate in una tabella, periodicamente aggiornata, memorizzata su tutti i dispositivi della sottorete;
- **dinamica**, dove le associazioni si ottengono all'occorrenza attraverso esplicite richieste inoltrate sulla rete.

Per quanto riguarda il protocollo ARP, il mittente inoltra in broadcast una richiesta ARP per l'indirizzo IP di interesse e la macchina con quell'IP risponde con il suo indirizzo MAC (ossia fisico). Per evitare di effettuare una stessa associazione tra indirizzo IP e MAC, per ogni pacchetto le coppie IP:MAC ottenute vengono registrate in una **cache locale**. Esiste anche un protocollo opposto ad ARP, chiamato **Reverse Address Resolution Protocol (RARP)**, che **permette di associare ad un indirizzo MAC un indirizzo IP**.

Variable length subnet mask (VLSM)

E' possibile dividere una subnet in più subnet e questo porta ovviamente alla creazione di diverse maschere di rete per ogni subnet.

Es.

192.205.7.0/26 - 64 indirizzi (62 hosts + 2 rete/bc)

Dividiamo la sottorete in 2 ulteriori sottoreti, necessitiamo di **1 bit** ulteriore – rimangono **5 bit per gli hosts** (32 indirizzi)

11111111	11111111	11111111	11100000	
8	8	8	3	27

S1: 192.205.7.0/27 -> 0-31 [0: rete, 31: bc; 1-30 hosts]

S2: 192.205.7.32/27 -> 32-63 [32: rete, 63: bc; 33-62 hosts]

Proviamo a dividere la sottorete 193.205.7.64/26 in modo da avere 3 reti:

Subnet A (SA): 28 hosts

Subnet B (SB): 12 hosts

Subnet C (SC): 12 hosts

Subnet root: 193.205.7.64/26

Netmask: 11111111.11111111.11111111.11000000

Proviamo a dividere la sottorete 193.205.7.64/26 in modo da avere 3 reti:

Subnet A (SA): 28 hosts

Necessitiamo di almeno **5 bit hosts**

11111111	11111111	11111111	11100000	
8	8	8	3	27

8	8	8	3	27

*SA1: 193.205.7.64/27 -> 64-95 [64: rete, 95: bc; 65-94 hosts]

SA2: 193.205.7.96/27 -> 96-127 [96: rete, 127: bc; 97-126 hosts]

Subnet B (SB): 12 hosts

Due strade:

1. Usiamo una sottorete differente
2. Subnet della Subnet A

Necessitiamo di almeno 4 bit hosts (1 bit subnet)

11111111	11111111	11111111	11110000	
8	8	8	4	28

8	8	8	4	28

Subnet di SA2

SA1: 193.205.7.96/28 -> 96-111 [96: rete, 111: bc; 97-110 hosts]

SA2: 193.205.7.112/28 -> 112-127 [112: rete, 127: bc; 111-126 hosts]

Subnet B e Subnet C vengono assegnate a Subnet A (SA2)

Subnet B:

SA1: 193.205.7.96/28 -> 96-111 [96: rete, 111: bc; 97-110 hosts]

Subnet C:

SA2: 193.205.7.112/28 -> 112-127 [112: rete, 127: bc; 111-126 hosts]

Riassumendo...

Subnet	Address	Broadcast	Origin
S1	193.205.7.0/27	193.205.7.31/27	193.205.7.0/26
S2	193.205.7.32/27	193.205.7.63/27	193.205.7.0/26
SA1	193.205.7.64/27	193.205.7.95/27	193.205.7.64/26
SA2	193.205.7.96/27	193.205.7.127/27	193.205.7.64/26
SB	193.205.7.96/28	193.205.7.111/28	193.205.7.96/27
SC	193.205.7.112/28	193.205.7.127/28	193.205.7.96/27

Internet Control Message Protocol (ICMP)

L'ICMP è un **protocollo utilizzato per il controllo del funzionamento della subnet**, come anche i protocolli ARP e RARP ad esempio. L'operatività della subnet viene controllata continuamente dai router, scambiandosi informazioni mediante messaggi conformi al protocollo ICMP. I **pacchetti ICMP** sono impiegati per la **segnalazione di errore**, ed in tal caso sono inviati al mittente, oppure per **messaggi di richiesta**. Nel primo caso possono esserci errori come:

- **destination unreachable**, quindi il pacchetto non può essere consegnato poiché la destinazione non è raggiungibile;
- **time exceeded**, nel caso in cui il [TTL](#) raggiunga valore 0, oppure quando non si sono ricevute tutte le parti di un pacchetto frammentato entro un tempo limite;
- **parameter problem**, quando vi sono errori nei parametri dell'header.

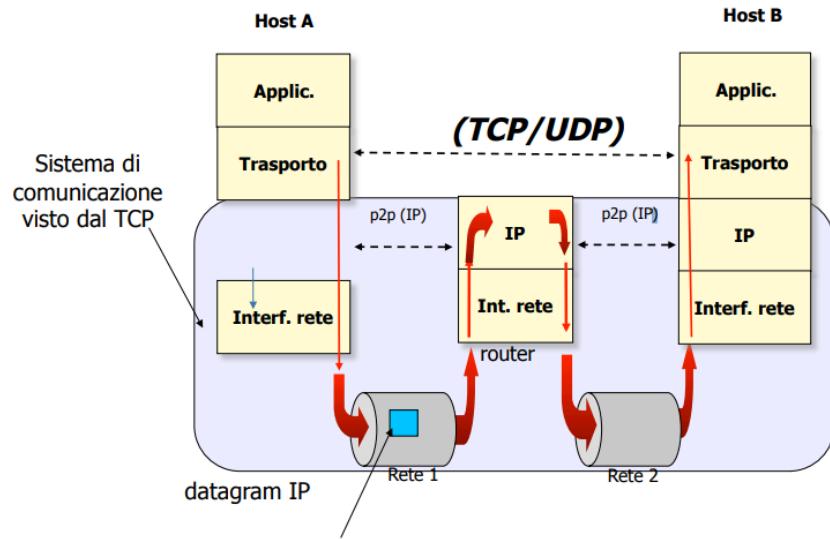
Quando, invece, si tratta di **messaggi di richiesta** abbiamo:

- **echo request/echo reply**, quindi richiesta/risposta di pacchetti echo, ossia si verifica se un host è raggiungibile e, quando si invia una richiesta, ci si aspetta una risposta;
- **timestamp request/timestamp reply**, uguale al precedente, ma con in più la registrazione dell'istante di partenza e di arrivo, in modo da misurare le prestazioni della rete.

TCP e UDP

TCP e UDP sono due protocolli di trasporto definiti su IP, ed in particolare **TCP, Transmission Control Protocol**, è un **protocollo di trasporto orientato alla connessione**, definito in RFC 793, RFC 1122 e RFC 1323 e progettato per fornire un flusso affidabile end-to-end su una connessione tra reti inaffidabile, mentre **UDP, User Data Protocol**, è un **protocollo senza connessione**, descritto in RFC 768 e che permette di inviare pacchetti IP senza stabilire una connessione.

TCP/IP



Funzionalità del TCP

TCP in fase di trasmissione:

- riceve un flusso di dati dall'applicazione;
- organizza questi dati in unità lunghe al massimo 64 Kb, eventualmente bufferizzandoli (i dati) prima di spedire il pacchetto;
- spedisce le unità di dati come pacchetti IP.

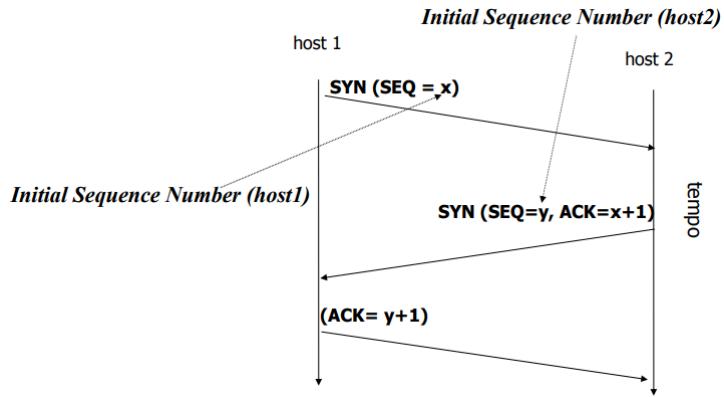
TCP in fase di ricezione:

- riceve i pacchetti IP;
- ricostruisce e consegna all'applicazione la sequenza corretta del flusso di byte originale.

Inoltre, il TCP, ha come funzionalità anche quella di ritrasmettere i pacchetti non ricevuti e riordinare quelli arrivati in ordine sbagliato.

Apertura di una connessione

Per l'apertura di una connessione TCP si utilizza un **protocollo 3-way handshake**.



I valori x e y della figura soprastante sono ricavati dagli host sulla base dei loro **clock di sistema**, il cui valore si incrementa di 1 unità ogni 4 microsecondi. Se il TCP riscontra l'assenza del processo che deve essere in attesa sulla porta destinazione, manda un **segmento di rifiuto** della connessione tramite il flag RST.

Un esempio di connessione

Connessione Telnet fra 10.6.1.9 e 10.6.1.2 catturata con **tcpdump**
porta client 4548 - porta server 23 (telnet)

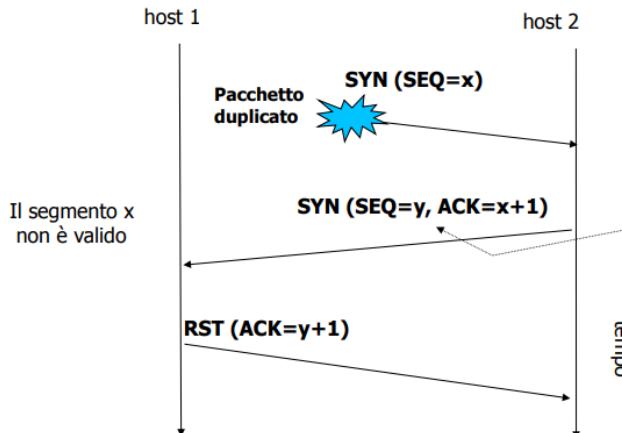
```
10.6.1.9.4548 > 10.6.1.2.23: S 2115515278:2115515278(0)
win 32120 <mss 1460,nop,nop,sackOK,nop,wscale 0> (DF)
opzioni da negoziare
(es. max segment size mss)

10.6.1.2.23 > 10.6.1.9.4548: S 1220480853:1220480853(0)
ack 2115515279 win 32120<mss1460,nop,nop,sackOK,nop,wscale
0> (DF)

10.6.1.9.4548 > 10.6.1.2.23: . ack 1220480854 win 32120
(DF)
```

* tcpdump -S -n -t \dst 10.6.1.2 and src 10.6.1.9\ or \dst 10.6.1.9 and src 10.6.1.2\)

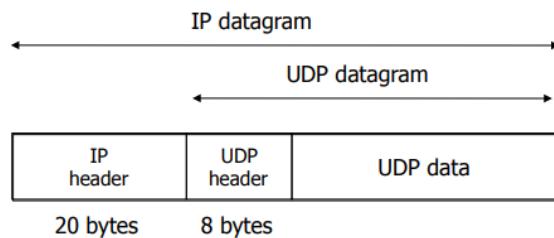
Nella rete possono generarsi dei **pacchetti duplicati** e, nel caso in cui succeda, verranno eseguite le seguenti operazioni:



Come abbiamo detto in precedenza, il numero di sequenza, espresso su 32 bit, corrisponde al valore di un clock. Di conseguenza lo stesso numero di sequenza, su 32 bit, non può essere generato nuovamente prima di circa 4 ore. A causa del TTL dei pacchetti IP, però, segmenti con lo stesso numero di sequenza non possono coesistere sulla rete.

UDP

Ogni operazione di output produce esattamente un pacchetto UDP che comporta l'invio di un pacchetto IP.



UDP è un protocollo che **non garantisce affidabilità di consegna**, ma **richiede meno overhead di una connessione TCP**. Se il pacchetto UDP eccede la **Maximum Transfer Unit (MTU)** della rete, esso viene frammentato. Il protocollo UDP è un **servizio di consegna best effort**, quindi si impegna a consegnare i dati, ma è possibile che essi vengano persi, duplicati o consegnati nell'ordine sbagliato. Anche in caso di errori, lo **UDP non opera alcun recupero**, quindi il segmento errato viene scartato o consegnato all'applicazione segnalando che presenta errori. Lo UDP, inoltre, è un **servizio senza connessione**, quindi **non vi è handshaking tra mittente e destinatario del segmento UDP**. A differenza di TCP, UDP:

- non introduce ritardo per instaurare la connessione;
- non mantiene lo stato di connessione, in quanto non deve gestire buffer di invio/ricezione, parametri per il controllo della congestione, numeri di sequenza, ecc.;
- produce un minor overhead (header TCP 20 byte contro gli 8 dell'header UDP);
- non controlla la congestione (il controllo della congestione effettuato dal TCP può avere un severo impatto su applicazioni real-time).

DNS

Il DNS, Domain Name System, è un **sistema gerarchico di naming utilizzato su Internet per tradurre gli indirizzi IP numerici delle risorse** (come server web) in **nomi di dominio leggibili da parte degli esseri umani**. In pratica il DNS riceve come dato il nome di un computer e restituisce un indirizzo IP tramite il **distributed**

lookup, un metodo che permette di distribuire le richieste di risoluzione dei nomi di dominio su più server DNS in tutto il mondo. I **domini** sono segmenti alfanumerici separati da punti (come ad es. www.dii.unisi.it o anche www.cisco.com).

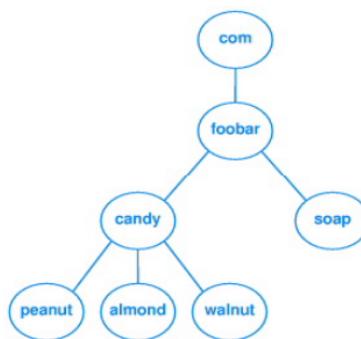
Un'emissione di un **nome di dominio** avviene quando un'organizzazione sceglie un nome desiderato, che deve essere ovviamente unico. Dopodiché avviene la **registrazione del dominio mediante un intermediario autorizzato ad interagire con l'autorità centrale**, che è quella che gestisce i **domini di primo livello (TLD – Top Level Domain)**. Il dominio viene associato ad una **specifico estensione**, in base alla natura del sito web o alle preferenze del registrante:

- .com, organizzazione commerciale;
- .edu, università (ad es. americana);
- .gov, governo americano;
- .mil, gruppo militare;
- .net, grosso network provider;
- .org, organizzazione (ad es. IEEE);
- .arpa, es. dominio temporaneo;
- .int, organizzazione internazionale;
- .it, codice paese (in questo caso Italia).

Durante la registrazione del dominio, bisogna tenere conto che potrebbero esserci restrizioni legate a **trademarks** (marchi registrati) e/o **copyright** (diritti d'autore), oltre ai **vincoli di legge internazionale**.

Es.

DNS - esempio



In questo caso il dominio di primo livello sarebbe .com, il dominio di secondo livello foobar, i domini di terzo livello candy e soap, ecc. **Non c'è standard universalmente riconosciuto**, infatti ogni organizzazione sceglie i propri nomi, sempre nel rispetto dei vincoli precedentemente discussi. Inoltre i nomi, anche all'interno di una stessa organizzazione, non devono necessariamente seguire una regola; ci possono essere ad esempio scelte diverse per i diversi gruppi

dell'organizzazione. DNS utilizza il protocollo UDP a meno che non ci sia una grande quantità di dati da trasferire; in tal caso si passa al protocollo TCP.

DNS client/server

Il sistema DNS coinvolge **due componenti principali**: i **client DNS**, anche noti come **resolver**, ed i **server DNS**, organizzati in una struttura gerarchica che consente una distribuzione efficiente delle responsabilità. I **server DNS sono collegati tra loro in modo da consentire il passaggio delle richieste di risoluzione dei nomi di dominio attraverso la gerarchia, dall'origine al destinatario**. La base della gerarchia è costituita dai **root server**: quando un **server DNS riceve una richiesta per la risoluzione di un nome di dominio, ed è sprovvisto dell'informazione nella sua cache, consulta i root server per ottenere indicazioni sul dove trovare i server responsabili del dominio specifico**. Oltre ai **root server** esistono **server DNS che sono autoritari per specifici domini o sottodomini**. In pratica il **server DNS** riceve come informazione dai **root server** la posizione dei **server autoritari** per il dominio di interesse, in modo da continuare nel percorso gerarchico fino alla destinazione. **Quando una query DNS (che è ricorsiva poiché la richiesta risale nei vari livelli gerarchici) fallisce, significa che il sistema non è in grado di risolvere un nome di dominio in un indirizzo IP o recuperare altre informazioni associate a quel dominio e quindi restituisce un messaggio di errore**.

SMTP (Simple Mail Transfer Protocol)

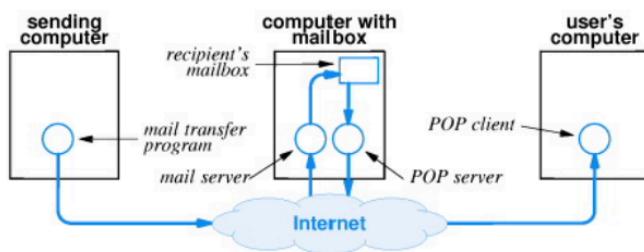
Il SMTP è un **protocollo utilizzato per la trasmissione di email attraverso una rete**, che si trova all'ultimo livello (livello applicazione) del modello TCP, e viene definito in RFC 821. Viene **utilizzato tra il programma di trasferimento di email sul computer del mittente ed il mail server sul computer del destinatario**, e permette di specificare come il client (ossia il programma di trasferimento di email sul computer del mittente) interagisce con il mail server (sul computer del destinatario). Il protocollo, inoltre, dettaglia come i destinatari vengono specificati e come il messaggio viene trasferito.



Computer senza mail servers – POP (Post Office Protocol)

Esistono dei computer senza un proprio mail server, e tipicamente si tratta di piccoli personal computer non sempre connessi alla rete. Per ricevere email **l'utente deve possedere una mailbox su un apposito server** e deve accedere alla stessa (mailbox). Da questi computer viene utilizzato il **Post Office Protocol (POP)**, definito in **RFC 1939**.

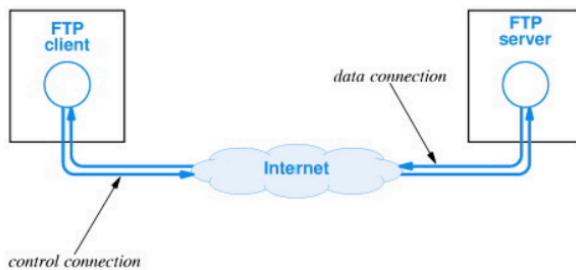
POP



File Transfer Protocol (FTP)

Il File Transfer Protocol (FTP) è un **protocollo di rete utilizzato per il trasferimento di file da un host ad un altro**, come suggerisce il nome, mediante modello TCP. Il FTP supporta file sia di tipo binario che di tipo ASCII, ha un ampio set di comandi e fino al 1995 era la maggiore sorgente di pacchetti su Internet. L'FTP fu definito in **RFC 959**.

FTP Illustrated



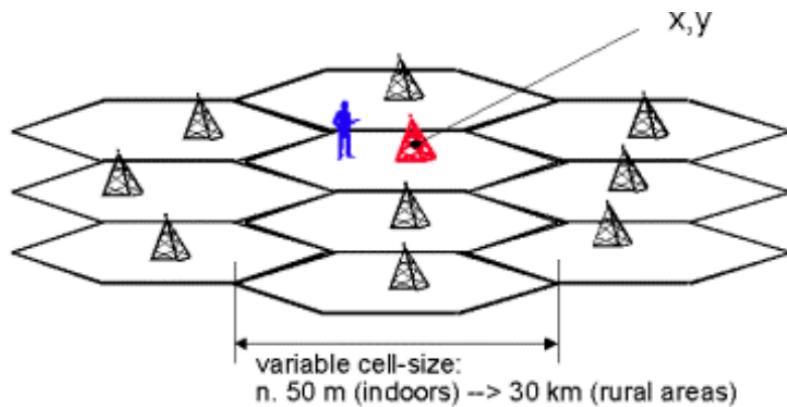
Hypertext Transfer Protocol (HTTP)

L'HTTP, Hypertext Transfer Protocol, è un protocollo **di comunicazione utilizzato per il trasferimento di informazioni su Internet**. In pratica i **server web** “servono” **pagine web**, ossia forniscono le pagine richieste dai client, che sono

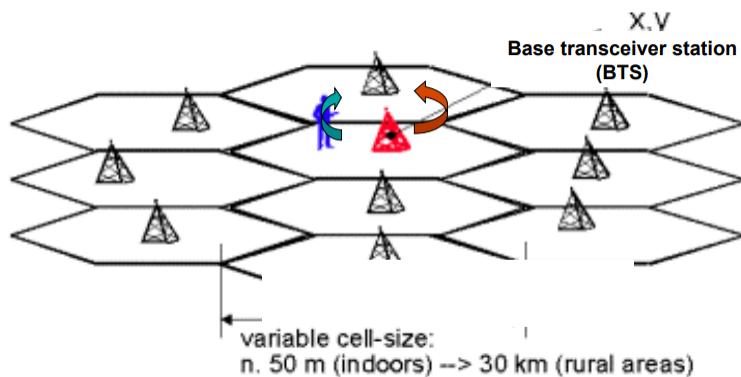
tipicamente rappresentati dai browser. HTTP, di solito, utilizza la **porta 80 come porta predefinita** per le comunicazioni tra server e client. **Questo protocollo si basa sul protocollo TCP** per quanto riguarda la comunicazione tra client e server, quindi viene stabilita una connessione tra i due per consentire il trasferimento affidabile dei dati. HTTP opera al livello più alto dell'architettura TCP, ossia il livello applicazione, e fu definito, nella sua versione v1.1, in **RFC 2068**.

Telefonia cellulare (Reti cellulari)

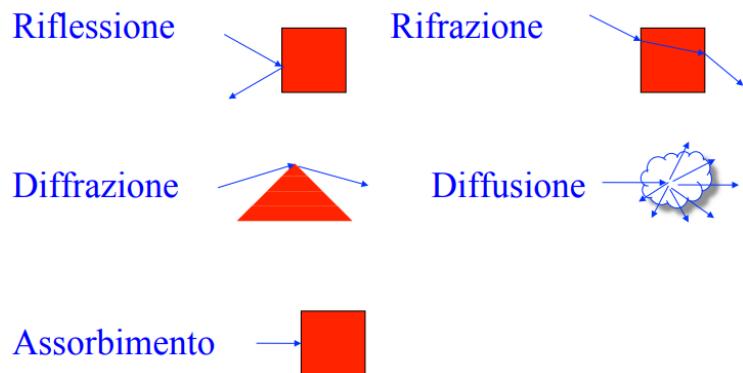
Ogni area geografica è suddivisa in più celle, ciascuna servita da un trasmittitore di debole potenza su frequenze destinate ad essere riutilizzate in celle non contigue.



Quando ci si muove tra le celle, il sistema cellulare commuta agganciando automaticamente il segnale più intenso in quel punto.

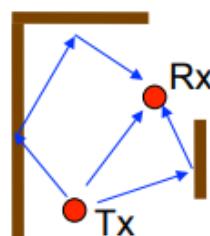


In funzione delle caratteristiche fisiche dell'ambiente attraversato e della frequenza utilizzata, **un'onda elettromagnetica trasmessa nello spazio libero subirà fenomeni di diverso tipo**, come



Un'onda radio, ossia un tipo di onda che occupa una porzione specifica dello spettro elettromagnetico utilizzata anche nelle tecnologie di telecomunicazione, è soggetta a fenomeni di:

- **attenuazione**, ossia la potenza del segnale si riduce
 - all'aumentare della distanza (pathloss, perdita di percorso);
 - a causa dell'attraversamento di ostacoli;
 - per assorbimento;
- **interferenza**, quando il segnale potrebbe subire modifiche a causa di
 - altre sorgenti che utilizzano la stessa banda;
 - rumore elettromagnetico ambientale;
- **cammini multipli**, ossia **repliche dello stesso segnale seguono percorsi diversi e arrivano sfasate** al ricevitore a causa di fenomeni di riflessione, diffrazione o diffusione.



Poiché il mezzo radio deve essere condiviso tra diversi sistemi di trasmissione, occorre ripartire le frequenze in modo tale che i diversi servizi non utilizzino le stesse regioni dello spettro elettromagnetico. Agli utenti di uno stesso servizio è consentito l'**accesso simultaneo al mezzo trasmittivo mediante tecniche di divisione**:

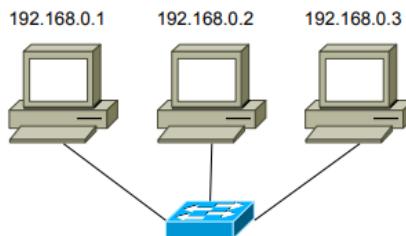
- **di frequenza (FDMA)**, che è un'applicazione specifica di FDM nel contesto dell'accesso multiplo), dove **utenti diversi trasmettono contemporaneamente su frequenze diverse**;
- **di tempo (TDMA)**, che è un'applicazione specifica di TDM nel contesto dell'accesso multiplo), dove **utenti diversi trasmettono in tempi diversi sulla stessa frequenza**;
- **di codice (CDMA)**, dove **utenti diversi trasmettono contemporaneamente sulla stessa frequenza utilizzando codici diversi**.

NAT (Network Address Translation)

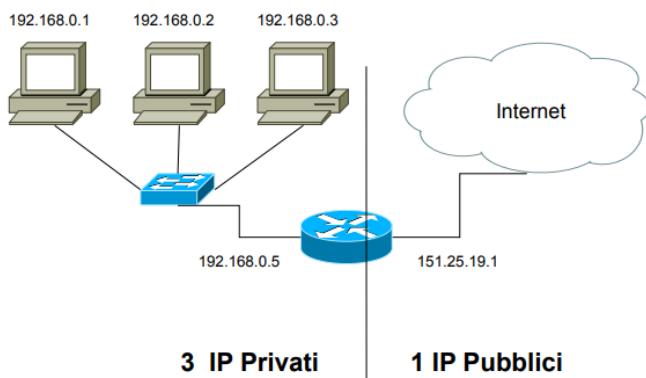
Il NAT, Network Address Translation, è una **tecnologia utilizzata nelle reti informatiche per consentire a più dispositivi di condividere un singolo indirizzo IP pubblico**. Nella realtà, si possono avere un numero N di macchine, ma un numero M di indirizzi IP, con $M < N$. Si rende pertanto necessario un sistema per consentire alle restanti macchine di collegarsi ad Internet. Per semplicità, di seguito, supponiamo di disporre di un solo indirizzo IP. Nell'assegnazione degli indirizzi IP, vi sono alcuni blocchi di indirizzi, descritti in RFC 1918, che vengono allocati per **reti di uso privato**:

- La rete di classe A 10.0.0.0 – 10.255.255.255**
- Le 16 reti di classe B 172.16.0.0 – 172.31.255.255**
- Le 256 reti di classe C 192.168.0.0 – 192.168.255.255**

Non esiste in Internet una rete con questi blocchi di indirizzi, e i router della rete sono programmati, o almeno dovrebbero esserlo, per non instradare pacchetti provenienti da questi indirizzi. Prendiamo ora come esempio una semplice Intranet senza uscita su Internet.



Assegniamo l'unico indirizzo IP pubblico disponibile ad un router con due interfacce di rete, una collegata all'intranet e una su internet (extranet).

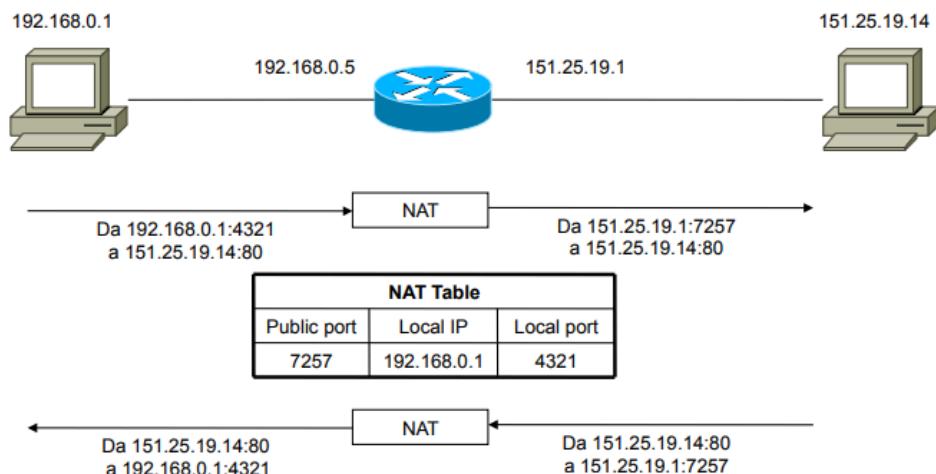


In questo caso, un pacchetto proveniente da 192.168.0.1 non può essere mandato su internet, in quanto verrebbe automaticamente cancellato dal primo router in rete e, se anche non succedesse, la risposta non arriverebbe a destinazione. L'unica via d'uscita, per un pacchetto, è l'indirizzo IP 151.25.19.1. Il **router riceve una richiesta di connessione** da 192.168.0.1 verso il nodo X, dopodichè,

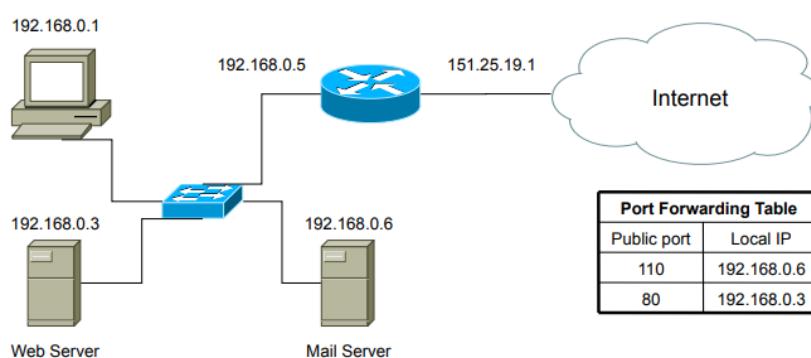
sostituendosi alla macchina locale, **effettua la richiesta con il suo indirizzo IP pubblico**. Una volta ricevuta risposta, la inoltra sulla rete privata tramite il suo indirizzo IP privato e **procede in questo modo per tutta la durata della connessione**. L'intero **meccanismo del NAT si regge interamente sull'utilizzo delle porte**. Quando il router riceve una richiesta di instradamento di connessione, effettua la connessione con il proprio indirizzo IP pubblico e **memorizza in una tabella**:

- l'**indirizzo IP (privato)** e la **porta sorgente locale**;
- la **porta locale con cui effettua la connessione** (relativa all'indirizzo IP pubblico del router).

Così facendo, il router è in grado, quando arriva un pacchetto di risposta sulla porta locale, di sapere a quale macchina instradarlo.



Se una macchina di rete interna fa partire una connessione, è possibile, per il NAT, tracciare la connessione e mantenerla attiva fino alla fine. Se, tuttavia, dall'esterno arrivasse una richiesta di connessione ad una porta che non è stata precedentemente allocata da una macchina interna tramite una connessione, il NAT non sa a quale delle macchine interne girarla, e di conseguenza scarterà il pacchetto. E' però possibile configurare il NAT a priori, per far sì che alla richiesta di una connessione dall'esterno, su una determinata porta, il NAT invii il pacchetto ad una macchina predefinita all'interno della rete: tale configurazione prende il nome di **port forwarding**.



Streaming audio/video stored

Della categoria degli streaming audio/video stored fanno parte **video e audio pre-registrati**, come film, show, serie tv, ecc. Le **caratteristiche chiave** di questa tipologia di multimedia networking applications sono lo **streaming**, l'**interattività** e la **riproduzione continua**. Per quanto riguardo l'**archiviazione**, quindi lo storing, abbiamo **due possibilità**:

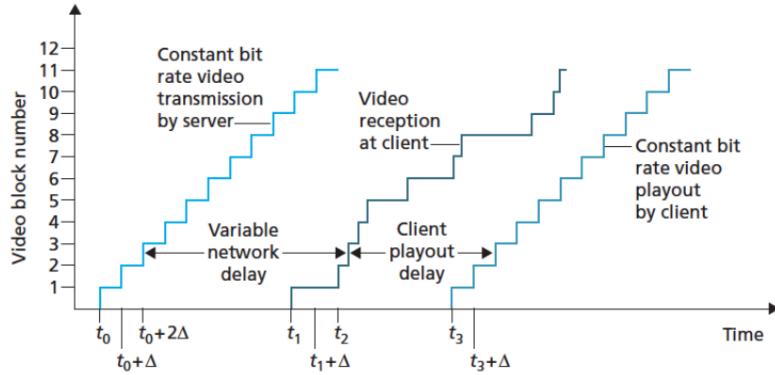
- **Content Distribution Network (CDN)**, che è un **sistema distribuito di server che collaborano per distribuire contenuti multimediali**. Questo sistema garantisce **ottime prestazioni dello streaming, riducendo la latenza** (ossia il ritardo che si genera end-to-end, quindi tra la generazione di un pacchetto da parte del mittente e la ricezione dello stesso da parte del destinatario) e garantendo una **maggior affidabilità**. Le CDN posizionano **copie dei contenuti in server distribuiti in varie località geografiche**, consentendo agli utenti di accedere ai dati dal server più vicino, in modo da avere la **miglior esperienza di streaming possibile**;
- **Peer-to-Peer (P2P)**, che invece implica la **condivisione diretta di contenuti tra gli utenti** senza la necessità di passare attraverso un server centrale. In questo caso, i **clienti che ricevono lo streaming fungono anche da fornitori di contenuti per altri utenti**. Questo approccio può **ridurre il carico sui server centrali, migliorando l'efficienza complessiva del sistema**, ma risulta **più complesso da gestire** e, solitamente, garantisce **prestazioni minori per quanto riguarda la qualità dello streaming**.

Per quanto riguarda lo streaming video, i video pre-registrati sono salvati su dei server, ai quali gli utenti inviano richieste, per vedere contenuti on demand.

Esistono **categorie diverse di sistemi video di streaming**:

- **UDP streaming**;
- **HTTP streaming**;
- **adaptive HTTP streaming**.

Queste categorie hanno **caratteristiche comuni**, come il **buffering lato client**, ossia memorizzare temporaneamente i dati video nel dispositivo del destinatario (client). Ciò consente di mitigare gli effetti delle **variazioni dei ritardi end-to-end**, con end-to-end che si riferisce all'intero percorso che i dati seguono dal mittente al destinatario, **variando la quantità della larghezza di banda** a disposizione tra server e client.



Ritardo di riproduzione del client in video streaming

Voice/video-over-IP (VoIP)

Della categoria voice/video-over-IP fanno parte i **servizi voce e video in real-time su internet**, come possono essere servizi di **chiamate vocali**, o anche di **videochiamate** (Whatsapp, Skype, Discord, ecc.). Per gestire queste applicazioni bisogna effettuare delle considerazioni piuttosto precise per quanto riguarda il **timing**, in quanto applicazioni di questo tipo sono altamente **sensibili ai ritardi**, ma allo stesso tempo la comunicazione in tempo reale è alla base delle stesse (applicazioni). Questo tipo di applicazioni, inoltre, **tollerano piccole perdite di informazioni**, ma è comunque di fondamentale importanza minimizzare il numero. Come abbiamo detto, di questa categoria fa parte la **telefonia internet**, che viene appunto chiamata Voice-over-IP (VoIP). **IP fornisce un servizio best-effort**, quindi prova a consegnare il maggior numero di pacchetti possibili, ma ci possono comunque essere ritardi e una percentuale di pacchetti persi.

Perdita di pacchetti

I segmenti UDP, come detto, sono incapsulati in un pacchetto IP, il quale attraversa la rete e passa attraverso i buffers dei router. E' possibile che uno o più di questi buffers sul tragitto tra il mittente ed il destinatario siano pieni, ed in questo caso il pacchetto IP può essere scartato. **Si potrebbero eliminare le perdite tramite l'invio di pacchetti utilizzando TCP**, e non UDP, ma **questa soluzione è inaccettabile per applicazioni conversazionali come le VoIP**, che richiedono uno scambio audio real-time, in quanto verrebbe meno la velocità di trasmissione, che è alla base di questo tipo di applicazioni. **Una delle poche eccezioni** è rappresentata da Skype, che utilizza UDP, ma se un utente di trova dietro un NAT o un firewall che bloccano i segmenti UDP, Skype può commutare automaticamente su TCP per garantire la continuità della comunicazione. La **percentuale di perdita tollerata**, nelle applicazioni VoIP, è tra l'1 ed il 20%, in quanto vi è occultamento

della perdita vista la velocità di trasmissione. La percentuale dal 10 al 20%, però, è sintomo di **qualità audio scadente**.

Ritardo end-to-end

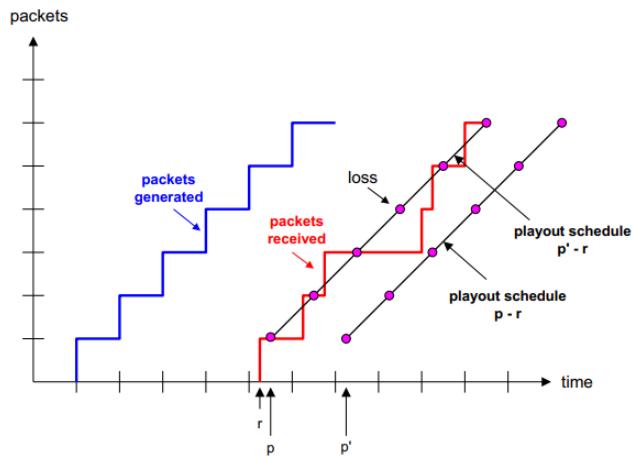
Il ritardo end-to-end, nel contesto delle applicazioni VoIP, si riferisce al **ritardo di trasmissione, di elaborazione e di coda, accumulati tra l'invio di un pacchetto audio da parte di un interlocutore e la sua ricezione da parte dell'altro interlocutore**. Ritardi end-to-end, sempre per quanto riguarda questo tipo di applicazioni, **più piccoli di 150 msec non sono percepiti dall'uomo**, ritardi tra i **150 e 400 msec possono essere accettabili**, ma non ideali, mentre ritardi oltre i **400 msec possono compromettere la conversazione e l'utilizzo del servizio**.

Packet jitter

Il packet jitter si riferisce alla **variabilità dei ritardi di arrivo dei pacchetti in una rete**. Possiamo dire che si tratti della misura di quanto il tempo, da quando un pacchetto è generato alla sorgente a quando viene ricevuto dal destinatario, possa variare da pacchetto a pacchetto. Un esempio del perché vi possano essere ritardi diversi tra i pacchetti è la presenza di code diverse per differenti router. Infatti in questo modo, **se c'è congestione di rete o fluttuazioni nel carico, i pacchetti possono essere accodati nei buffer e possono essere rilasciati in modi diversi, causando variabilità nei tempi d'arrivo**.

Ritardo di riproduzione

Per rimuovere gli effetti del packet jitter, o almeno mitigarli, esistono **due approcci principali**: il ritardo di riproduzione fisso ed il ritardo di riproduzione adattivo. Per quanto riguarda il **ritardo di riproduzione fisso, si stabilisce un ritardo costante prima di riprodurre i pacchetti ricevuti**, in modo che se anche i pacchetti arrivano in modo variabile, vengono messi in attesa prima della riproduzione. L'utilizzo di un ritardo fisso, però, **introduce una latenza costante**, che in applicazioni di tipo VoIP **non è ideale**.



Ritardo di riproduzione fisso

Il **ritardo di riproduzione adattivo**, invece, è ovviamente un **approccio più flessibile**, che appunto si adatta dinamicamente ai cambiamenti nelle condizioni di rete. In questo caso la **rete monitora la variabilità dei tempi di arrivo dei pacchetti, regolando il ritardo di riproduzione di conseguenza**. Se la **variabilità è bassa**, il ritardo di riproduzione può essere ridotto per ridurre la latenza complessiva, mentre, al contrario, se la **variabilità aumenta**, il ritardo di riproduzione può essere aumentato per gestire meglio il packet jitter.

Ritardo di riproduzione adattivo

t_i = timestamp del pacchetto i-esimo
 r_i = tempo di ricezione del pacchetto i
 p_i = tempo di inizio riproduzione
 u = costante = es. 0.01
 K = costante per ritardare l'inizio di riproduzione (es. 4)

$$d_i = (1 - u) d_{i-1} + u (r_i - t_i)$$

Stima del ritardo medio della rete

$$v_i = (1 - u) v_{i-1} + u | r_i - t_i - d_i |$$

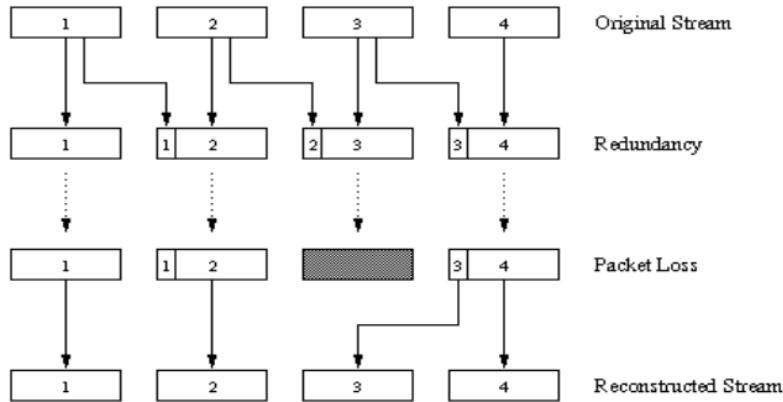
Stima della deviazione media del ritardo di rete

$$p_i = t_i + d_i + K v_i$$

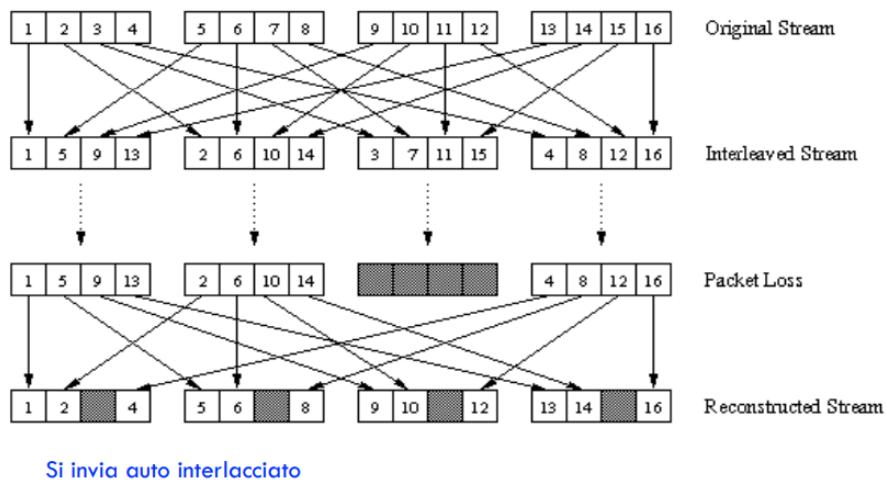
Riproduzione dei pacchetti

Correzioni anticipate degli errori

Il **Forward Error Correction, FEC**, è una **tecnica utilizzata per correggere gli errori** che possono verificarsi durante la trasmissione di dati attraverso una rete. In pratica **viene inviato un flusso audio a bassa qualità come informazione aggiuntiva rispetto ai dati originali**, e di conseguenza **richiede una maggiore larghezza di banda**, che non è possibile avere in tutte le applicazioni.

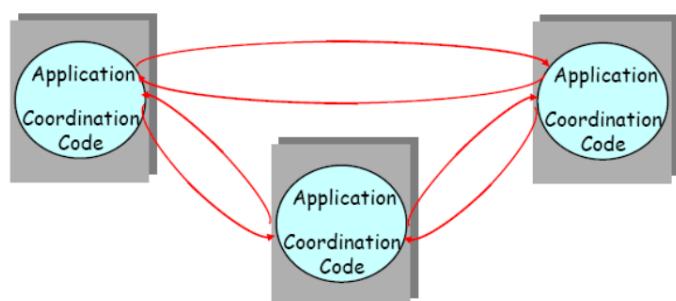


Esiste anche un'altra tecnica, chiamata **interlacciamento**, che si ottiene dividendo i pacchetti originali in diverse sottounità, le quali vengono inviate in pacchetti diversi. Ciò significa che se un pacchetto interlacciato viene perso, non viene perso l'intero pacchetto di dati iniziali ma solo una sottounità.

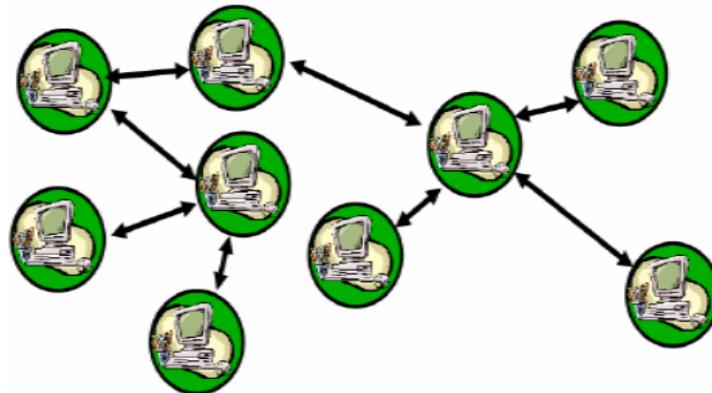


Modello Peer-to-Peer (P2P)

Nel modello P2P, le applicazioni sono basate, appunto, su **processi peer**, i quali non sono né client né server. Nel modello client/server ci sono server centrali che forniscono servizi o risorse ai client, mentre nel modello P2P, i **peer sono più simili tra loro e possono svolgere sia il ruolo di client che quello di server**.



Questo tipo di modello si è sviluppato verso la metà degli anni 2000, quando si cominciò ad avere la percezione che Internet stesse seguendo schemi prevedibili e centralizzati. L'emergere di nodi Internet connessi in modo sporadico, come telefoni cellulari, laptop, ecc., inoltre, contribuì allo sviluppo del modello P2P. Questo perché dispositivi del genere potevano scambiare informazioni, anche senza conoscersi tra loro. Il modello P2P, in realtà, rappresenta il più vecchio tipo di architettura nel mondo delle comunicazioni, ma si sviluppò solo a metà degli anni 2000, come abbiamo detto, in campo informatico. Per quanto riguarda il mondo delle comunicazioni, **i telefoni sono un esempio di P2P, in quanto ognuno può chiamare e ricevere chiamate indipendentemente dall'esistenza di un server centrale, ma anche il routing in Internet ne è un esempio, in quanto coinvolge una comunicazione tra nodi che possono essere considerati pari nella rete.** Le **tecniche P2P stanno riportando Internet alla sua visione originale**, nella quale ogni utente non solo consuma contenuti, ma contribuisce anche attivamente alla creazione e alla condivisione.



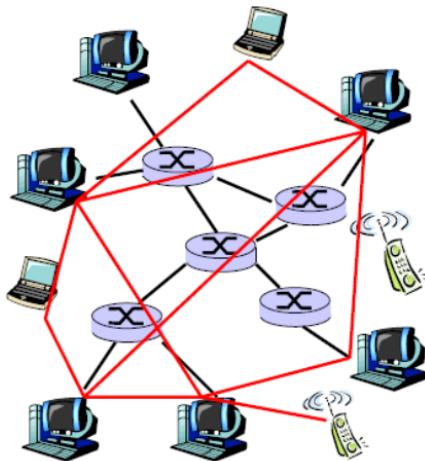
Il modello P2P presenta diverse caratteristiche principali:

- **è formato da una rete di nodi con capacità/responsabilità equivalenti**, ossia ogni nodo può agire sia da client che da server;
- i nodi, che come abbiamo detto possono agire sia da client che da server, **vengono chiamati servants**;
- **scambio diretto di informazioni tra host**, senza passare attraverso un server centralizzato;
- **utilizza una rete transitoria** che permette ad un gruppo di computer di connettersi e collaborare tra loro, condividendo risorse;
- i peer connessi costruiscono una **overlay network** (rete sovrapposta) **virtuale**, al di sopra della rete fisica (es. di overlay sono i CDN o anche le applicazioni P2P).

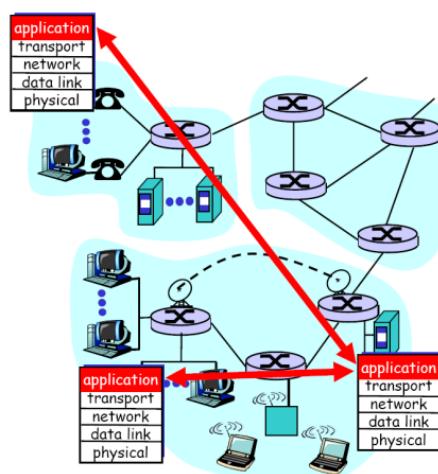
Overlay networks

Un overlay network è un **insieme di connessioni logiche tra hosts**, che sono dispositivi collegati a Internet, i quali **non devono fisicamente essere vicini tra**

Iloro, in quanto possono essere collegati in modo logico, appunto, tramite le overlay network. Le overlay network possono essere **strutturate o non strutturate**, in base a quanto sono organizzate. In questo tipo di reti, la **manutenzione risulta una grande problematica**: devono infatti essere utilizzati sistemi di monitoraggio per quanto riguarda le connessioni logiche e la gestione delle risorse.



Le overlay networks sono **completamente implementate nel livello applicazione**, e ciò significa che **tutte le operazioni e le funzionalità sono gestite da software applicativo** anziché essere incorporate direttamente nei protocolli di rete a livello più basso. Gli sviluppatori hanno **grande flessibilità** per quanto riguarda la progettazione di overlay networks, infatti possono definire la topologia e i protocolli di comunicazione di queste reti in base alle loro esigenze. Per quanto riguarda la rete fisica sottostante, invece, gli sviluppatori non se ne devono preoccupare, in quanto, come abbiamo detto, le overlay networks sono completamente gestite a livello applicazione.



Classificazioni dei sistemi P2P

La **classificazione dei sistemi P2P** può avvenire **in base al grado di decentralizzazione**:

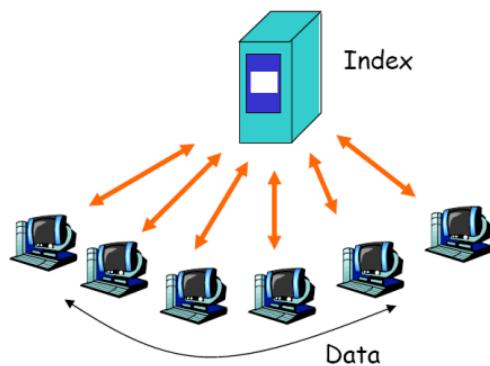
- hybrid decentralized P2P;
- purely decentralized P2P;
- partially centralized P2P,

oppure in base al grado di strutturazione:

- structured P2P;
- loosely structured P2P;
- unstructured P2P.

Hybrid decentralized P2P

Nei sistemi hybrid decentralized P2P, viene utilizzato un **server centrale** per facilitare l'interazione tra i peer ed eseguire le ricerche per l'identificazione dei nodi (peer) della rete che ad es. possiedono la risorsa desiderata. La presenza di un server centrale, però, porta la rete ad avere un **singolo punto di fallimento**: infatti se il server centrale si guastasse, l'intera rete potrebbe subire un'interruzione significativa. Un altro problema risiede nella **scalabilità limitata**, infatti un numero crescente di peer comporterebbe anche un carico maggiore da gestire per il server centrale. Un esempio di questo tipo di sistemi è Napster.

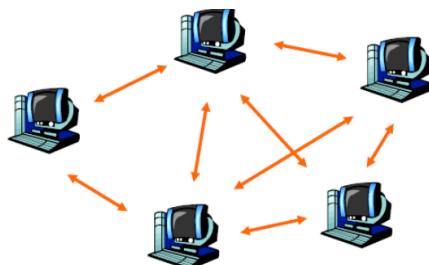


Purely decentralized P2P

Nei sistemi purely decentralized P2P, **non viene utilizzato un server centrale** e tutti i nodi, ossia i servents, eseguono le stesse attività. In questo tipo di sistemi esistono però **diversi problemi**:

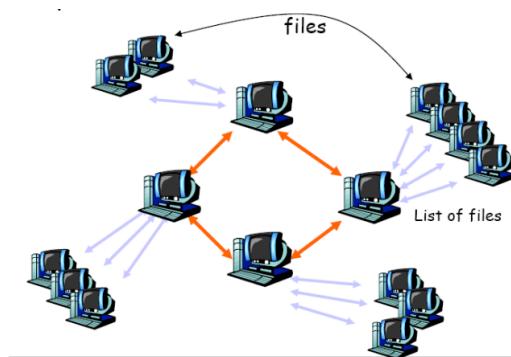
- **consistenza dei dati**, in quanto ogni servent può aggiornare le informazioni indipendentemente e questo, ovviamente, può portare a conflitti e inconsistenze nei dati;
- **gestibilità**, in quanto in assenza di un server centrale è più complesso gestire la rete;
- **sicurezza**, perché questo tipo di sistemi può essere più vulnerabile a minacce di sicurezza, in quanto non c'è un punto centrale su cui concentrare gli sforzi di protezione;

- **overhead delle comunicazioni**, in quanto avvengono direttamente tra i nodi, senza coordinazione centrale, e questo può richiedere più risorse rispetto ad un sistema centralizzato.



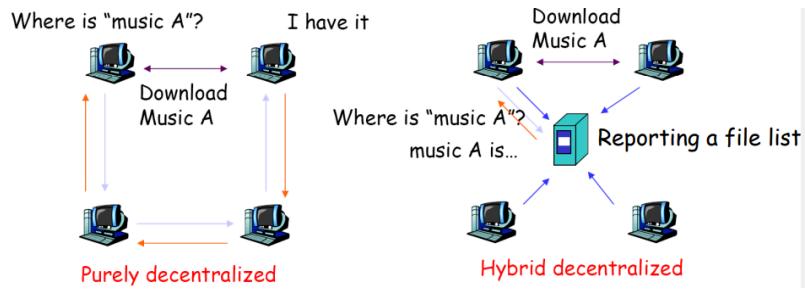
Partially centralized P2P

Nei sistemi partially centralized P2P, alcuni nodi, che prendono il nome di **supernodi**, assumono un ruolo più importante, centrale. I supernodi **agiscono come punti centrali all'interno della rete** e possono svolgere attività di indicizzazione, coordinare le ricerche o gestire le connessioni tra i peer. Questi supernodi **migliorano l'efficienza della rete**, ma la centralizzazione parziale introduce un **certo grado di dipendenza da questi nodi**, creando un **potenziale punto di vulnerabilità**.



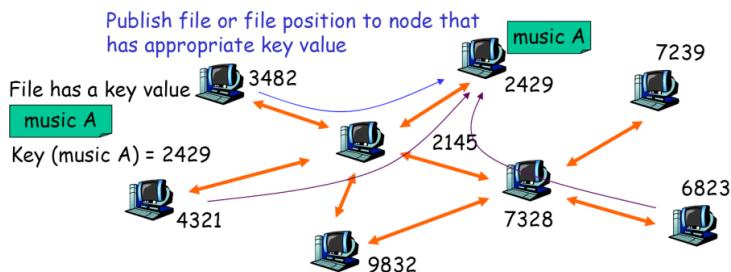
Unstructured P2P

Nei sistemi unstructured P2P, **ciascun peer è responsabile della gestione dei propri dati e la posizione in cui un dato è archiviato non segue uno schema predefinito**. Poiché la posizione dei dati non è nota a priori, vengono utilizzati meccanismi di broadcasting per la ricerca. La posizione dei dati, inoltre, non è correlata alla topologia della rete, e questo ovviamente rende la **rete più flessibile ma meno efficiente nelle ricerche**. Un esempio di questo sistema unstructured P2P è Napster.



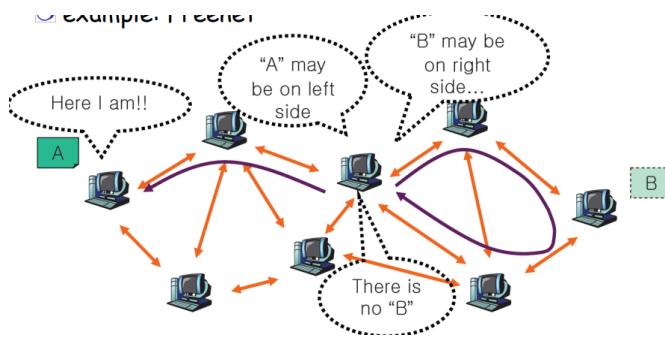
Structured P2P

Nei sistemi structured P2P, la **topologia della rete è organizzata in modo preciso e i file sono posizionati in specifiche posizioni all'interno della struttura**, di solito seguendo un criterio basato su **funzioni di hash** o altri algoritmi. In questi sistemi, tra l'identificatore di un file e la posizione precisa nella rete, esiste una mappatura che facilita la ricerca ed il recupero efficiente dei dati. Questo tipo di sistemi, però, ha come **svantaggio una maggiore complessità di gestione e manutenzione**.



Loosely structured P2P

Nei sistemi loosely structured P2P, si trova un **compromesso tra i sistemi structured P2P e quelli unstructured P2P**. In questo contesto, infatti, la **disposizione dei dati è influenzata da suggerimenti di routing**, ma non è completamente specificata come nelle reti P2P strutturate. Ovviamente questo tipo di sistema ha sia i vantaggi che gli svantaggi degli altri due, ma in minor parte. Rispetto ai sistemi structured P2P, infatti, consente una **maggior flessibilità e comunque una certa organizzazione**, anche senza la loro rigidità (dei sistemi structured P2P). La mancanza di una struttura rigida, d'altra parte, comporta una **minore efficienza nelle ricerche, ma una complessità di gestione minore** rispetto ai sistemi structured P2P e maggiore rispetto ai sistemi unstructured P2P.



Esempio di classificazione di applicazioni P2P

	Unstructured Networks	Loosely Structured Networks	Structured Networks
Hybrid Decentralized	Napster		
Pure Decentralized	Gnutella	Freenet	Chord, CAN, Tapestry
Partially Centralized	KaZaa, new-Gnutella		

Condivisione di file P2P (file sharing)

La condivisione di file P2P è stata una cosiddetta **killer application**, ossia una delle applicazioni più significative e di successo nella storia di Internet. **Vantaggi** di questa applicazione:

- **area di scambio potenzialmente illimitate**, che consentono agli utenti di condividere una vasta gamma di contenuti;
- **grande spazio di archiviazione sicura**, in quanto vengono utilizzate le tecniche di duplicazione e ridondanza dei file su più nodi, nella rete P2P, per assicurare che i file siano sempre conservati, anche nel caso in cui alcuni nodi dovessero essere offline;
- **anonimato**, poiché essendo i file distribuiti su diversi nodi, senza una traccia diretta dell'origine, autori ed editori vengono preservati;
- **gestibilità**, in quanto i file e le risorse sono distribuite tra gli utenti, riducendo la dipendenza da server centralizzati.

Per quanto riguarda invece gli **svantaggi**, abbiamo:

- **consumo di larghezza di banda di rete**, in quanto i dati vengono scambiati direttamente tra gli utenti, e questo può influire sulle prestazioni della rete;
- **sicurezza**, perché questo tipo di sistemi può essere più vulnerabile a minacce di sicurezza, in quanto non c'è un punto centrale su cui concentrare gli sforzi di protezione;

- **capacità di ricerca**, che è limitata in quanto non si tratta di un sistema centralizzato, ossia con un server centrale.

Esempi di applicazioni P2P per quanto riguarda il file sharing sono Napster e BitTorrent.