E.I. - A.A. 2020/21

Reti

Riassunto slide + approfondimenti

Ciao, questo è un riassunto che ho fatto PER ME STESSO quando ho dovuto studiare per l'esame, lo condivido sperando possa essere utile a qualcuno. Le informazioni al suo interno sono un misto tra quelle presenti nelle slide del professore e quelle presenti nel libro; non mi assumo nessuna responsabilità riguardante la loro correttezza o precisione.

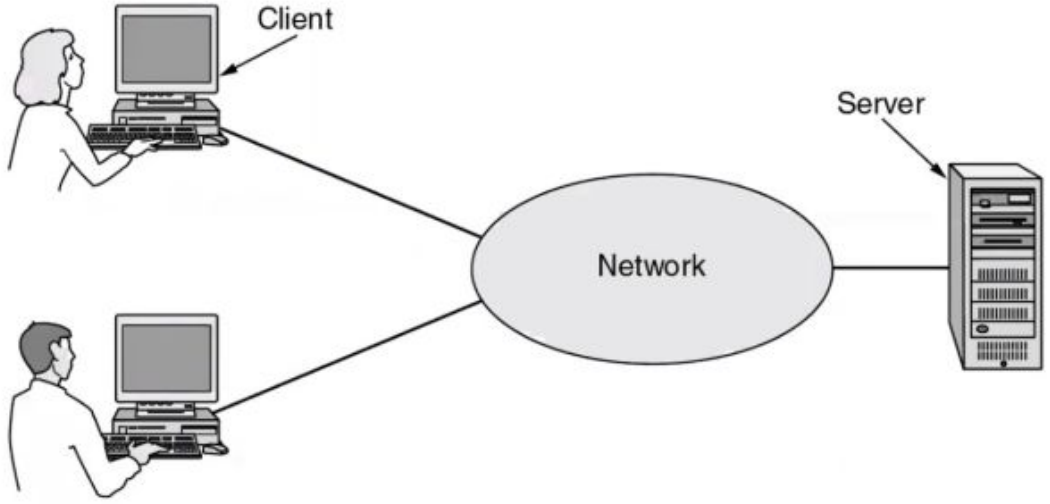
Cos’è una rete?

“Si ha una rete ogni volta che c’è un grafo nascosto”: scegliamo questa definizione perchè è molto generale e non usa implicitamente il concetto di rete per definire la rete stessa. Alcuni esempi di reti sono Internet, il nostro computer con le periferiche, il bluetooth ecc.

Classificazione delle reti

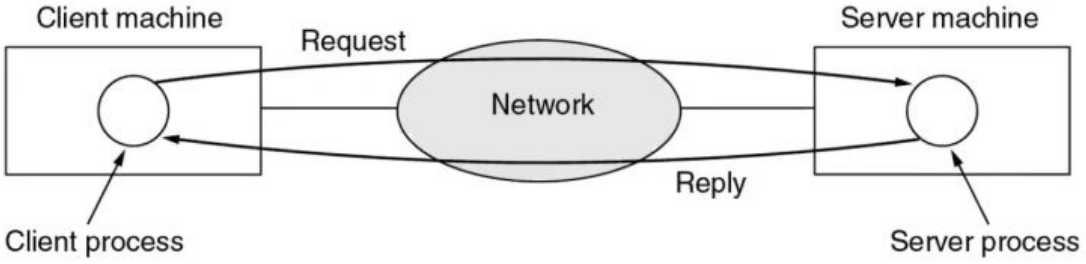
Le reti si possono classificare secondo vari criteri.

Modello Client-Server



I dati sono memorizzati in ***server***, a cui gli impiegati (ipotetici) accedono tramite ***client*** (i loro pc). Modello applicabile indipendentemente dalla distanza tra *client* e *server* (es. accesso a pagine Web).

In questo modello sono coinvolti due processi, uno sulla macchina *client* e uno sulla macchina *server*, che comunicano tra loro attraverso una serie di **richieste** e **risposte**, come mostrato nella figura sottostante.



Classificazione per tecnologia di trasmissione

Broadcast

Un solo canale di trasmissione condiviso tra tutte le macchine sulla rete.

Multicast

Trasmissione a un sottoinsieme di macchine.

Point-to-point

Collegamenti tra coppie di macchine.

Classificazione per taglia

Interprocessor distance	Processors located in same	Example
1 m	Square meter	Personal area network
10 m	Room	Local area network
100 m	Building	
1 km	Campus	
10 km	City	Metropolitan area network
100 km	Country	Wide area network
1000 km	Continent	
10,000 km	Planet	The Internet

PAN

Personal Area Network, permette la comunicazione tra dispositivi nel raggio di un metro quadrato circa; es: PC collegato alle sue periferiche (wired o wireless).

LAN

Local Area Network, rete privata che può coprire lo spazio di una stanza, un edificio o un insieme di edifici vicini; es. rete di campus, aziende ecc.

MAN

Metropolitan Area Network, rete che copre un'intera città; es. TV via cavo.

WAN

Wide Area Network, rete che copre una vasta area geografica, come un paese o un continente, e che può avere al suo interno diverse sottoreti. È attraversata da flussi di dati direzionali da host a host, il cui instradamento è gestito da ***router***.

Classificazione per topologia

In base alla **topologia** (la “forma” della rete), le reti si possono distinguere in moltissime tipologie, ma le principali sono:

- point-to-point;
- bus;
- star;
- ring;
- mesh;
- tree;
- hybrid.

Internet

Si dice che internet è una rete “quasi gerarchica” perché non riesce a effettuare un inglobamento perfetto; inoltre, in Internet si verifica una situazione simile ai processori per quanto riguarda la **virtualizzazione**.
All'interno di internet ci sono **milioni di computer** connessi (*hosts*) che eseguono **applicazioni di rete**. Questi hosts sono collegati tra loro attraverso ***links* di comunicazione** (fibra, rame, satelliti ecc.), ognuno dei quali ha un diverso transmission rate (***bandwidth***). Vi sono poi i ***routers***, che inoltrano pacchetti di dati secondo protocolli ben definiti (TCP, IP, HTTP ecc.).
Per questi motivi Internet si può definire una **“rete di reti abbastanza gerarchica”**, e si possono individuare delle differenze tra la rete Internet pubblica e le reti *intranet* private.
I due tipi di **standard** principali che regolano il funzionamento di Internet sono **RFC** (Request for Comments) e **IETF** (Internet Engineering Task Force).

Alla luce di ciò, si può dire che in generale una rete ha una **infrastruttura** che si evolve su più strati (***layers***) che comunicano tra loro attraverso **protocolli** e **servizi** (interfacce).

Protocolli

I protocolli definiscono il **formato**, **l’ordine** dei messaggi inviati e ricevuti e le **azioni** da intraprendere quando si riceve o trasmette nella rete.

Layers

Ogni layer assolve uno specifico compito.

Addressing

Meccanismo per l’**identificazione di mittenti e destinatari** di un messaggio.

Error Control

Controllo degli errori (es. ordine corretto dei messaggi).

Flow Control

Controllo di flusso, volto ad esempio ad impedire il sovraccarico di un servizio (mittente vuole inviare messaggi più velocemente di quanto il destinatario possa accettare).

Multiplexing

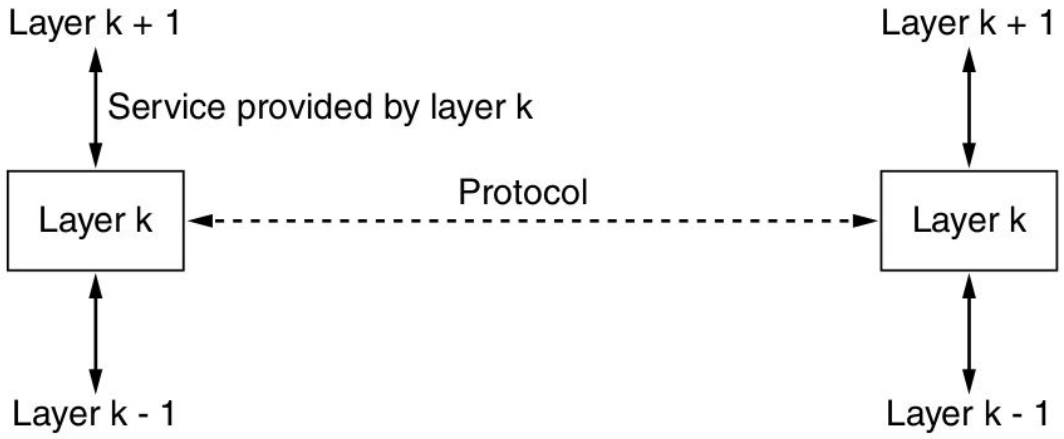
Utilizzo di **un solo canale** per trasmettere **diversi segnali**.

Routing

Scelta di un cammino in caso di molteplici cammini possibili per un messaggio.

Servizi

Funzionalità offerte da un layer (tramite interfacce). Tipicamente i **servizi** sono "**verticali**", mentre i **protocolli** "**orizzontali**":



I servizi possono essere di due diversi tipi: **Connection-Oriented** e **Connectionless**. I primi sfruttano un canale dedicato per comunicare, i secondi no.

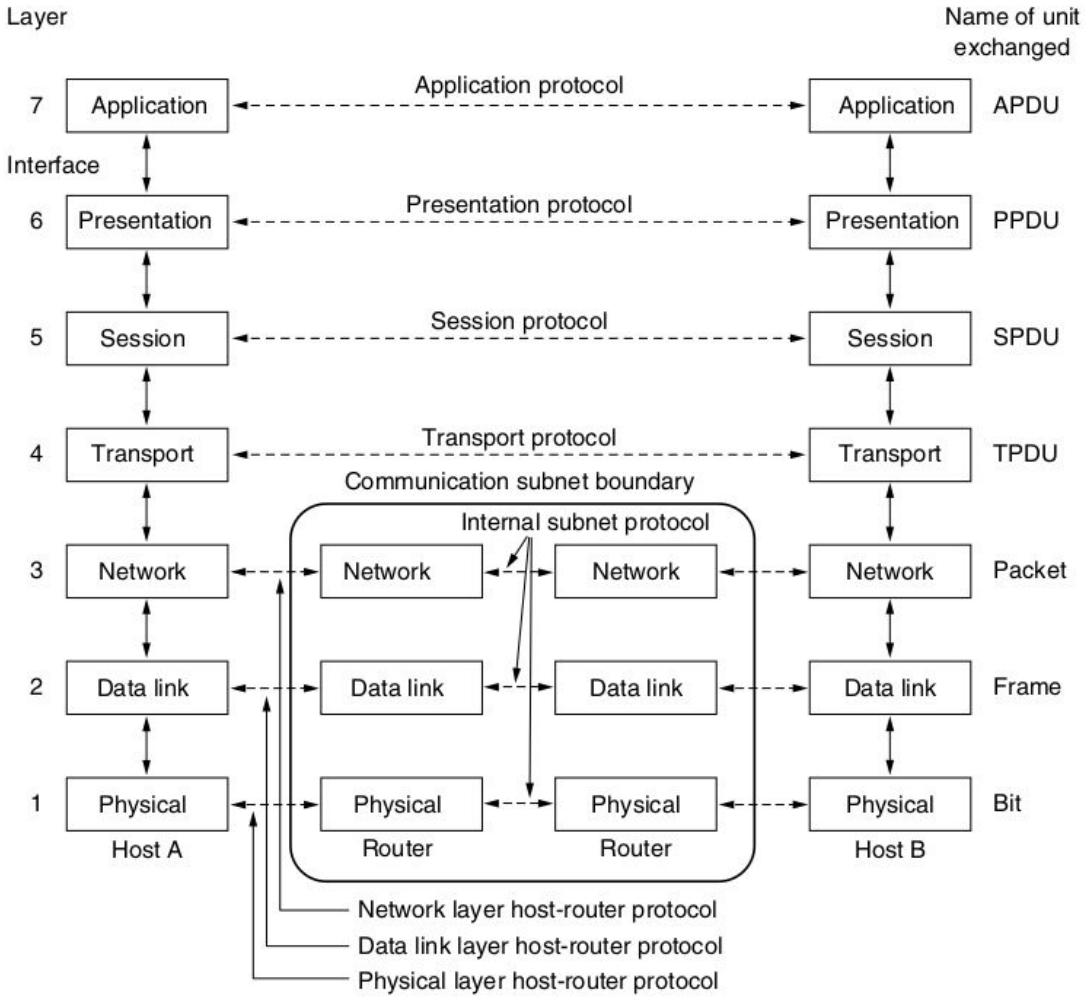
I modelli di riferimento

Per definire le proprietà di una rete e dare ordine a quanto detto finora si utilizzano due **modelli** ad alto livello: **OSI** e **TCP/IP**.

OSI Reference Model

OSI sta per **Open Systems Interconnection**, ed è il **modello ufficiale** prodotto dalla **ISO** (Organizzazione Internazionale degli Standard) e si può dividere in due strati principali: **upper layers** per le applicazioni e **lower layers** per le

comunicazioni. I principali **vantaggi** di questa architettura “a strati” sono costituiti dalla **separazione delle funzionalità** e dalla **maggiore interoperabilità** che essa garantisce.



Lower layers

Strato 1: Fisico

Manda/riceve i ***bits*** sul **media fisico**. Definisce ad esempio la quantità di Volt o gli intervalli di tempo secondo cui i bit sono inviati come segnali attraverso i canali. Rappresenta le fondamenta su cui è costruita la rete.

Strato 2: Data Link

Incapsula le unità di informazione e cerca di rilevare gli errori per non trasmetterli ai livelli superiori.

Strato 3: Network

Consegna dei pacchetti (**routing**).

Strato 4: Trasporto

Accetta dati dallo strato superiore e si assicura che **arrivino correttamente** allo strato network.

Upper layers

Strato 5: Sessione

Permette agli utenti su macchine diverse di stabilire una **sessione tra loro** (con servizi quali controllo del dialogo, gestione dei token e sincronizzazione).

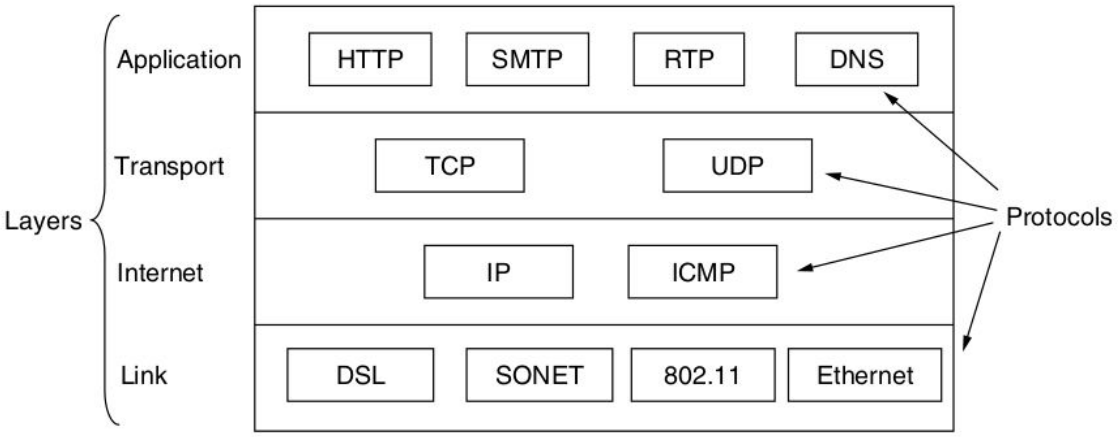
Strato 6: Presentazione

Si occupa della **sintassi** e della **semantica dell'informazione** trasmessa, assicurandosi che i dati siano rappresentati nel formato opportuno per macchine diverse.

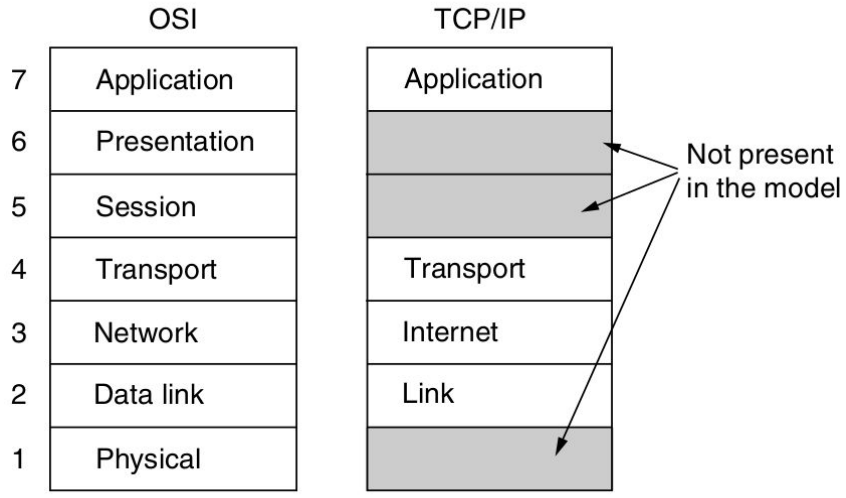
Strato 7: Applicazione

Comprende una varietà di **protocolli** comunemente richiesti dagli utenti.

TCP/IP Reference Model



Confronto con OSI

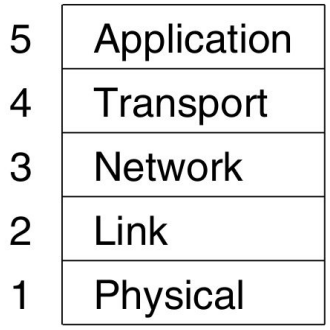


Problemi del TCP/IP

- Servizi, interfacce e protocolli non sono nettamente separati;
- Non esiste un modello generico, in quanto è stato creato per “**retrofitting**”;
- Lo strato “host-to-network” non è un vero strato;
- Non ci sono i livelli fisico e data link;
- Molti protocolli non sono separati, e quindi sono difficili da rimpiazzare/upgradare singolarmente.

Nonostante tutto ciò e nonostante OSI sia apparentemente perfetto, **TCP/IP** è lo **standard de facto**; nella pratica si usa un modello ibrido.

Hybrid Model



Strato fisico

Tipi di trasmissione

Trasmissioni wired

UTP

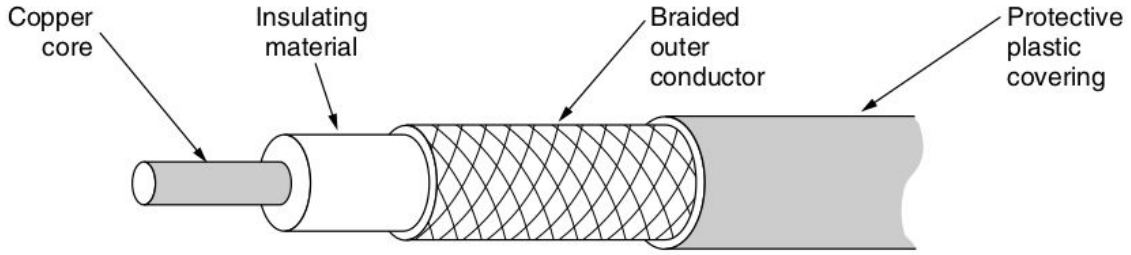
Acronimo di **Unshielded Twisted Pair**, tipo di cavo costituito da una coppia di fili spessi circa 1 mm avvolti uno intorno all'altro ("twisted") tra loro per **limitare il crosstalk** (interferenza reciproca).

UTP3  Bandwidth: ~250MHz; meno spire/cm.

UTP5  Bandwidth: ~600MHz; più spire/cm.

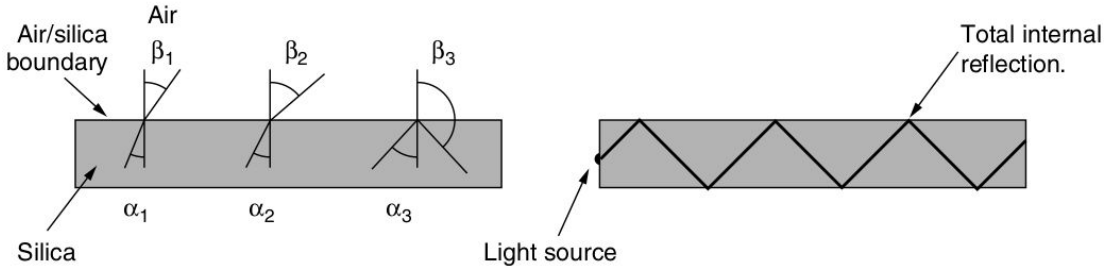
I cavi UTP si possono utilizzare per trasmettere segnali analogici e digitali, l'ampiezza di banda dipende dal diametro del cavo e dalla distanza percorsa.

Cavo coassiale

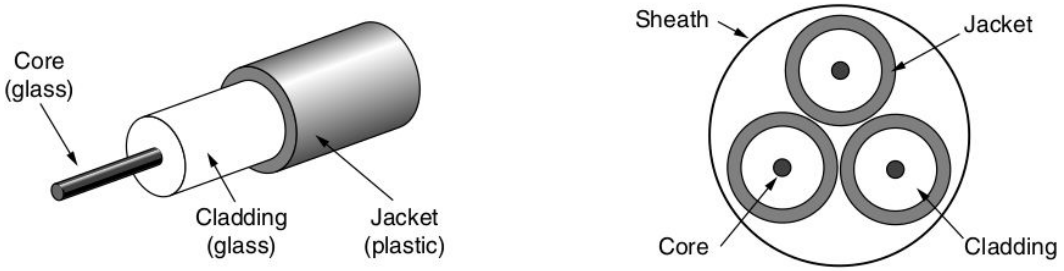


Cavo in rame, composto da un nucleo conduttore coperto da un rivestimento isolante a sua volta circondato da un conduttore cilindrico, che infine è avvolto da una guaina protettiva di plastica. È molto più schermato dei cavi UTP ed è molto usato per la TV via cavo e all'interno delle MAN.
La banda disponibile dipende dalla qualità, dalla lunghezza del cavo e dal rapporto segnale-rumore del segnale dati. In molti ambiti il cavo coassiale è stato sostituito dalla fibra ottica per i tratti più lunghi.
Bandwidth: ~1GHz.

Fibra ottica



Un sistema di trasmissione ottico è formato da tre componenti fondamentali: la **sorgente luminosa**, il **mezzo di trasmissione** e il **rilevatore**. Per convenzione, un impulso di luce indica il valore 1 e l'assenza di luce indica il valore 0. Il **mezzo di trasmissione** è una **fibra di vetro** sottilissima (8-50 micron), realizzata in silicio. Quando viene colpito dalla luce, il rilevatore genera un impulso elettrico. Collegando a un estremo della fibra una sorgente di luce e un rivelatore all'altro, si crea un **sistema di trasmissione unidirezionale** che accetta un segnale elettrico, lo converte e lo trasmette sotto forma di impulso luminoso; all'altra estremità della fibra converte nuovamente l'output in segnale elettrico. Generalmente le fibre sono raggruppate in fasci, protetti da un'ulteriore guaina più esterna.



I cavi utilizzati nei sistemi di trasmissione ottici possono essere a **una o a tre fibre**.

Fibra monomodale e multimodale

Esistono due tipi di fibra, la monomodale e la multimodale. La **monomodale** è più costosa e utilizzata soprattutto per le lunghe distanze, in cui la **luce** può propagarsi **solo in linea retta senza rimbalzare**. La **multimodale** invece può contenere **più raggi** che **rimbalzano ad angoli diversi**, in questo caso si dice che ogni raggio ha una modalità diversa, da qui il nome multimodale.

Connessioni tra fibre e tipi di luce

Per effettuare le connessioni in un sistema di trasmissione ottico si possono usare:

- **connettori** (perdita del 10-20% di luce);
- **allineatori** meccanici (perdita del 10% di luce);
- **fusione** (perdita dell'1% di luce).

Le fibre trasportano **diversi tipi di luce**, ognuno dei quali ha proprietà diverse. Ecco un confronto tra **LED** e **laser a semiconduttore**:

Item	LED	Semiconductor laser
Data rate	Low	High
Fiber type	Multimode	Multimode or single mode
Distance	Short	Long
Lifetime	Long life	Short life
Temperature sensitivity	Minor	Substantial
Cost	Low cost	Expensive

Fibra ottica VS cavi in rame

Vantaggi

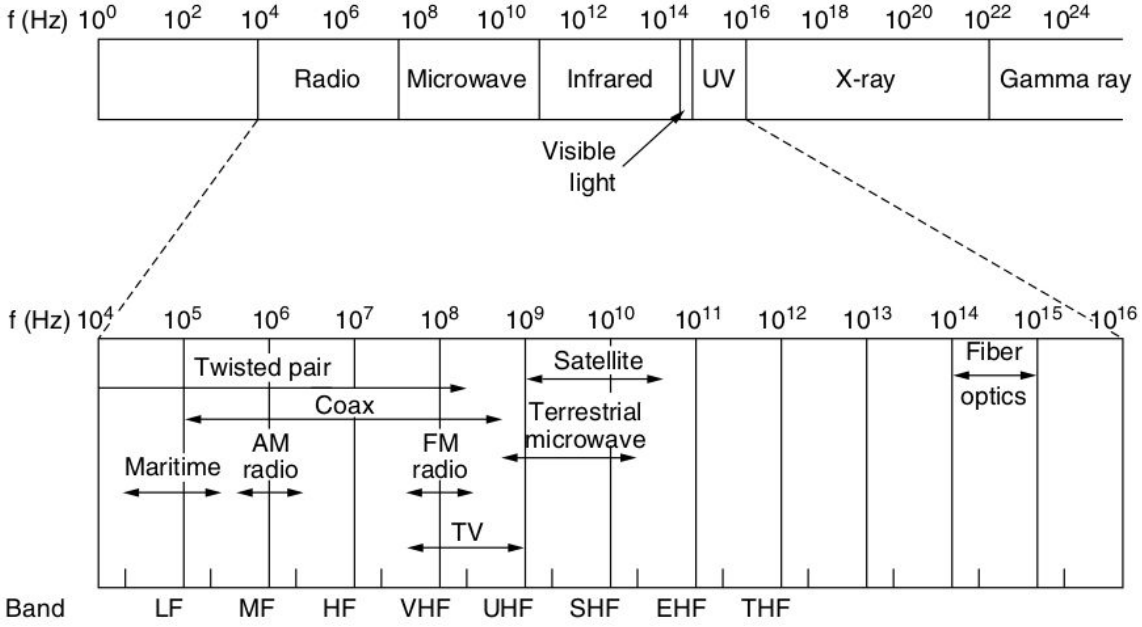
Mettendo a confronto la fibra ottica coi cavi in rame, emergono diversi vantaggi e svantaggi.
Il primo punto a favore della fibra è la **maggiore ampiezza di banda**, che la rende necessaria per le reti di fascia alta; inoltre, a causa del basso livello di attenuazione, i ripetitori possono essere installati ogni 50 Km di linea, mentre nel caso dei cavi in rame i ripetitori devono essere installati ogni 5 Km.
La fibra ha anche il vantaggio di non essere influenzata dalle sorgenti elettriche, dai campi elettromagnetici, dalle interruzioni della linea elettrica e dai corrosivi chimici presenti nell'aria, caratteristica che la rende **adatta anche agli ambienti più inospitali**. Inoltre, la fibra è **sottile e leggera**, quindi sostituendo tutti i cavi in rame con cavi in fibra, è possibile svuotare i condotti con costi ridotti di cablaggio e manutenzione. Infine, i dati che viaggiano su fibra sono difficili da intercettare.

Svantaggi

D'altro canto, la fibra è una tecnologia meno nota e si può **danneggiare** se la si **piega troppo**. Poiché la trasmissione ottica è intrinsecamente unidirezionale, la **comunicazione bidirezionale** richiede **due fibre o due bande di frequenza** su una singola fibra. Infine, le interfacce per la fibra ottica costano più di quelle elettriche e l'**unione di due tratte** in fibra ottica risulta **molto laboriosa**.

Trasmissioni wireless

Lo spettro elettromagnetico



Le **frequenze** si assegnano **in base a chi fa più del bene** o a chi offre di più, ma la banda **ISM** (Industrial, Scientific, Medical, usata da bluetooth, alcune reti 802.11, 2.4GHz ecc.) viene lasciata **libera**.

Trasmissioni radio

La **trasmissione radio** è **omnidirezionale**, non necessita cioè di particolari allineamenti tra trasmettitore e ricevente. In particolare, le onde radio a **bassa frequenza** hanno caratteristiche differenti da quelle ad **alta frequenza**; le prime (es. AM) passano gli ostacoli ma si disperdono altrettanto facilmente e seguono la curvatura terrestre, mentre le seconde (es. FM) non passano bene gli ostacoli, vengono assorbite dalla pioggia e non seguono la curvatura terrestre, rimbalzando sulla ionosfera.

Trasmissioni a micro-onde

Le **trasmissioni a micro-onde**, occupando frequenze superiori a 100MHz, presentano l'enorme vantaggio di poter **viaggiare quasi in linea retta** e possono quindi essere meglio focalizzate e viaggiare più a lungo; perché ciò sia possibile, però, **trasmettitore e ricevente** devono essere **ben allineati**. Per anni, le trasmissioni a micro-onde sono state usate anche per distribuire il segnale telefonico e quello televisivo.
Purtroppo però, le onde di questo tipo non passano bene gli edifici, e sono soggette a serie **interferenze atmosferiche**; inoltre, le trasmissioni a micro-onde sono anche poco costose.

Trasmissioni a infrarossi e millimetriche

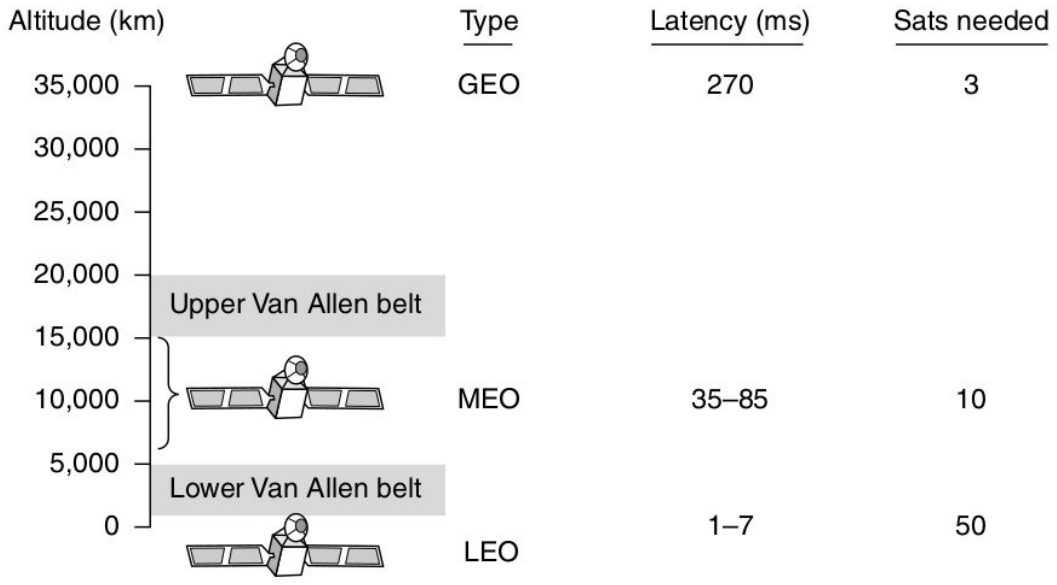
Usate ad esempio nei telecomandi, sono, a differenza delle micro-onde, **direzionali** e riescono ad attraversare gli oggetti solidi con ancora più difficoltà. I due vantaggi principali di questo tipo di trasmissioni sono costituiti dal loro costo (**economico**) e dalla **sicurezza** che garantiscono, data la distanza ridotta a cui devono trovarsi trasmettitore e ricevente e la direzionalità di cui sopra.

Trasmissioni luminose

Non presentano vantaggi particolari, in quanto **non funzionano** in presenza di **sole** o **interferenze atmosferiche**.

Satelliti di comunicazione

Un satellite di comunicazioni può essere immaginato come un **grande ripetitore** di microonde posto nel cielo. Questo dispositivo contiene diversi transponder, ossia ricetrasmettitori satellitari, i quali ascoltano una parte dello spettro, amplificano il segnale e lo ritrasmettono su altre frequenze per evitare interferenze.



Come si nota dalla figura, più i satelliti orbitano **vicini alla Terra** e più ne servono, ma allo stesso tempo in questo modo i **tempi di latenza** e la **potenza** richiesta **diminuiscono** notevolmente; inoltre, il lancio di satelliti in orbita bassa è meno costoso. Sempre nella figura si possono notare due "**fasce di Van Allen**", che indicano strati di particelle molto cariche intrappolate nel campo magnetico terrestre: attraversandole, qualsiasi satellite verrebbe velocemente distrutto.

Le bande principali su cui operano i satelliti di comunicazione sono le seguenti:

Band	Downlink	Uplink	Bandwidth	Problems
L	1.5 GHz	1.6 GHz	15 MHz	Low bandwidth; crowded
S	1.9 GHz	2.2 GHz	70 MHz	Low bandwidth; crowded
C	4.0 GHz	6.0 GHz	500 MHz	Terrestrial interference
Ku	11 GHz	14 GHz	500 MHz	Rain
Ka	20 GHz	30 GHz	3500 MHz	Rain, equipment cost

Satelliti MEO

Satelliti medio-orbitali (2000-35786 km, tra le due fasce di Van Allen), sono stati i primi nella storia dell'umanità e si trovano in altitudini comprese tra le due fasce di Van Allen. Si spostano lentamente lungo la longitudine terrestre, impiegando circa 6h per compiere un giro completo; devono perciò essere **tracciati** mentre si spostano nel cielo. Possono inoltre essere raggiunti con trasmettitori meno potenti di quelli usati per i satelliti GEO, e al momento non sono utilizzati per le comunicazioni, ma si usano ad esempio per il **GPS**.

GPS

Tecnologia del dipartimento della difesa USA (**satelliti NAVSTAR**) resa poi gratuita in tutto il mondo; dato che utilizza **satelliti MEO** e quindi non stazionari, quando ci si connette bisogna attendere il cosiddetto **tempo di fix**, cioè qualche minuto impiegato per triangolare correttamente la posizione dei satelliti vicini per ottenere quella del dispositivo che la richiede. Questo tempo di fix è stato poi ridotto grazie all'utilizzo di **barometro** e **A-GPS**, un sistema che stima prima una posizione approssimativa basandosi sulle celle telefoniche nelle vicinanze.

Satelliti GEO

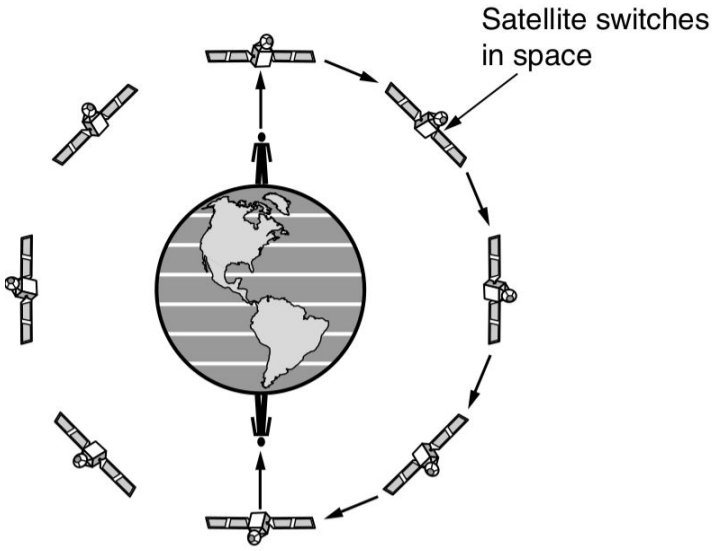
Satelliti geostazionari (altitudine 35786 km, molto sopra la fascia di Van Allen Superiore) che risultano più appetibili sotto molti aspetti e che stanno **sull'equatore in orbita circolare**. Per evitare interferenze possono essere **massimo 180**, e sono utilizzati principalmente come satelliti spia, per il meteo e per la TV satellitare (DSB, frequenze nella banda Ku).

Satelliti LEO

Satelliti **basso-orbitali** (300-1000 km, sotto la fascia di Van Allen inferiore) con **latenza di pochi millisecondi** e i cui lanci sono relativamente economici, sono usati ad esempio per servizi vocali quali **Iridium** e **Globalstar**.

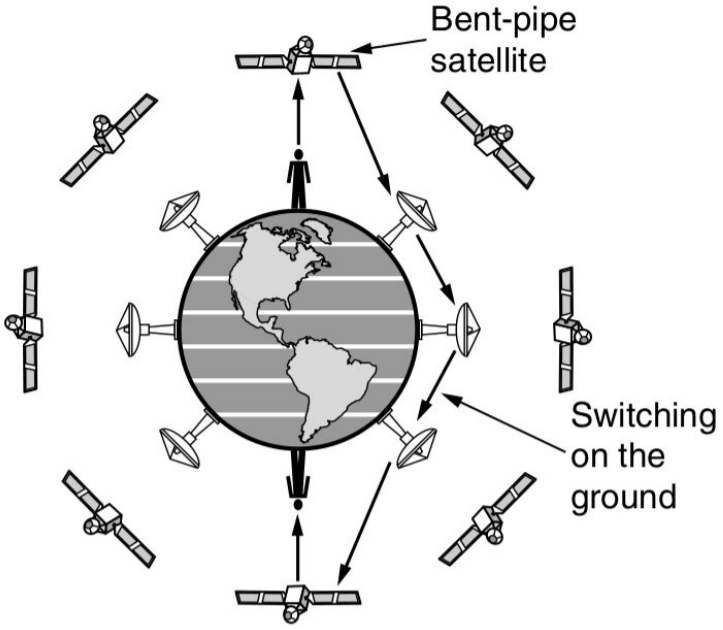
Iridium

Sistema composto da **66 satelliti**, unico nel suo genere perchè **copre l'intera superficie terrestre** (poli inclusi) fornendo servizi di voce, fax, data etc. ovunque nel mondo. I suoi satelliti formano 6 "collane" attorno alla Terra, per un totale di 1628 celle mobili che coprono il pianeta. Attualmente Iridium non è molto utilizzato in ambito vocale, ma fa parte del **Tsunami Warning System**. Inoltre, questo sistema satellitare garantisce un **data rate molto basso** (2200-3800 baud).



Globalstar

Sistema molto diverso da Iridium, in quanto composto da una serie di "**ripetitori bent-pipe**" con stazioni gateway nel mezzo; presenta **52 satelliti** (meno costosi delle controparti Iridium) ma non copre tutta la superficie terrestre. Un punto a favore di Globalstar è però costituito dal fatto che in molte aree, grazie ad un accordo con i gestori GSM, **non necessita di un telefono apposito**.



“Rottamare” i satelliti

I cosiddetti “**space debris**” sono satelliti lasciati alla deriva perché inutilizzati; attualmente in orbita LEO ci sono più di **46 milioni** di detriti di dimensioni tra 0.1 e 10 cm. La soluzione migliore è allontanare questi satelliti inutilizzati nella cosiddetta “**orbita cimitero**”. Per quanto riguarda i detriti già presenti, l’**Inter-Agency Space Debris Coordination Committee** si occupa di tracciarli (se le loro dimensioni superano i 10 cm)

Satelliti: situazione attuale

Attualmente le comunicazioni sono più sviluppate attraverso tecnologie terrestri (es. fibra) , ma nonostante ciò **il satellite non è affatto morto** in quanto va ancora bene per diversi scenari di utilizzo:

- **zone impervie** o poco popolate;
- **reti** strategiche/**militari**;
- utenti dedicati (es. Iridium);
- complemento alla rete terrestre (es. Globalstar);
- studio di fenomeni terrestri (mappe, meteo ecc.);
- **reti broadcasting passive** (es. GPS).

Le basi della comunicazione

Misure della capacità di trasmissione

Bandwidth

Larghezza di banda, costituisce la **misura fisica** della capacità di trasmissione di un canale e rappresenta l’insieme di frequenze trasmesse; si misura in **Hertz**.

Data Rate

Misura informativa, informalmente rappresenta **quanta informazione passa in un secondo**; formalmente di può rappresentare in come **bit o baud rate** (bit o simboli trasmessi al secondo).

L’informazione in bit esprimibile da un alfabeto di V simboli si ricava attraverso la formula $log_2(V)$; quindi, la relazione tra bit rate e baud rate è la seguente: $bitrate = baudrate * log_2(V)$. Ovviamente, più frequenze riusciamo a trasmettere, maggiore sarà il data rate, quindi **per poter trasmettere a frequenze più alte servirà una potenza maggiore**.

Bitrate

La quantità di informazioni digitali che è trasferita o registrata nell'unità di tempo. Stiamo parlando quindi di velocità di trasmissione, espressa in bit/s. La velocità di trasmissione è anche detta Banda e dipende dal tipo di mezzo trasmissivo utilizzato e dalle sue condizioni fisiche al momento dell’uso; possiamo anche considerare il bitrate come un baudrate con alfabeto ridotto a 0 e 1.

Baudrate

Rappresenta il numero di simboli che viene trasmesso in un secondo. Non va confusa con il sopracitato bitrate in quanto misurano unità differenti; infatti ad un simbolo corrisponde un numero di bit differente in base alle tecniche di modulazione utilizzate.
Il baudrate misura relativamente all'**alfabeto di trasmissione** (composto da un certo numero di simboli) che abbiamo a disposizione.

Serie di Fourier

Un **segnale che ha una durata finita** può essere gestito immaginando semplicemente che esso **ripeta infinite volte l'intero schema** (intervallo T e 2T è identico all'intervallo 0 a T).
È possibile quindi rappresentare i segnali tramite funzioni, le quali permettono un'analisi e una modellazione più efficace. La **Serie di Fourier** non è altro che la **scomposizione di un segnale in componenti sinusoidali** (possibilmente infiniti).

$$g(t) = \frac{1}{2}c + \sum_{n=1}^{\infty} a_n \, sen(2\pi nft) + \sum_{n=1}^{\infty} b_n \, cos(2\pi nft)$$

$f = 1/T$ rappresenta la frequenza fondamentale, a_n e b_n sono rispettivamente le ampiezze seno e coseno dell'n-esima armonica e c rappresenta una costante.
Su questo teorema si basano le reti e il passaggio dei dati tramite i mezzi di trasmissione; purtroppo nella pratica i **mezzi di trasmissione attenuano** in modo non uniforme i componenti della serie di Fourier, generando così una **distorsione**. Per ovviare a questa distorsione, le ampiezze fino ad una certa frequenza vengono trasmesse senza modifiche, da quella frequenza in poi vengono attenuate. L'intervallo di frequenze trasmesse senza una forte attenuazione è chiamato **Banda Passante**. Generalmente nella realtà viene indicata la banda passante compresa tra 0 e la frequenza dove la potenza è attenuata del 50%.

Attenuazione

Ogni impulso energetico trasmesso in un mezzo che non sia il vuoto subisce una **attenuazione in potenza**, che si indica col **rapporto tra potenza trasmessa e potenza ricevuta**. Questa attenuazione dipende però dalla frequenza, quindi una forma d'onda in generale subisce attenuazioni diverse a seconda delle sue componenti nella serie di Fourier; di conseguenza, possiamo dire che la **larghezza di banda** è limitata e **dipende** fortemente **dal mezzo di trasmissione**.

Teorema di Nyquist

A questo punto, viene da domandarsi se, **data la larghezza di banda** di un certo canale, sia possibile **calcolare il limite massimo della quantità di informazione** che è possibile trasmettere; la risposta a questa domanda è affermativa e ci viene spiegata dal **teorema di Nyquist**, secondo il quale il **data rate massimo** (in bit al secondo) è $2B \, log_2 L$, dove B è la banda massima e L i livelli del segnale che vengono usati. Questo teorema vale però in un **canale in cui non ci sono**

interferenze, cosa altamente **improbabile** nella pratica. Nel mondo reale, infatti, ci sono sempre interferenze e si calcola il **rapporto tra la potenza del segnale e la potenza del rumore del canale**; spesso questo rapporto si indica in scala logaritmica ($10\ log_{10}(S/N)$) usando i decibel, e si dice **SNR Ratio**. Tutto ciò si può **generalizzare tenendo conto del rumore**, ottenendo così il **teorema di Shannon**, secondo cui il massimo data rate è $B\ log_2(1 + S/N)$.
Purtroppo però, i data rate massimi calcolati con queste formule, specie se in condizioni ottimali, sono **massimi fisici**, quindi quasi **impossibili da raggiungere** nella pratica.

Dispersione

La dispersione costituisce un problema ancora più grave della attenuazione e consiste in un **cambio della forma d'onda**. Inoltre, la dispersione può essere diversa a seconda della frequenza e rappresenta un problema **molto grave nelle lunghe distanze**. La soluzione più immediata sarebbe quella di limitare la lunghezza massima di ogni tratto e aggiungere dei ripetitori che aggiustino e ritrasmettano il segnale corretto; si è però scoperta l'esistenza di **onde bidirezionali** molto difficili da fermare e **stabili** anche in caso di collisioni con altre onde. Queste onde si chiamano **solitoni**, e soffrono di una dispersione minima.

Trasmissione di segnale digitale

La trasmissione di segnale digitale può avvenire in **3 modi**: **modulando in ampiezza**, **frequenza** (frequency shift keying) o **fase** (phase shift keying).
Durante l'invio di informazioni, il segnale può subire **attenuazione**, **distorsione** o venir disturbata dal rumore; questo porta ad **evitare l'uso** di un largo i**ntervallo di frequenze**, di cui normalmente le onde quadre utilizzate nei segnali digitali avrebbero bisogno.
Per aggirare questi problemi viene usata la **trasmissione AC**, che **permette la modulazione** della sua ampiezza (AM), frequenza (FM) o fase.

Modulazione di frequenza

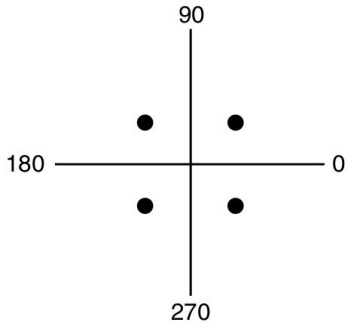
La modulazione in frequenza non è altro che una tecnica di trasmissione utilizzata per trasmettere informazioni usando la **variazione di frequenza dell'onda portante**. Rispetto alla modulazione in ampiezza ha il vantaggio di essere **molto meno sensibile ai disturbi** e permette una trasmissione di **miglior qualità**. Ha inoltre un'efficienza energetica molto maggiore dato che la potenza del segnale modulato FM è esclusivamente quello della portante.

Modulazione di fase

Possiamo usare vari sfasamenti in ogni singolo impulso, in modo da avere un alfabeto di simboli più capiente e quindi, a parità di baud, aumentare il bitrate.

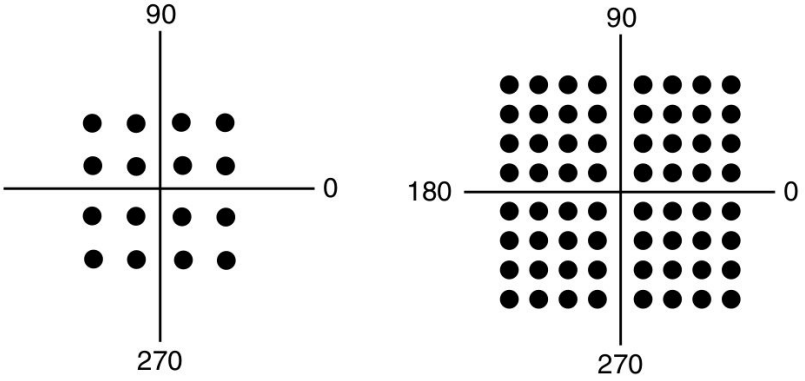
QPSK

Quando si usano **4 sfasamenti** (45°, 135°, 225°, 315°) si ha un alfabeto di 4 simboli, quindi il **bitrate è doppio rispetto al baud** (in quanto ricordiamo che il bitrate dispone solo di due simboli). Questa tecnica si chiama **Quadrature Phase Shift Keying**.



QAM

Verrebbe a questo punto spontaneo aumentare gli sfasamenti per aumentare il bitrate, ma se continuassimo a cambiare solo la fase otterremmo differenze poco distinguibili, quindi è meglio **combinare più tipi di modulazione insieme**. Un buon approccio sarebbe quello di tenere la frequenza al massimo e combinare modulazioni di ampiezza e fase. A seconda del numero di simboli possiamo ottenere diverse tipi di QAM, come ad esempio il **QAM-16** (16 simboli, bitrate quadruplo del baudrate e doppio del QPSK) o il **QAM-64** (64 simboli, bitrate sestuplo del baudrate e triplo del QPSK).



Quelle appena viste non rappresentano però le **combinazioni migliori** di modulazioni, che sono invece costituite dai **circular QAM**; nonostante ciò, si usano comunque i **QAM rettangolari** perchè risultano **più facili da generare e decodificare**.

Le grandi reti

Con “grandi reti” si fa riferimento a reti di scala mondiale, primi fra tutti (in ordine cronologico) **telegrafo** (wired) e **fax** (considerabile come un overlay del telegrafo).

Il sistema telefonico

Nato come overlay del telegrafo e **inizialmente** solo **point-to-point**, solo **successivamente** ci si affida a degli **switching center** (centralini in cui operatori collegavano fisicamente i cavi tra gli utenti), che poi aumentano gerarchicamente (centralini dal primo al quinto livello). La rete così ottenuta è chiamata **PSTN**, Public Switched Telephone Network. Attualmente, a livelli più alti **l'infrastruttura** è costituita da **cavi coassiali o in fibra**, mentre il cosiddetto "**ultimo miglio**" è composto da cavi **UTP3**.
Attraverso l'infrastruttura telefonica, i dati viaggiano in digitale tranne per quanto riguarda l'ultimo miglio, in cui tornano ad essere analogici. Per effettuare il **passaggio da digitale ad analogico** è necessario un dispositivo chiamato **modem**.

La linea telefonica

Calcolando, attraverso la formula di Shannon, il **limite fisico della linea telefonica** tra due utenti che usano il modem si ottiene un valore di circa **35 kbit/s**, che raddoppia nel caso in cui uno dei due utenti utilizzi un servizio digitale; si potrebbe poi spingere fino a 64 kbit/s, ma si è scelto di **adeguarsi allo standard americano di 56 kbit/s**.

Lo switching

Analizziamo ora il modo in cui gli switching center dirigono il traffico all'interno della gerarchia telefonica. Ciò può avvenire con 3 diverse tecniche: il circuit switching, il message switching e il packet switching.

Circuit switching

Col circuit switching, si crea un **collegamento fisico tra chi comunica**; ciò ovviamente richiede del tempo, portando così ad avere un **delay iniziale**, ma ha come vantaggio il fatto di garantire un **collegamento dedicato**.

Message switching

In questo caso invece, non si aspetta di creare un collegamento, ma si **lancia direttamente il messaggio**, e il cammino viene creato man mano che lo attraversiamo. Questa tecnica è anche detta **store-and-forward**. Il problema qui è che se un **messaggio** è **particolarmente grande** occupa risorse di uno switch, rischiando di **ritardare altri messaggi**.

Packet switching

Per risolvere il problema del message switching, col packet switching si **divide il messaggio in tanti packets** di lunghezza massima prefissata; così facendo, non solo non occupiamo risorse, ma permettiamo anche **trasmissioni** di parti del messaggio **in parallelo**.

Il Fax

Nato anch'esso come **overlay del telegrafo**, risulta per questo motivo **più lento del telegrafo stesso**. Inoltre, venne commercializzato per trasferire immagini, quindi non se ne capisce il potenziale per quanto riguarda il testo.
Si dividono in base alla velocità in vari gruppi e possono offrire **diverse risoluzioni** (es. standard, fine e superfine). Nel tempo, il fax ha usato **diversi standard di trasmissione**, fermandosi al **V34.bis** data la sua natura point-to-point e il limite fisico di 35 kbit/s.

DSL

Le **Digital Subscriber Line** nascono per **superare il limite dei 56 kbit/s** imposto dai vecchi modem, ed entrano subito in competizione con la TV via cavo (che utilizzava il cavo coassiale con portata di 10 Mbps) e col satellite (portata 50 Mbps).
Il vecchio limite a 56 Kbps era dato dal local loop, cablato ancora con cavi UTP3 troppo costosi da sostituire completamente; si opta quindi per la **rimozione del limite di 4000 Hz sulla banda telefonica** (che era filtrata perchè le frequenze più altre erano inutili e creavano interferenze), arrivando a una banda possibile di 1.1 MHz. Per non compromettere il funzionamento degli apparecchi casalinghi (progettati per una banda massima di 4000 Hz) si inserisce quindi un filtro (**splitter**, passivo) "a valle" che separi il segnale telefonico da quello per i dati dividendo le frequenze superiori a 4000 Hz da quelle inferiori.

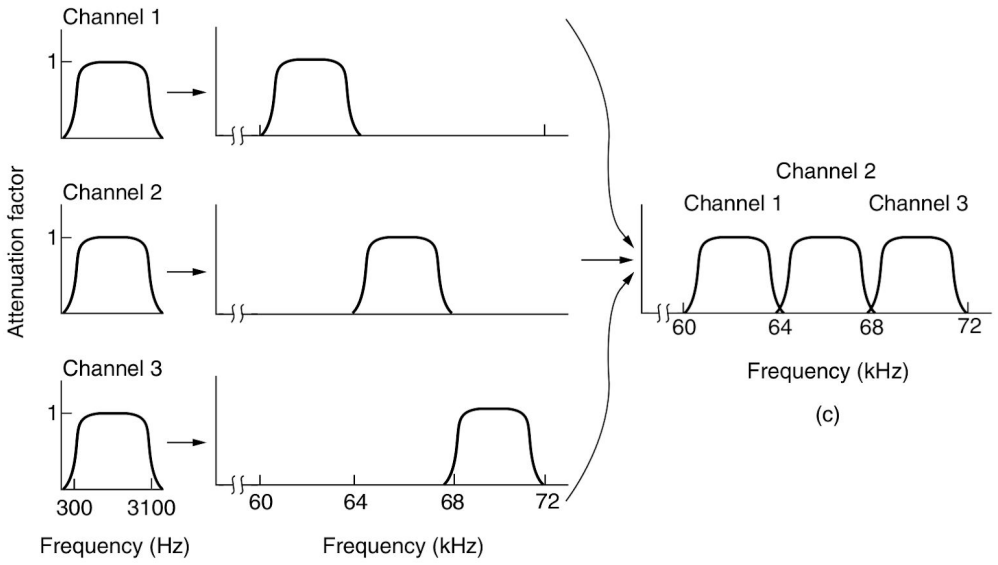
Multiplexing

Le aziende telefoniche hanno sviluppato elaborati schemi per convogliare molte conversazioni lungo un singolo collegamento fisico. Questi schemi di **multiplexing** possono essere divisi in due categorie di base: **FDM** (Frequency Division Multiplexing) e **TDM** (Time Division Multiplexing).

FDM

In **FDM**, lo **spettro di frequenza** è **diviso in bande** di frequenza e **ogni utente ha il possesso esclusivo** di parte della banda.
I filtri limitano l'ampiezza di banda utilizzabile a circa 3.100 Hz per canale; quando si uniscono in multiplexing diversi canali, a ciascun canale sono allocati 4.000 Hz in modo da **mantenere gli elementi ben separati**. Dopo aver aumentato la frequenza di ogni canale vocale di una quantità diversa, si esegue l'unione facendo attenzione a non sovrapporre alcun canale. Anche se i canali sono separati da intervalli (bande di guardia), **c'è sempre una leggera sovrapposizione** tra canali adiacenti perché i filtri non hanno bordi netti. A causa di questa sovrapposizione, un forte picco di segnale sul bordo di una canale viene avvertito nel canale adiacente sotto forma di **rumore non termico**.
La **quantità di banda** dedicata a ogni canale è definita **gruppo**. **Più gruppi** possono essere uniti in multiplexing per creare un **supergruppo**. L'unità successiva si chiama **mastergroup** ed è composta da **più supergruppi**. Con gli standard comunemente

utilizzati, un **mastergroup** può arrivare a tenere **fino a 600 conversazioni** in contemporanea su un solo cavo.



WDM

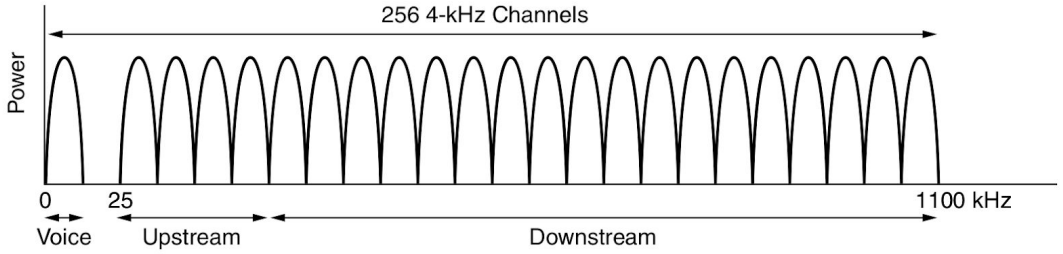
Per i canali in fibra ottica si utilizza una **variazione di FDM** chiamata WDM (**Wavelength Division Multiplexing**). Si tratta semplicemente di un FDM utilizzato a frequenze molto alte. Si può adoperare il multiplexing sulla fibra a lunga tratta se ogni canale ha il proprio intervallo di frequenza (in questo caso, lunghezza d'onda) e tutti gli intervalli sono distinti. L'unica differenza con la tecnica FDM elettrica è la presenza di un **sistema ottico completamente passivo**, e perciò altamente **affidabile**, basato su un reticolo di diffrazione. La tecnologia WDM si è evoluta più velocemente dei computer, arrivando a **32000 Gbps**.

TDM

Oltre al multiplexing con divisione di frequenza, esiste anche il TDM, cioè il **multiplexing a divisione di tempo**, chiamato anche **PCM** (Pulse Code Modulation) nella sua incarnazione telefonica. Utilizzando il TDM, gli utenti possono usare **tutta la banda a disposizione a turno per un piccolo intervallo di tempo**.

ADSL

Asymmetric DSL, è il tipo di DSL più comune ed è asimmetrica in quanto dà **più spazio al downstream rispetto all'upstream**.
Per gli standard ADSL, **l'FDM** si usa in un modo detto **Discrete MultiTone**; si **spezza cioè la banda in sottocanali** indipendenti di uguale ampiezza, ognuno trattato come una connessione telefonica a sé stante, rallentata/accelerata indipendentemente.



L'ADSL è stata poi sviluppata con **diversi standard**, che vanno dalla **ADSL Lite**, con velocità di 1,5 Mbit/s in downstream e 0.5 Mbit/s in upstream, alla **ADSL2+**, con 24 Mbit/s in downstream e 3,5 Mbit/s in upstream. Gli **standard più recenti** supportano anche varianti "all-digital", e le ADSL2+ in particolare utilizzano una **banda doppia** (2,2 MHz invece di 1.1).

VDSL

Per superare queste velocità, si **riduce il più possibile la presenza di cavi UTP3** per passare ad una infrastruttura in fibra. Si ottiene così la **Very high speed DSL**, che usa **frequenze più alte** della ADSL. In questo caso le varie tipologie di VDSL a disposizione variano in base a quanto cavo in UTP3 è presente, passando dalla VDSL iniziale, con 55 Mbit/s in download e 3 in upload, alla VDSL2+, con i suoi 300 Mbit/s in download e 100 in upload. Successivamente, per **gestire meglio la divisione della banda** tra download e upload sono stati creati vari **band plans**.

La televisione

Le **due tecnologie principali** utilizzate in ambito di trasmissione televisiva sono il **satellite** e il **digitale terrestre**. La **TV satellitare**, a causa delle caratteristiche del satellite che abbiamo già visto, utilizza modulazione **QPSK**, mentre il **digitale terrestre** utilizza **QAM**, in 3 varianti base: QAM-4, QAM-16, QAM-64.
Il digitale terrestre non è però esente da problemi, anzi; soffre in particolare di **problematiche** legate alla conformazione del **territorio** e di problemi di distorsione. Per ovviare a tutto ciò, si usa il **FDM** nello stesso modo in cui si usa nell'ADSL, cioè **suddividendo i canali in tante trasmissioni indipendenti**, dividendo solitamente tra trasmissioni normali e HD; per creare spazio extra, inoltre, si usa la **compressione MPEG2**.

La telefonia mobile

Per distinguere le varie evoluzioni della telefonia mobile, si parla di **generazioni**. Occorre ricordare che tutta la telefonia mobile si basa su un **problema fondamentale**: la divisione del territorio, cioè come **gestire l'infrastruttura fissa** che permette le connessioni in mobilità. Alla base di questa infrastruttura ci sono le **celle telefoniche**, cioè gli switching center che coprono una certa zona di territorio.

0G

Analogica, deriva dalle trasmissioni radio, che si sono poi evolute nei sistemi **Push To Talk**. Funziona grazie a **un solo canale** che permette di ricevere e trasmettere (**half duplex**, o si trasmette o si riceve; per questo PTT).

IMTS

Con l'IMTS si passa a **2 frequenze**, quindi non serve più il PTT; di conseguenza, **aumenta** anche **la privacy**, in quanto non si sentono più le comunicazioni altrui. In più, si passa a trasmettitori molto potenti per creare **celle di centinaia di km**, scontrandosi con la **quantità di canali** decisamente **ridotta**. In **zone remote** però, dove 23 canali sono sufficienti a causa dello scarso traffico telefonico, l'IMTS è ancora in uso, visti i **pochi ripetitori richiesti**.

1G

Vent'anni dopo, viene introdotto l'Advanced Mobile Phone System (**AMPS**, noto in Italia come **TACS**, Total Access Communication System. In questo sistema (**analogico**), le **celle** coprono solo **10-20 km**, il numero di utenti serviti aumenta e **diminuisce la potenza richiesta** per la trasmissione, portando alla diffusione di **apparecchi telefonici più leggeri e meno costosi**. Riducendo la copertura delle celle, però, **aumentano le interferenze**; la soluzione si trova **separando le frequenze** e dividendole secondo una **matrice esagonale** per ovviare a problemi dati dalla struttura delle città. Inoltre, il numero di frequenze diverse dipende da zona, densità e necessità di espansione.
Il funzionamento della struttura a celle è quindi il seguente: ogni cella ha al centro lo switching center (la stazione base) e ogni **cellulare** è **sempre connesso** a una sola **cella** finché non si sposta; momento in cui avverrà **l'handoff**, cioè il **passaggio di segnale** ad un'altra cella.
Dato che ogni cella gestisce tanti utenti occorre fare **multiplex** e si è scelto ancora una volta **l'FDM**, per gestire 832 canali full-duplex (solo 45 effettivi per cella). Ogni 15 minuti circa ogni cellulare manda il suo **numero seriale** (32 bits) e il suo **numero telefonico** (34 bits) per registrarsi alla cella più vicina. Nel momento in cui si chiama si usa un canale condiviso apposito per attivare la richiesta; allo stesso modo, in ricezione c'è un apposito canale di paging (anch'esso condiviso) che i cellulari controllano per sapere se ci sono chiamate che li riguardano e, se ne trovano una, la **cella** fornisce un **canale esclusivo per la comunicazione**.

Handoff

Quando il segnale è troppo debole, lo switching office assegna il cellulare alla cella vicina che riceve con potenza più alta. In base al modo in cui avviene questo passaggio, si può fare una distinzione tra **hard handoff** e **soft handoff**: nel **primo caso**, la vecchia stazione lascia il cellulare prima che la nuova lo agganci e si verificano perciò **lag o cadute di linea**; viceversa, nel **secondo caso** la nuova cella acquisisce il cellulare prima che la vecchia lo lasci, ma il cellulare dovrebbe essere in

grado di collegarsi a due frequenze in contemporanea, **aumentando i costi e la potenza richiesta**.
C'è poi il caso del **MAHO** (Mobile Handoff): ogni cellulare è associato a una cella di cui monitora la potenza del segnale; se il segnale è debole, la cella può disconnetterlo e lui cerca una cella migliore. In questo modo il **carico sul cellulare è minimo**, in quanto si sfruttano i **tempi morti del TDM** per misurare la potenza del segnale.

2G

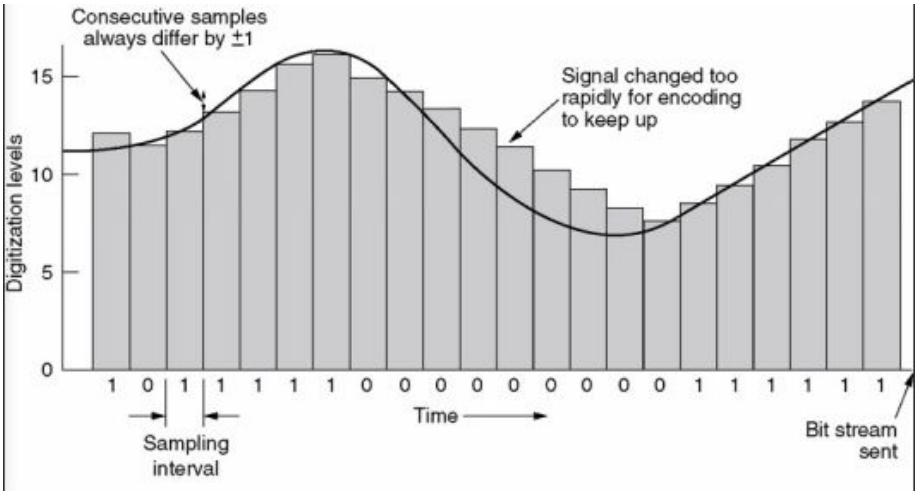
Prima generazione digitale, diversi standard ne fanno parte.

D-AMPS e PDC

D-AMPS è lo **standard USA**, **PDC** usa la stessa tecnologia ma riporta delle differenze per il mercato **giapponese**.
D-AMPS e AMPS possono coesistere nelle stesse celle, in quanto D-AMPS riusa tutte le bande di AMPS sugli 850 MHz ma in digitale e ne ha di aggiuntive per aumentare la capacità del servizio, che viaggiano intorno ai 1900 MHz; per questo motivo, usando lo standard D-AMPS bastano **antenne più piccole**.
In più, grazie alla compressione digitale del flusso voce, si riescono a mandare sulla linea **6 volte gli utenti del vecchio AMPS**. Una volta compresse, le comunicazioni vengono gestite con **TDM**, ma ciò peggiora la qualità del suono rispetto alle generazioni precedenti. Per quanto riguarda l'handoff, in D-AMPS si usa il MAHO.

Delta modulation

La delta modulation è un **metodo di digitalizzazione e compressione di un segnale analogico**. Il **segnale** analogico viene **suddiviso in unità di tempo**, e successivamente viene **digitalizzato** tramite la differenza di valore tra il valore precedente e quello successivo, attraverso una scala di +-16 che viene espressa tramite 5 bit. La delta modulation si basa su questo principio, ma viene ridotto alla variazione rappresentata da 1 bit: se il segnale successivo è più alto di quello precedente allora viene rappresentato da 1, altrimenti 0. Si basa sul fatto che il segnale cambia in modo relativamente lento rispetto alla frequenza di campionamento, ciò rende gran parte dell'informazione ridondante. Se il **cambiamento del segnale** analogico è **troppo ampio**, si ha una **perdita di dati**, ma in ambito telefonico (più precisamente nelle chiamate), questo non rappresenta un problema.



GSM

Global System for Mobile Communication, sistema in uso in **Italia e in Europa**. È molto simile a D-AMPS, infatti si usa **FDM** con **TDM**, ma con canali più ampi (tengono più utenti e hanno un data-rate più alto) e **frequenze diverse**. In particolare, ogni canale gestisce 13 kbps, e la qualità della voce e della trasmissione dati ne risente positivamente.
La struttura del GSM prevede **quattro tipi di celle**:

- **macro**, le più grandi, **sopraelevate** rispetto agli edifici;
- **micro**, più piccole delle macro, **alte** quanto gli edifici;
- **pico**, molto piccole e usate per aree molto **dense** (anche indoor);
- **umbrella**, piccole **estensioni** per coprire i buchi tra le celle principali.

Un'altra caratteristica del GSM è l'uso della **SIM** (Subscriber Identity Module), piccola scheda disponibile in varie taglie che contiene due informazioni molto importanti: IMSI e Ki. **IMSI** sta per International Mobile Subscriber Identity, ed è **l'identificativo** della SIM, mentre la **Ki** è la **chiave di autenticazione**, utile perchè **GSM supporta l'autenticazione crittografica a chiave condivisa**. Il collegamento utilizzando GSM avviene infatti in questo modo: il cellulare manda l'IMSI della SIM all'operatore, che genera un numero casuale e lo manda al cellulare; a questo punto, il cellulare firma il numero con la Ki e lo ri-manda all'operatore, che lo firma a sua volta con la stessa Ki e controlla che sia uguale a quello appena ricevuto.

CDMA

Il sistema **Code Division Multiple Access** è **completamente diverso** da AMPS, D-AMPS e GSM. Invece di dividere l'intervallo di frequenze assegnate in poche centinaia di canali a banda stretta, CDMA permette a **ogni stazione** di **trasmettere per tutto il tempo** attraverso l'**intero spettro di frequenza**. Trasmissioni multiple simultanee sono separate usando la teoria della codifica. CDMA rende meno rigida la premessa secondo la quale i frame entrati in collisione si alterano completamente; CDMA, infatti, presume che segnali sovrapposti si sommino linearmente. La chiave di CDMA è pertanto la capacità di estrarre il segnale desiderato scartando tutto il resto.

In particolare, possiamo dire che con CDMA si lavora su uno **spazio multidimensionale**, in cui si stabiliscono degli **assi adeguati** e poi si usano le **regole di composizione e proiezione** per fare encoding/decoding. Per creare assi che usino solo +1 e -1 si usano le **matrici di Hadamard**, di grandezza attualmente limitata a **668** (=utenti massimi supportati da una cella).
Nel CDMA, la base trasmette a potenza fissa, nota al cellulare che, in base alla potenza ricevuta, può calcolare quanto è lontano dalla base e trasmettere alla potenza opportuna; in questo modo, la **banda** a disposizione è **gestita senza sprechi**.
Inoltre, il CDMA **sfrutta al meglio l'intermittenza**, cioè non spreca tempo per i silenzi.

2,5G

GPRS

Lo standard GPRS (**General Pocket Radio Service**) è comunemente classificato come **2,5G**, in quanto si inserisce a cavallo tra 2G e 3G; è essenzialmente un overlay del 2G che permette la gestione del traffico a pacchetti in multiplex, **senza sprechi di banda** (che costituivano il problema principale del GSM), senza bisogno di un canale dedicato e con **tariffe a traffico**. GPRS supporta i **protocolli IP e PPP** per Internet e **alloca dinamicamente i canali** dati e quelli voce, a seconda delle richieste di traffico. Esistono vari tipi di cellulari GPRS, che cambiano in base ai modi in cui combinano GSM e GPRS.

2,75G

EDGE

Sta per **Enhanced Data rates for GSM Evolution**, ed è **retrocompatibile** con GPRS e GSM. Utilizza **modulazioni di fase** oltre a quelle di frequenza e si classifica in varie versioni corrispondenti a diverse velocità; la **velocità** attuale **varia** però anche in **base alla qualità** del servizio.

3G

Supporta **più utenti e più data rate**, usando **bande di frequenza più larghe** rispetto al 2G. Gli standard 3G principali sono W-CDMA e CDMA2000.

W-CDMA

Sta per **Wideband** (usa una banda per canale molto larga, **5 MHz**) **CDMA**, ed è conosciuto in **Europa** come **UMTS** (Universal Mobile Telecommunications System). Il suo **data rate tipico** è **384 kbps**.

CDMA2000

Standard **statunitense**, la cui **velocità massima** tipica è di **144 kbit/s**; la sua **banda** per canale è larga **1,25 MHz**.

3,5G

HSDPA

Sta per **High Speed Downlink Packet Access** ed è **l'evoluzione** (retrocompatibile) di **UMTS**, con velocità che vanno da 1,8 a 42 Mbps.

3,75G

HSUPA

High Speed Uplink Packet Access, **evoluzione** di **HSDPA** con **velocità** massima di **11,5 Mbps** (quindi **meno del 3,5G** nella sua evoluzione finale).

4G

HSOPA

High Speed OFDM Packet Access, anche detto **LTE**, ha **bande variabili da 1,25 MHz a 20 MHz** e può raggiungere **velocità** di **1,2 Gbps** in download e 600 Mbps in upload. Ne esistono **varie versioni**, che arrivano fino alla CAT19 con velocità notevoli, dovute alla **combinazione di varie tecniche** e al maggior uso di banda.

5G

Il 5G userà **frequenze prima occupate dal digitale terrestre**, quindi si introdurrà il **nuovo standard TV DVB-T2** per liberarle; in particolare, si parla di frequenze tra i **28 e i 39 GHz**.

Trasmissione stereo wireless

Esempio di multiplexing wireless, in quanto nella stereofonia ci sono due canali da trasmettere.

FM Stereo

Il problema principale delle trasmissioni stereo FM è stato che le **trasmissioni stereo** sono venute **dopo** e dovevano perciò essere **compatibili** con le vecchie **trasmissioni monofoniche**.
Dato che la banda mono viaggiava su frequenze comprese tra 30 e 15000 Hz, è stato introdotto un **segnale pilota su frequenza 19000 Hz**, che ha il compito di **segnalare la presenza di segnale stereo**. A questo punto, vengono creati **due nuovi segnali**: il primo costituito dalla **media dei segnali stereo destro e sinistro**, e il secondo corrispondente alla **metà della differenza tra segnale sinistro e segnale destro**; i **ricevitori mono** riceveranno così solo il **primo** dei nuovi segnali, mentre quelli **stereo** decodificheranno anche il **secondo** per r**icostruire il segnale stereo**. Tutto ciò va però a peggiorare il rapporto segnale-rumore.

RDS

Esiste anche un altro **multiplexing**, chiamato **Radio Data System**, in cui i dati sono digitali e trasmessi su **frequenze di 57 KHz**; in questo caso il **data rate** è molto **basso**.

DAB

Esiste poi una **evoluzione della radio digitale**, il cosiddetto **Digital Audio Broadcasting**, ma non ha ancora preso piede per motivi di **costi, distorsioni e qualità**.

Lo strato Data Link

Strato che si occupa del **collegamento dati**, in particolare della loro **codifica/decodifica**. In più, questo strato fornisce un'interfaccia allo strato superiore (rete), e si occupa di **error control** e **flow control** (regola cioè il flusso di dati a seconda delle capacità della rete e del ricevente). Per fare tutto ciò si usa l'**interfaccia di rete**, che fornisce i servizi necessari al livello di rete per la trasmissione dei dati.

Tipologie di servizi

Unacknowledged connectionless

In questo servizio, i **pacchetti** vengono **inviati senza aspettare conferma** di avvenuta ricezione, e **senza** stabilire una **connessione dedicata**; per questi motivi, questo tipo di servizio è utile per voce/streaming media, o nel caso in cui il **canale** sia **molto affidabile**.

Acknowledged connectionless

Servizio analogo al precedente appena visto, con la differenza che in questo caso i **pacchetti** vengono "**confermati**" con ricevuta di ritorno.

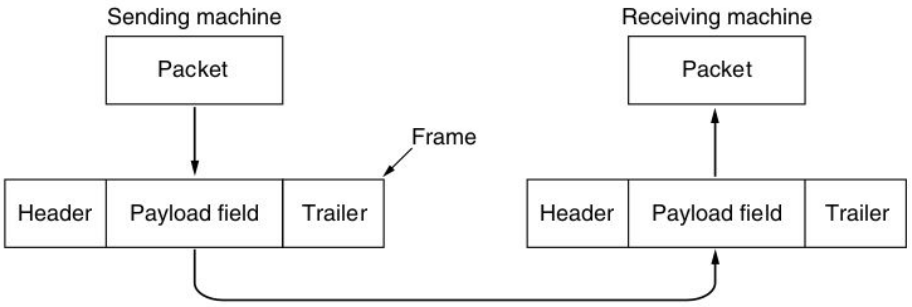
Acknowledged connection-oriented

In questo tipo di servizio, i **pacchetti** vengono "**confermati**" e in più viene stabilita una **connessione dedicata**.

La codifica

Framing

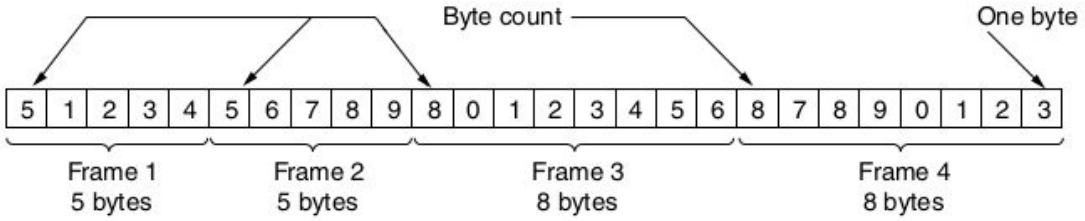
Per quanto riguarda la codifica, si prendono dei **pacchetti dati** dallo strato superiore e si codificano in appositi **frames**.



Il **principale** problema della trasmissione di frame sta **nell'accorgersi** dove un **frame inizia** e **finisce**, e uno dei metodi a cui si è pensato per risolverlo è il cosiddetto character count.

Character count

Questo metodo consiste nell'**inserire** all'interno dell'header l'informazione del **numero di caratteri** che costituiscono il payload, ma ciò ovviamente presenterebbe **problemi in caso di errori**, molto frequenti nell'ambiente delle reti.



Flag bytes

Byte stuffing

Questo secondo metodo consiste invece nell'utilizzo di un byte speciale (**flag byte**) con cui **segnalare l'inizio** e la **fine** di **ogni frame**; per poter usare byte uguali al flag byte all'interno del messaggio stesso serve però un metodo di **escaping**, che in questo caso si chiama **byte stuffing**. Anche in questo caso si presenta però un problema: usare **grandezze fisse** (in questo caso bytes) **prima o poi non va più bene**.

Bit stuffing

Per risolvere il problema causato dal byte stuffing si usa questo **metodo analogo**, ma applicato **a livello di bit**. In questo caso, **ogni frame inizia e finisce con una particolare sequenza di bit** (01111110). Ogni qualvolta che il livello data link del **mittente** incontra **5 1 consecutivi** nel payload, **inserisce uno 0**. Al contrario, quando il **ricevente** vede 5 1 consecutivi seguiti da uno 0 (escaping), **elimina lo 0**. Nel caso in cui tra i dati ci sia il flag byte 01111110, viene trasmesso come 011111010, ma ricevuto come 01111110.

Error control

Le tecniche viste finora ci permettono di sapere come identificare un frame sulla rete; andiamo ora a definire come comportarsi in caso di errori.

Error detection

Fanno parte di questa strategia tutte le **tecniche** che ci dicono **se un frame ha subito errori** durante la trasmissione. Per questo motivo, error detection è utile nel caso in cui il **canale** è **molto affidabile**, o quando un **errore non** è **critico**.
In particolare, per fare error detection avremo bisogno di **informazione extra rispetto ai dati** (come nel caso del framing) che ci permetta di trovare gli errori; proprio come nel framing, avremo quindi una fase di **encoding** in cui aggiungiamo la **protezione dagli errori**, e una di **decoding** in cui ci **liberiamo di questa protezione**; l'**error detection** avverrà nella fase di **decoding**, in cui il ricevente controllerà la presenza di errori.

Parity bit

Questa tecnica di error detection consiste nell'inserire **un bit di parità ogni m bits**; bit di parità che sarà uno **0 o un 1 in base alla somma degli m bits precedenti**. Il limite di questa tecnica è però che funziona solo con **errori di potenza 1**, riesce cioè a identificare solo gli errori che toccano un solo bit, e inoltre comporta una **diminuzione notevole del data rate**.

Built-in error detection

Nella pratica, codici di questo tipo si usano anche fuori dall'ambito delle reti, e in molti casi **l'informazione extra** per l'error detection **fa parte dei dati stessi**, rendendo il **riconoscimento degli errori indipendente dal metodo di trasmissione** (es. carte di credito, IMEI, ISBN, codici a barre).

CRC

Cyclic Redundancy Check è una **codifica polinomiale**, e in quanto tale è basata sul fatto di trattare le **sequenze di bit come dei polinomi a coefficienti** che possono assumere **solo** i valori **0 oppure 1**. Un frame di k bit è visto come una lista di coefficienti per un polinomio con k termini che variano da x^{k-1} a x^0. Tale polinomio è detto di grado k-1. Il coefficiente di termine più alto (quello più a sinistra) è il coefficiente per x^{k-1}; il successivo è per x^{k-2} e così via. Per esempio 110001 ha 6 bit e quindi rappresenta un polinomio di 5° grado con coefficienti 1, 1, 0, 0, 0 e 1: $x^5+x^4+x^0$. Nell'aritmetica polinomiale in base 2, **addizione e sottrazione corrispondono allo XOR**. Le divisioni lunghe vengono eseguite come in binario, salvo che le sottrazioni sono in modulo 2 come spiegato sopra. Si dice che un divisore "sta" nel dividendo se il dividendo ha tanti bit quanti il divisore.
Quando si utilizza una codifica polinomiale, la sorgente e la destinazione devono mettersi d'accordo in anticipo su un **polinomio generatore G(x)**. Il generatore deve avere i bit di ordine più alto e più basso uguali a 1. Per poter calcolare il checksum di

un frame di m bit che corrisponde al polinomio M(x), il frame deve essere più lungo del polinomio generatore.
L'idea è quella di aggiungere un **checksum alla fine del frame** in modo che il polinomio rappresentato dal frame con checksum sia divisibile per G(x). Quando la destinazione riceve il frame con checksum prova a dividerlo per G(x). Se c'è un resto vuol dire che c'è stato un errore di trasmissione.
L'algoritmo per calcolare il checksum è il seguente:

1. posto r il grado di G(x), aggiungere r bit con valore zero dopo la parte di ordine più basso del frame, così che adesso contenga m + r bit e corrisponda al polinomio x^r M(x)
2. dividere la sequenza di bit corrispondente a G(x) per la sequenza corrispondente a x^r M(x) usando la divisione modulo 2.
3. sottrarre il resto (che contiene sempre al massimo r bit) dalla sequenza corrispondente a x^r M(x) usando la sottrazione in modulo 2. Il risultato è il frame con checksum pronto per la trasmissione. Chiamiamolo polinomio T(x).

Dovrebbe essere chiaro che T(x) è divisibile (modulo 2) per G(x). In ogni divisione, se si sottrae il resto dal dividendo, quello che resta è divisibile per il divisore.
Questa tecnica è alla base dei codici di error-correction più avanzati, come ad esempio Reed-Solomon.

Error correction

La tecnica dell'error correction prevede la **correzione degli errori "in diretta"**, evitando così ritrasmissioni inutili, onerose per il canale nel caso in cui questo sia particolarmente inaffidabile.
Prima di procedere con la correzione dell'errore, occorre stimare la gravità del danno che potrebbe essere causato; per fare ciò, si considerano meno gravi gli errori che danneggeranno un solo bit nel messaggio, e con l'**aumentare dei bit danneggiati aumenta anche la gravità dell'errore**. In particolare, si chiama **distanza di Hamming la differenza di bit tra due messaggi di uguale lunghezza** (es. due messaggi sono distanti 1 se sono diversi per un bit ecc.), e le tecniche di error control si considerano più o meno potenti in base alla massima distanza di Hamming che riescono a sopportare.

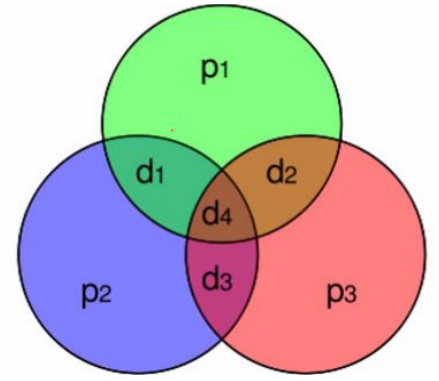
Repetition codes

Per rilevare (e successivamente correggere) errori di potenza più grande di 1, una buona tecnica è quella del **repetition code R_N**, che consiste nel **ripetere ogni bit trasmesso N volte** (es. con R_3 010 diventa 000111000). In questo modo, tra due messaggi diversi c'è una distanza di almeno N, quindi con R_N possiamo fare **error detection fino a potenza N-1**.
Per quanto riguarda l'**error correction**, nel caso in cui un codice generi messaggi legali distanti minimo N, ci basta che la potenza dell'errore sia meno della metà di N; in questo modo, la **potenza massima** dell'errore che saremo in grado di **correggere**

corrisponderà a **(N/2)-1 se N sarà pari, o a (N-1)/2 se N sarà dispari**. Anche in questo caso, però, il problema che si pone è rappresentato da una **notevole diminuzione del data rate** (1/N con R_n).

Codici di Hamming

I codici di Hamming fanno parte dei cosiddetti **codici lineari**, perché le loro operazioni si possono esprimere tramite combinazioni lineari, e utilizzano i **bit di parità in modo più efficiente**; possiamo infatti considerarli come una generalizzazione dei codici parity bit e repetition. Un codice di Hamming che codifica Y bits di dati usandone X con distanza Z si può scrivere come (X, Y, Z).



L'idea di base è che **i bit extra si dividono equamente la parità dei bit dati**, e **l'encoding** si può scrivere **linearmente** usando una **matrice generatrice**. I codici di Hamming migliori/più conosciuti sono (7, 4), (8, 4), (11, 7).

Teorema del peso minimo

Si dice che il **peso di un messaggio** binario è il **numero di 1** presenti al suo interno. Secondo il teorema del **peso minimo**, se il peso minimo dei vettori della matrice generatrice X per Y è **d**, allora **ogni coppia di messaggi codificati dista almeno d**; il **codice di Hamming necessario** sarà quindi **(X, Y, d)**, e sarà un codice e**rror detecting** di potenza **(d-1)** e **error correcting** di potenza **(d-1)/2**.

Errori burst

Generalmente gli errori hanno una struttura che ci è utile a identificarli e a correggerli. Nella pratica, gli errori occorrono spesso in **burst** (**più di uno alla volta**, in particolare almeno il primo e l'ultimo bit sono sbagliati) e ciò crea numerosi problemi ai codici error-correcting.

Interleaving

Una tecnica di error correction in grado di correggere anche gli errori burst è chiamata interleaving (**tecnica della matrice invertita**) e consiste nel **calcolare i parity bit dei dati in ordine diverso** rispetto a quello di trasmissione dei dati stessi. In particolare, calcoleremo **un bit di parità per ognuna delle n** colonne e invieremo **tutti i bit di dati come k righe** e i bit di ogni riga da sinistra a destra come al solito; nell'**ultima riga** inseriremo i **n parity bits**.

L'interleaving può essere vista come una tecnica che converte codici in grado di identificare o correggere singoli errori in codici in grado di identificare o correggere errori burst.

Erasures

Un altro tipo di errori è costituito dalle erasures, cioè vere e proprie **cancellazioni di dati**.

Codice Reed-Solomon

Il codice Reed-Solomon fa sempre parte dei **codici lineari**, e l'idea intuitiva che sta alla sua base consiste nell'**usare i resti di una divisione di polinomi come bit di parità extra**. Un codice RS(X,Y), come nel caso di Hamming, codifica Y parole usandone X; può quindi **correggere (X-Y)/2 errori**. Una delle differenze principali rispetto ai codici di Hamming, sta nel fatto che il codice Reed-Solomon può considerare gruppi di bytes invece che singoli bits come unità di base.
Questo codice è **molto utilizzato** sia nel caso di **errori burst** (per esempio nei CD) sia nel caso di **erasures**, in quanto risulta molto efficace in entrambe le situazioni: può addirittura **correggere errori e cancellazioni contemporaneamente**.

Error control al di fuori delle reti

Moltissimi **errori** si verificano per esempio nei nostri computer, per esempio nei **dischi rigidi**; per controllarli esiste una tecnica chiamata **RAID**, disponibile in diverse tipologie a seconda di come i codici parity o di Hamming vengono implementati.
Anche nelle **RAM** si verificano molti errori, ma si è scelto di preoccuparsi solo quelli di potenza 1 nelle RAM generiche; esistono poi le RAM **ECC**, che usano un codice di **Hamming** (72, 64), penalizzando un po' il data rate, e le RAM **SECDED**, che correggono **errori di potenza 1 e 2**.
L'error control avviene anche all'interno di schede video o di memoria.

Teorema di Shannon

Prima di questo teorema, sembrava che, col diminuire del tasso di errore grazie ai codici visti finora, diminuisse anche il data rate. Shannon dimostra invece che, **dato un certo tasso di errore x**, ci sono codici che arrivano a un **data rate massimo pari all'entropia del canale** (cioè $H_2(x) \equiv x\, log\frac{1}{x} + (1-x) log\frac{1}{(1-x)}$).

LDPC

Uno dei codici error-correcting che più si avvicina a quanto detto da Shannon è il codice LDPC (**Low Density Parity Check**), usato ad esempio nella TV digitale, nel Ethernet e nelle ultime versioni di 802.11.
I codici LDPC sono **codici lineari**, in cui **ogni bit di output** è formato da una **frazione dei bit di input**; ciò porta a una matrice che rappresenta il codice e che ha una bassa densità di 1 (da cui deriva il nome del codice). Le parole ricevute sono

quindi decodificate con un algoritmo di approssimazione che migliora iterativamente ***on a best fit of the received data to a legal codeword***, correggendo gli errori.

Flow control

I protocolli stop-and-wait

L'idea alla base di questi protocolli è la seguente: **si manda un frame**, per poi **aspettare** un messaggio di conferma (**ack**) dal ricevente, che ci segnala che possiamo inviare un altro frame. Il principale vantaggio di questo tipo di protocolli è che basta un **canale half duplex** per implementarli, in quanto non c'è mai comunicazione contemporanea. Gli svantaggi stanno invece nella **lentezza** e nel fatto che in caso di errori il mittente potrebbe ritrovarsi ad **aspettare all'infinito** una conferma dal ricevente che non arriverà mai.

PAR e ARQ

Per risolvere il secondo problema dei protocolli stop-and-wait si aggiunge un **timeout**, allo scadere del quale il mittente procederà con un **secondo invio del messaggio**. Inoltre, per **evitare ricezioni multiple** dello stesso messaggio, si aggiunge un **bit per distinguere due frame contigui** e capire se uno di questi è stato già ricevuto. I protocolli che dispongono di queste informazioni aggiuntive rispetto ai protocolli stop-and-wait si chiamano **Positive Acknowledgement with Retransmission** o anche **Automatic Repeat reQuest**.
In questi casi, però uno dei problemi maggiori sta nell'**overhead** che si crea a causa dei messaggi di conferma, che sono tanti quanti i pacchetti inviati.

Piggybacking

Per risolvere il problema appena citato, ci viene in aiuto la tecnica del piggybacking: invece di inviare un messaggio di conferma ogni volta che riceviamo un frame, aspettiamo di dover **inviare un frame dati**, e **alleghiamo a questo il messaggio di conferma**. In questo modo i pochi bit necessari per la conferma aggiungeranno un **overhead minimo** al messaggio dati che avremmo inviato comunque, invece di occupare un intero frame a parte.
Ovviamente, questa tecnica funziona meglio se la **comunicazione** è abbastanza **equilibrata** tra le due parti; in caso contrario, si rischia addirittura di **sprecare banda in più.**

Problema: l'utilizzo della linea

Con i protocolli stop-and-wait si rischia di **attendere più di quanto si impiega per trasmettere**, e ciò significa sottoutilizzare il canale; in particolare questo problema è evidente quando il **prodotto bandwidth*round-trip-delay** è **molto grande**. Il reale utilizzo della linea che stiamo facendo, nel caso di protocolli con ack si può calcolare nel seguente modo: $\frac{S}{(S+C*R)}$, dove S è la taglia del frame, C è la capacità del canale e R è il tempo di round-trip; se **S<CR** avremmo un'**efficienza minore del 50%**.

Una possibile soluzione per evitare di aspettare troppo consiste nell'aggiungere un **timer supplementare** (la cui durata dev'essere più breve rispetto al timer principale) da parte del ricevitore; una volta scaduto, mandiamo di nuovo l'ack, anche senza piggybacking.

Il NAK

Per quanto riguarda i timer del sender, sappiamo che se il tempo è poco variabile ci basta settare il timer su un tempo poco più lungo di quello medio di attesa. Se il **tempo** dovesse essere molto **variabile**, invece, si setta il **timer del sender** a un valore abbastanza **largo**, e in caso di **rischio di errore** molto **alto** (calcolato dal receiver) il **receiver invia un NAK** (Not AcK), cioè un messaggio che avvisa il sender che il **pacchetto corrispondente non è stato ricevuto** e gli dà il permesso di ritrasmettere immediatamente, senza aspettare il suo timer.
Il NAK si può attivare tramite o un apposito timer, per ogni slot della sliding window, che tenga conto di ricevere il pacchetto con il numero corrispondente, o in caso di ricezione di un pacchetto "fuori flusso".

Sliding Windows

La tecnica delle sliding windows sfrutta l'idea del **pipelining**: non ci preoccupiamo cioè quando non siamo certi se un solo frame arriverà a destinazione, ma quando ciò potrebbe succedere a n frame, dove n è tipicamente una potenza di due, per non sprecare bits. La **taglia della sliding window** può **variare** sia per il sender che per il receiver, dando luogo a vari protocolli; in particolare, **più la finestra è ampia e più pacchetti saranno inviati senza bisogno di conferma**.

Go Back N

Tipologia di **protocolli sliding windows** in cui la t**aglia della finestra di chi riceve** è uguale a **1** (cioè noi siamo "rilassati" e il nostro interlocutore è "apprensivo"); ciò funziona bene quando **non ci sono molti errori** ma il **prodotto** bandwidth*round-trip-delay è **alto**. Quindi, in caso di mancata conferma, dovremmo essere pronti a rimandare il messaggio: ciò significa che dovremmo avere un **buffer di taglia n frames** assieme a n timer per l'eventuale ritrasmissione. Nel caso in cui il buffer si esaurisca, non invieremo più frame.

Selective repeat

In questa variante dei protocolli sliding windows, **entrambe le parti si "rilassano"** e ampliano la propria finestra: anche il receiver dovrà quindi allocare un buffer che abbia come taglia il numero di pacchetti inviabili senza conferma (l'apertura massima della sua finestra).

Wikipedia: Il Selective Repeat è un metodo simile al Go-Back-N, con la differenza che si ha una finestra di ricezione oltre a quella di trasmissione.

RiassuntoBibbia: Con il selective repeat, quando viene ricevuto un frame in errore viene scartato, mentre i frame buoni ricevuti successivamente vengono salvati in un buffer; quando la sorgente va in timeout, solo il frame più vecchio senza ACK viene ritrasmesso. Se quel frame arriva correttamente, la destinazione può passare in sequenza allo strato network tutti i frame presenti nel buffer. La ripetizione selettiva può inviare dei NACK (Not acknowledgement) quando trova un errore, così da stimolare la ritrasmissione prima dello scadere del timer. La ripetizione selettiva corrisponde ad avere una finestra di ricezione maggiore di 1.

Problemi

Contrariamente a quanto possa sembrare, se la taglia della finestra è troppo grande si rischia di fare confusione, in particolar modo nel caso in cui non sia garantita la sequenzialità del canale. Conviene perciò limitare l'**apertura al massimo alla metà della grandezza delle sliding view**.

Protocollo HDLC

Bits	8	8	8	≥0	16	8
	01111110	Address	Control	Data	Checksum	01111110

- **Data**: il **payload** del frame, con i dati veri e propri
- **Checksum**: calcolato usando **CRC**
- **Address**: componente di **indirizzamento**, utile nel caso in cui ci siano terminali multipli da distinguere dentro la stessa rete

Control

Il controllo di flusso avviene tramite una sliding window di grandezza massima 3 bit.
Ci possono essere **3 tipi di frame**: information, supervisory e unnumbered.

Il frame Information

Bits	1	3	1	3
(a)	0	Seq	P/F	Next

- **Seq**: contiene il n**umero del controllo di flusso** della sliding window
- **P/F**: sta per **Poll/Final**; quando il bit indica **P**, si chiede al ricevente di **iniziare la trasmissione**, se invece indica **F** si segnala che la **trasmissione** va **conclusa**

Il frame Supervisory

Frame che si occupa della supervisione del flusso dati.

(b)

1	0	Type	P/F	Next

- **Type**: indica quale dei **quattro possibili** tipi di controllo di supervisione è in uso
    - Type **0**: **ACK** (RECEIVE READY nello standard); segnala se il **flusso** è **sbilanciato**
    - Type **1**: **REJECT**; è un **NAK generalizzato** che segnala che tutti i frame a partire da quello indicato in poi nella sliding window vanno ritrasmessi. In questo caso Next indica il primo frame
    - Type **2**: **RECEIVE NOT READY**; segnala che ci sono **problemi di congestione** nel receiver e quindi la trasmissione va bloccata finché il receiver non manda un ACK
    - Type **3**: **SELECTIVE REJECT**; **NAK classico**.

Il frame Unnumbered

Usato per **ulteriori comandi di controllo**.

(c)

1	1	Type	P/F	Modifier

Alcuni dei comandi utilizzati in questo frame sono:

- **DISC**: sta per disconnect e segnala che la macchina sta uscendo dalla rete in via definitiva
- **SNRM** (Set Normal Response Mode): segnala che una nuova macchina è entrata nella rete e indica un canale asimmetrico, in cui il nuovo entrato è meno importante
- **SABM** (Set Asynchronous Balanced Mode): crea una connessione bilanciata, in cui chi entra ha gli stessi diritti degli altri
- **FRMR** (Frame Reject): indica che è arrivato un frame con una sequenza di controllo non corretta/sconosciuta
- UA (Unnumbered Ack): comando dedicato che si occupa degli ACK dei comandi di unnumbered.

Protocollo PPP

Regola le connessioni internet dedicate **punto a punto** (infatti sta per **Point-to-Point Protocol**). Si occupa di **LCP** (**Link Control Protocol**), cioè di **attivare le connessioni**, **testarle**, stabilire le **opzioni di negoziazione** e **chiuderle**. PPP fornisce inoltre un modo per negoziare le opzioni del livello di rete in modo che sia indipendente dall'uso del livello di rete; il metodo scelto consiste nell'avere un **NCP** (**Network Control Protocol**) per ogni livello di rete supportato.

Bytes	1	1	1	1 o 2	Variabili	2 o 4	1

Flag 01111110	Address 11111111	Control 00000011	Protocol	Payload	Checksum	Flag 01111110

Il frame PPP è stato disegnato cercando di renderlo il più **simile** possibile a **HDLC** (come si può notare dal delimitatore, che è lo stesso); PPP usa però **byte stuffing** invece che bit stuffing e non usa le tecniche di numbering e ACK per creare una comunicazione affidabile. PPP è inoltre un **protocollo a lunghezza variabile**. Analizziamo i vari campi:

- **Address**: stesso significato che aveva in HDLC, ma qui non si usa (per questo ha **valore fisso 11111111**);
- **Control**: potrebbe fare controllo di flusso ma **non si usa** neanche questo campo, quindi tutti i frame sono di tipo non numerato e il campo ha **valore fisso 00000011**;
- **Protocol**: specifica il **protocollo** che PPP sta implementando (principalmente 2 tipi);
- **Payload**: qui ci sono i **dati**, il cui significato dipenderà dal protocollo specificato in Protocol, ed è possibile aggiungere alcune **opzioni di configurazione** (es. settare il controllo di error-detection a 2 o 4 bytes, settare la lunghezza dei campi protocol o payload, rimuovere i campi Address e Control ecc.);
- **Checksum**: calcolato con **CRC** (PPP fa error detection ma non error recovery).

LCP

Link Control Protocol, sono **protocolli di negoziazione** che restano nel data link layer, **cominciano con bit 1** e si occupano di **stabilire la linea**.
LCP usa **11 tipi di frame**, di cui:

- 4 di configurazione;
- 2 di terminazione;
- 2 di rifiuto;
- 2 di echo;
- 1 di test.

Configurazione LCP

- **configure-request**: inviandola al receiver, il **sender** propone **opzioni per la configurazione della linea**;
- **configure-ack**: risposta del **receiver**, con la quale comunica l'**accettazione delle configurazioni** proposte (compensando in questo modo la mancanza di un controllo ACK nel contenitore PPP);
- **configure-nak**: eventuale **risposta negativa** dal **receiver** (che rifiuta alcune opzioni della configure-request, di cui chiede un **nuovo invio**) al sender;
- **configure-reject**: il **receiver** comunica al sender che non c'è nulla da fare e **rifiuta** la configure-request **senza** richiedere **ulteriori invii**.

Terminazione LCP

- **terminate-request**: il **sender** chiede al receiver di **chiudere la connessione**;
- **terminate-ack**: il **receiver conferma la chiusura** della connessione.

Rifiuto LCP

- **code-reject**: il **receiver** comunica al sender che non ha compreso la richiesta (**richiesta sconosciuta**);
- **protocol-reject**: il **receiver** comunica al sender che **non conosce/supporta il protocollo** o che c'è stato un errore sulla linea.

Echo LCP

- **echo-request**: il **sender** chiede al receiver di **rimandargli** indietro il **frame inviato**;
- **echo-reply**: il **receiver** invia al sender il **frame richiesto**.

Questa operazione è utile per **controllare/misurare la qualità della linea** di comunicazione.

Test LCP

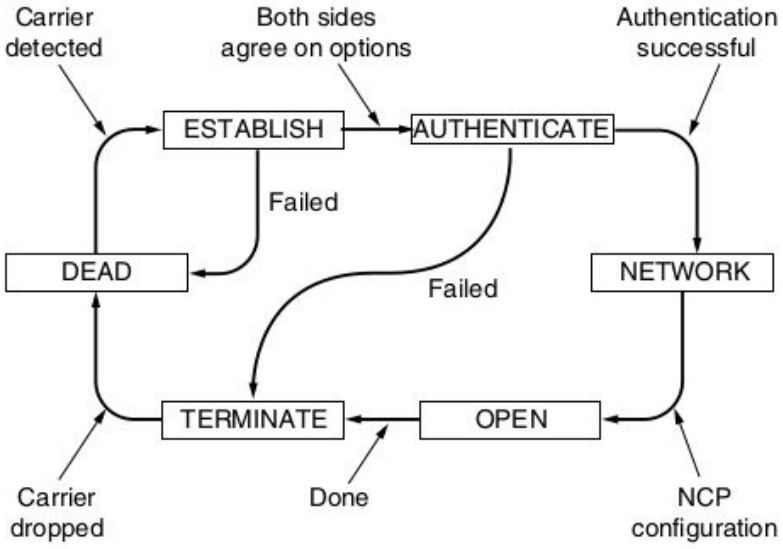
- **discard-request**: il **sender** chiede al receiver di **ignorare il frame inviato**.

Questa operazione serve a fare un **test preliminare** della linea o a **trovare loops** al suo interno.

NCP

Network Control Protocol, protocolli che **interagiscono con quelli dello strato network** e che cominciano con bit 0.

Ciclo di connessione PPP



MTU

Maximum Transmission Unit, è una misura che specifica la **dimensione massima del payload**; modificando l'MTU del PPP si può aumentare la qualità della linea. Settare una **MTU più grande** comporterà per esempio un a**umento della dimensione dei pacchetti**, una riduzione dell'overhead e quindi un aumento anche della banda a disposizione (operazione utile nel caso in cui il canale abbia pochi errori); viceversa, settando una **MTU più piccola** avremmo una **diminuzione della dimensione dei pacchetti**, un aumento dell'overhead e meno banda a disposizione (operazione utile nel caso in cui il canale abbia molti errori). Queste modifiche non sono però da prendere alla leggera, perchè **l'MTU** di **PPP interagisce con tutti gli altri protocolli** in tutti i flussi in atto; in pratica, **nell'ADSL sarebbe ideale un MTU grande, ma diminuendolo sotto certe soglie critiche la banda ne risente positivamente**.

PPP e ADSL

All'interno delle **ADSL** il PPP si usa principalmente in due varianti: **PPPoE** (**over Ethernet**) e **PPPoA** (**over ATM**), che **incapsulano PPP** rispettivamente dentro flussi Ethernet ed ATM. In particolare, possiamo analizzare il **viaggio** che un ipotetico **pacchetto PPP** deve percorrere **per arrivare alla rete Internet** dividendolo in varie parti:

1. flusso dati 1: **dal computer al router** (es. PPP → PPPoE → Ethernet);
2. flusso dati 2: **dal router al modem** (es. Ethernet → ATM);
3. flusso dati 3: **dal modem al primo DSLAM** (Digital Subscriber Line Access Multiplexer, in poche parole il provider) (es. ATM → LLC/VC-MUX);
4. flusso dati 4: **dal provider a un punto interno**;
5. flusso 5: **dal punto interno a Ethernet** (ricostruzione del flusso Ethernet);
6. flusso 6: **Internet**.

Nei flussi di competenza del **provider** solitamente il **primo tratto** è coperto da tecnologia **ATM**, mentre **l'ultimo** (quello più vicino a Internet) da **Ethernet**. Nel caso in cui il collegamento a Internet avvenga tramite mezzi wireless la situazione si complica ulteriormente.

ATM

Sta per **Asynchronous Transfer Mode** e usa **TDM**, dividendo i dati in un flusso di celle di ampiezza fissa. Si può considerare come **analogo di HDLC** ma nato per **telefonia/bancomat**. Gestisce il **controllo di flusso** con delle **sliding windows**, fa **error detection** con **CRC-8** e gestisce l'indirizzamento.

L'indirizzamento è gestito attraverso una **doppia gerarchia**, divisa tra cammini (path) e canali (channel), che corrispondono alle voci **VPI (Virtual Path Identifier)** e **VCI (Virtual Channel Identifier)** che troviamo nelle configurazioni ADSL. Inoltre, come ci segnala l'uso di termini come Virtual, ATM è un **protocollo connection-oriented**.

Dato che stiamo parlando di canali multiplex, che interagiscono quindi con altri flussi dati, non è strano che anche **ATM** venga **embeddato** a sua volta **dentro** altri tipi di **protocolli per il multiplex**:

- **LLC (Logical Link Control)**: supporta protocolli multipli per canale;
- **VC-MUX (Virtual Connection Multiplexing)**: supporta un solo protocollo per canale.

Ciclo iniziale PPPoE

La **connessione all'ADSL** inizia col nostro **computer/modem** che invia un **frame PPPoE Active Discovery Initiation** contenente il suo **indirizzo fisico (MAC)**. A questo punto, **ogni DSL-AC** ("concentratore di accessi", servizio ADSL) disponibile **risponde** con un **PADO (PPPoE Active Discovery Offer)** in cui si offre per la connessione, dando il proprio indirizzo. Il **computer** risponde allora con una **PPPoE Active Discovery Request (PADR)** in cui **segnala il servizio ADSL** che ha **scelto**; **servizio** che fa quindi l'ACK usando un frame **PADS (PPPoE Active Discovery Session-confirmation)**. Quando sarà il momento, la **connessione** verrà **terminata** con un frame **PADT**, cioè **PPPoE Active Discovery Termination**.

I contention systems

Abbiamo visto finora protocolli che regolamentano comunicazioni point-to-point; i **contentions system** sono invece quei sistemi di comunicazione in cui c'è un **unico canale condiviso** da molti, e si possono per questo creare **contenziosi** (contentions appunto).
Alla base di questi sistemi ci sono delle **assunzioni**, riguardanti vari aspetti. Sappiamo per esempio che si parla di **single channel**, quindi un canale singolo disponibile per tutti, e di **collision**, cioè il fatto che nel caso in cui **due frame si sovrappongono**, diventano **inutilizzabili**. Un'altra assunzione possibile riguarda il **tempo**, che può essere **continuo** (quindi senza un orologio centralizzato sulla base del quale accordarsi) o **slotted** (diviso cioè in intervalli). Per quanto riguarda il canale invece, possiamo dotare o meno le varie stazioni del **Carrier Sense**, cioè la possibilità di **conoscere o meno lo stato del canale prima di utilizzarlo** per trasmettere.

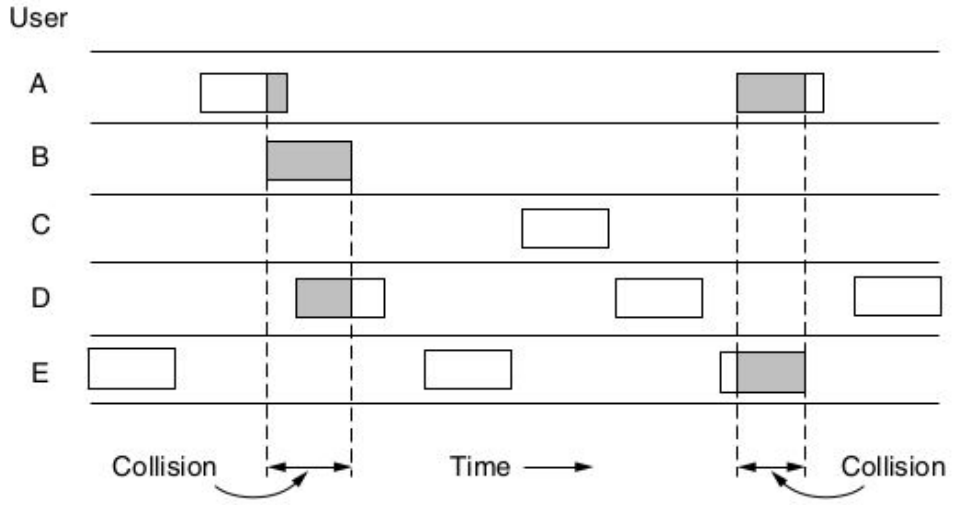
Sistemi cablati

ALOHA

Negli anni '70, Norman Abramson ideò un metodo innovativo per risolvere il **problema dell'assegnazione del canale**. Il progetto di Abramson, chiamato sistema ALOHA, utilizza la trasmissione radio broadcast basata su stazioni terrestri, ma l'idea di fondo può essere applicata a qualunque sistema dove utenti non coordinati competono tra loro per utilizzare un singolo canale condiviso. Esistono due versioni di ALOHA: **puro e slotted**, entrambi **no carrier sense**.

ALOHA puro

L'idea di fondo del sistema ALOHA è semplice: consentire agli utenti di t**rasmettere ogni volta che hanno dati da inviare**. Naturalmente ci potranno essere **collisioni** che danneggeranno i frame, ma un trasmettitore potrà sempre scoprire se il suo frame è andato distrutto. Se per qualche motivo non è possibile ascoltare il canale durante la trasmissione, si deve adottare un sistema di acknowledge. Se il **frame** è stato **distrutto**, il **trasmettitore** rimane **in attesa per un intervallo casuale** prima di ripetere la trasmissione. Il tempo di attesa deve essere casuale, altrimenti gli stessi frame continueranno a collidere in un ciclo senza fine.



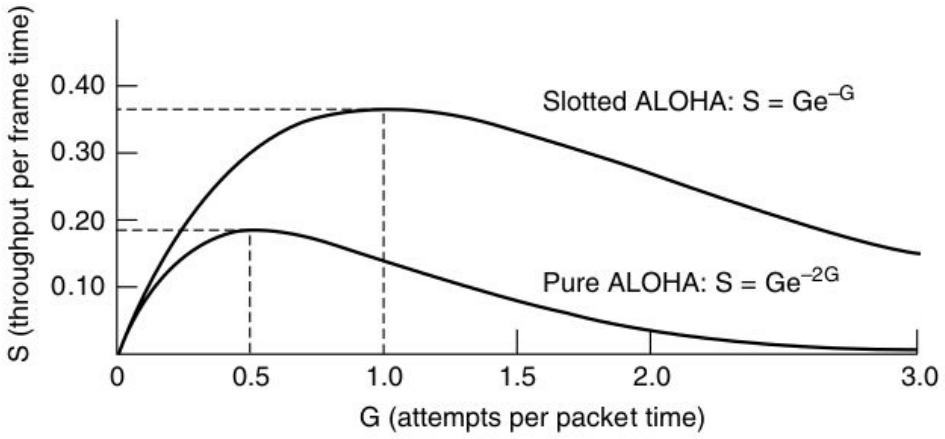
Tutti i frame hanno la **stessa lunghezza** perché la **capacità di trasporto** dei sistemi ALOHA è **massima** quando si utilizza un **frame con dimensione uniforme**. Ogni volta che due frame tentano di occupare contemporaneamente il canale si verifica una collisione che danneggia entrambi gli elementi. Basta che il primo bit di un nuovo frame si sovrapponga all'ultimo bit di un frame quasi completato, per considerarli entrambi totalmente distrutti e quindi da ritrasmettere. Per calcolare l'efficienza di questo protocollo, si consideri prima di tutto un gruppo infinito di utenti interattivi seduti davanti ai loro computer (stazioni). Un utente si trova sempre in uno di due stati: sta scrivendo oppure è in attesa. Inizialmente tutti gli utenti si trovano nel primo stato, ossia stanno scrivendo. Quando termina una riga l'utente smette di scrivere e attende una risposta; a questo punto la stazione trasmette un frame che contiene la riga e controlla il canale per scoprire se la trasmissione ha avuto successo. In caso positivo, l'utente vede apparire la risposta e ricomincia a scrivere; in caso negativo, l'utente continua ad aspettare e il frame è ritrasmesso fino a quando la trasmissione non ha successo. Si supponga che il tempo di frame indichi la quantità di tempo richiesta per trasmettere un frame di lunghezza standard e fissa (pertanto è pari alla lunghezza del frame divisa per il bit rate). A questo punto si può ammettere per ipotesi che la popolazione infinita di utenti generi nuovi frame in accordo con la distribuzione di Poisson, con media di N frame per tempo di frame. Il presupposto della popolazione infinita garantisce che N non diminuisce quando gli utenti cominciano a essere bloccati. Se N > 1, la comunità di utenti sta generando frame a una frequenza più elevata della frequenza che il canale è in grado di gestire,

e quasi ogni frame subirà una collisione. Per garantire una ragionevole capacità di trasporto si dovrà assumere O < N < 1. Oltre ai nuovi frame, le stazioni ritrasmettono anche i frame entrati precedentemente in collisione. Supporremo di Poisson anche la probabilità di k tentativi di trasmissione per tempo di frame (il vecchio e il nuovo combinati insieme), con media G per tempo di frame. Chiaramente, G≥N. A basso carico (N≈0) ci saranno poche collisioni e di conseguenza poche ritrasmissioni, perciò G≈N. A carico elevato ci saranno molte collisioni, perciò G>N. Se G è il carico che si presenta, qualunque sia il carico la capacità di trasporto S sarà G volte la probabilità P_0 di una trasmissione che ha successo, ossia $S=GP_0$, dove P_0 è la probabilità che un frame non entri in collisione con un altro frame. In quali condizioni il frame evidenziato arriverà a destinazione senza riportare alcun danno? Si indichi con t il tempo richiesto per inviare il frame: la parte iniziale del frame evidenziato entra in collisione con la fine di un qualsiasi altro frame generato da un utente nel tempo compreso tra t_0 e t_0+t. Il destino del frame evidenziato era segnato già prima che iniziasse la trasmissione del primo bit, ma poiché in un sistema ALOHA puro una stazione non ascolta il canale prima di iniziare a trasmettere, non c'è modo di sapere che un altro frame è già stato inviato. Analogamente, i frame che inizieranno nel periodo compreso tra t_0+t e t_0+2t incapperanno sicuramente nella fine del frame evidenziato.
La probabilità che k frame siano generati durante un dato tempo di frame è data dalla distribuzione di Poisson: $Pr[k] = \frac{G^k e^{-G}}{k!}$ perciò la probabilità di zero frame è e^{-G}. In un intervallo lungo due tempi frame, il numero medio di frame generati è pari a 2G. La probabilità che nessun altro traffico inizi durante l'intero periodo vulnerabile è perciò data da $P_0=e^{-2G}$. Usando $S=GP_0$ si ottiene $S=Ge^{-2G}$. La capacità di trasporto massima si ha per G=0,5 con S=1/2e, ed è circa 0,184. In altre parole, si può **sperare di utilizzare al massimo il 18% del canale**, un dato in assoluto basso, ma buono in relazione al fatto che **non dipende dal numero di stazioni che possono trasmettere**.

Slotted ALOHA

Dividendo il tempo in intervalli discreti, dove **ogni intervallo corrisponde a un frame**, la **capacità** di un sistema ALOHA **raddoppia**. Questo approccio richiede però che gli utenti concordino i limiti degli intervalli, e la necessaria sincronizzazione si può soddisfare con una speciale stazione che emette un segnale all'inizio di ogni intervallo, proprio come un orologio. In questo metodo, chiamato slotted ALOHA, c'è una differenza importante rispetto all'ALOHA puro: un computer non può inviare dati ogni volta che viene premuto il tasto di avanzamento riga. Quando l'utente completa una riga il computer deve attendere l'inizio dell'intervallo (slot) successivo e ciò trasforma il sistema continuo ALOHA puro in un **sistema discreto**. Poiché ora il periodo vulnerabile è dimezzato, la probabilità che non ci sia altro traffico durante lo stesso intervallo occupato dal frame di prova è pari a e^{-G}, quindi: $S=Ge^{-G}$.
Slotted ALOHA ha il massimo a G = 1, con una capacità di trasporto pari a S=1/e che equivale a circa 0,368, il doppio del valore di ALOHA puro. Se il sistema sta

operando a G=1, la probabilità che si presenti un intervallo vuoto è di 0,368. La **migliore situazione** che si può ottenere con slotted ALOHA prevede il 3**7% di intervalli vuoti**, il **37% di trasmissioni senza errori** e il **26% di collisioni**. Operando con valori di G più alti si riduce il numero di intervalli vuoti ma aumenta esponenzialmente il numero di collisioni. Per osservare con quanta rapidità può crescere il numero di collisioni si consideri la trasmissione di un frame di prova: la probabilità che esso eviti la collisione è e^{-G}, valore uguale alla probabilità che tutti gli altri utenti non trasmettano nello stesso intervallo; la probabilità di una collisione è allora $1-e^{-G}$; la probabilità che una trasmissione richieda esattamente k tentativi (k-1 collisioni seguite da un successo) sarà $P_k = e^{-G}(1 - e^{-G})^{k-1}$.



CSMA

Sta per **Carrier Sense Multiple Access** ed esiste in 3 varianti principali.

1-persistent CSMA

Quando ha dei dati da trasmettere, una **stazione** prima di tutto **ascolta il canale** per scoprire se qualcun altro in quel momento sta trasmettendo. Se il canale è occupato, la stazione aspetta fino a quando non si libera. Quando si accorge che il **canale** è **libero trasmette** un frame; in caso di **collisione**, la stazione rimane in **attesa** per un **intervallo casuale** prima di ritentare la trasmissione. Il protocollo è chiamato 1 -persistent perché la stazione trasmette con una **probabilità di 1 quando trova il canale libero**. Il ritardo di propagazione ha un effetto importante sulle prestazioni del protocollo: c'è infatti una piccola possibilità che subito dopo l'inizio di una trasmissione da parte di una stazione, un'altra stazione sia pronta a inviare dati e controlli il canale. Se il segnale della prima stazione non ha ancora raggiunto la seconda, quest'ultima potrebbe ritenere il canale libero e iniziare perciò a trasmettere i propri dati, causando una collisione. Più è lungo il **ritardo di propagazione**, più diventa importante l'effetto e **peggiori** le **prestazioni** del protocollo. Le collisioni continuano a verificarsi anche se il ritardo di propagazione è uguale a zero. Se due stazioni diventassero pronte mentre una terza sta trasmettendo, entrambe attenderebbero educatamente il termine della trasmissione prima di iniziare la loro emissione simultanea; il risultato finale, anche in questo

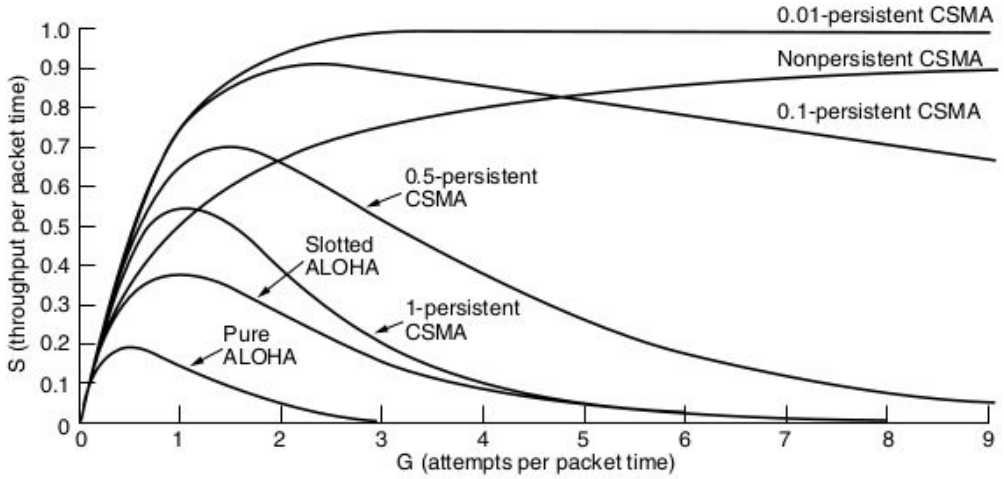
caso, sarebbe una collisione. Se non fossero così impazienti ci sarebbero meno collisioni. Anche così, questo **protocollo** è **migliore** del sistema **ALOHA puro** e conduce a prestazioni migliori perché entrambe le stazioni hanno la decenza di desistere dall'interferire con il frame della terza stazione.

p-persistent CSMA

Si applica ai **canali divisi in intervalli temporali** e funziona così: quando è pronta a trasmettere, **ogni stazione controlla il canale**. Se lo trova **libero**, **trasmette** subito con una **probabilità p**, altrimenti **rimanda** fino all'**intervallo successivo** con **probabilità q=1-p**. Se anche quell'intervallo risulta libero, la stazione trasmette oppure rimanda un'altra volta (anche in questo caso le probabilità sono p e q). Il processo si ripete **fino a quando il frame non è stato trasmesso** o un'altra stazione non inizia a trasmettere; in questo caso, la stazione sfortunata si comporta come se ci fosse stata una **collisione** (ossia attende per un **intervallo di tempo casuale** e poi ricomincia). Se inizialmente rileva che il canale è occupato, la stazione attende fino all'intervallo successivo e poi applica l'algoritmo appena descritto.

Nonpersistent CSMA

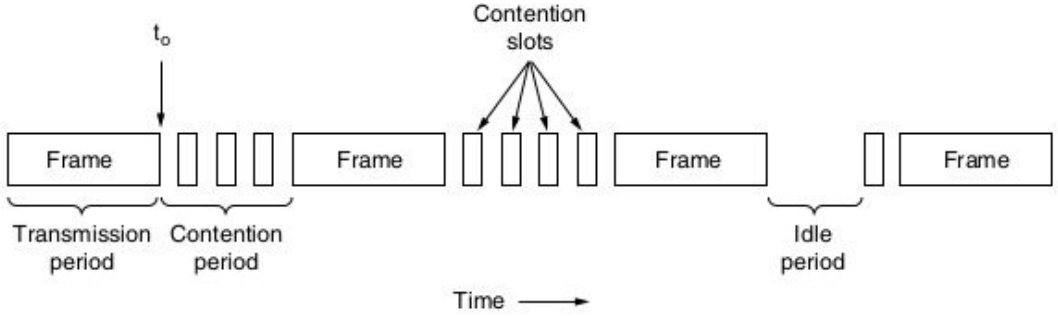
Prima di trasmettere, ogni stazione controlla il canale: se nessun altro sta trasmettendo inizia a inviare i dati, ma se il **canale** è **occupato** la stazione non esegue un controllo continuo per trasmettere subito il proprio frame; invece attende per un **intervallo di tempo casuale** prima di ripetere l'algoritmo. Di conseguenza, questo meccanismo permette di utilizzare meglio il canale ma **allunga i ritardi**.



CSMA/CD

Un nuovo miglioramento si ottiene consentendo a **ogni stazione** di **annullare** la propria **trasmissione in caso di collisione**. In altre parole, se due stazioni, dopo aver controllato il canale, iniziano a trasmettere contemporaneamente, entrambe rileveranno la collisione quasi immediatamente. Invece di completare la trasmissione dei rispettivi frame, oramai irrimediabilmente danneggiati, le stazioni interrompono bruscamente la trasmissione. La terminazione rapida dei frame danneggiati **risparmia tempo e banda**. Questo protocollo, chiamato CSMA/CD (**CSMA con**

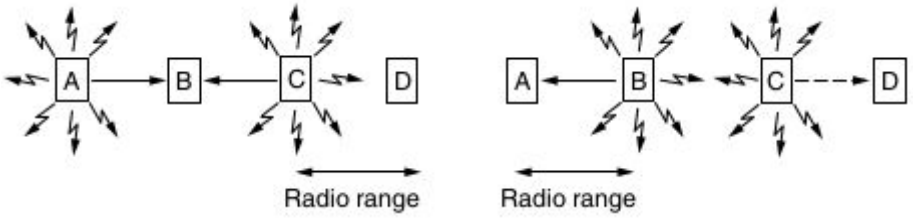
Collision Detection), è ampiamente utilizzato nel sottostrato MAC delle LAN e in particolare costituisce la base delle famose LAN Ethernet.
All'interno di questo protocollo, possiamo separare **3 possibili stati della rete: trasmissione, contention e idle**.



La parte più importante è quella chiamata **contention**, cioè quella in cui si decidono le sorti della trasmissione: il tempo da allocare a questa fase sarà **due volte il tempo di roundtrip** (tempo di trasmissione tra le due stazioni più distanti).

Sistemi wireless

Nel caso del wireless, ulteriori **difficoltà** sono dovute al fatto che la **topologia di rete non** è più **fissa**, ma cambia in modo dinamico; in questo senso, il **controllo** passa da globale a **locale**, in quanto non c'è più un singolo canale per tutti, ma varie zone spaziali dove alcune stazioni interagiscono e altre no.



Vediamo che cosa accade se A trasmette a B, come mostrato nella figura. Se C controlla la presenza di portante sul mezzo di trasmissione non potrà rilevare A, perché si trova al di fuori della sua portata; di conseguenza C pensa erroneamente di poter trasmettere a B. Se inizia a trasmettere, C interferisce con B distruggendo il frame inviato da A. Il problema di una **stazione** che **non è in grado di rilevare** i potenziali **concorrenti** sul mezzo di trasmissione a causa della distanza eccessiva è chiamato **problema della stazione nascosta**.
Ora si consideri la situazione inversa: B trasmette ad A. Se C controlla la presenza di portante sul mezzo di trasmissione, rileva una trasmissione in atto ed erroneamente pensa di non poter inviare dati a D; in realtà la trasmissione causerebbe una cattiva ricezione solo nella zona compresa tra B e C, che non ospita nessuno dei ricevitori designati. La situazione genera quello che è chiamato **problema della stazione esposta**.

MACA

Uno dei primi protocolli progettati per le LAN wireless è stato MACA (**Multiple Access with Collision Avoidance**). L'idea di base è semplice: il trasmettitore incita

il ricevitore a trasmettere un **piccolo frame** in modo che le **stazioni** che si trovano nelle vicinanze, rilevando questa trasmissione, **evitino di inviare dati** durante la trasmissione imminente del frame di dati più grande. Si supponga che A invii un frame a B. A invia prima di tutto un frame **RTS** (**Request To Send**) a B. Questo piccolo frame (**30 byte**) contiene la **lunghezza del frame di dati** che verrà inviato successivamente. B risponde con un frame **CTS (Clear to Send**, corrisponde a un generico **ACK**), che contiene la lunghezza dei dati copiata dal frame RTS. Non appena riceve il frame CTS, A inizia la trasmissione. **Tutte le stazioni** che ricevono il frame **RTS** si trovano chiaramente **vicine ad A**, e devono rimanere in **silenzio** per un tempo che consenta al frame CTS di raggiungere A senza causare conflitti. Le stazioni che ricevono il frame **CTS** sono chiaramente **vicine a B**, e devono rimanere in **silenzio** durante la trasmissione dei dati la cui lunghezza può essere determinata esaminando il frame CTS; una volta **terminata la trasmissione**, le **stazioni circostanti** attendono un tempo casuale per ritrasmettere (**ALOHA non-persistente**).

MACAW

Acronimo di **MACA per Wireless**, è una versione migliorata del protocollo precedente; le aggiunte principali sono le seguenti:

- vista l'assenza di ACK nello strato data link, i frame perduti erano ritrasmessi dopo molto tempo; per questo motivo, viene **introdotto** un **ACK dopo ogni frame trasmesso con successo**;
- viene utillizzato **CSMA** per **controllare se una stazione vicina sta già trasmettendo** un frame RTS alla stessa destinazione.

Ethernet

Standard IEEE 802.3

Ne esistono vari tipi, nominati **XBaseY**, dove **X** indica la **banda** in Mbps, **base** indica che è una connessione **baseband** (a frequenza unica) e **Y** indica il **tipo di cavo**. I vari tipi di Ethernet differiscono anche a seconda della massima lunghezza di ogni tratto senza ripetitori. Le prime versioni di Ethernet si evolvero, a parità di velocità, diminuendo la lunghezza massima ma aumentando notevolmente il numero delle stazioni collegabili; con Ethernet 10BaseT la velocità era di 10Mbps, si utilizzava un cavo twisted pair lungo massimo 100 metri, ma si potevano connettere ben 1024 stazioni. Esiste inoltre un tipo di **Ethernet 10Base-F**, che usa la **fibra ottica** permettendo la creazione di segmenti lunghi fino a 2 km; inoltre, essendo anche molto fine, rappresenta il tipo ideale per la connessione tra edifici.

Codifica Manchester

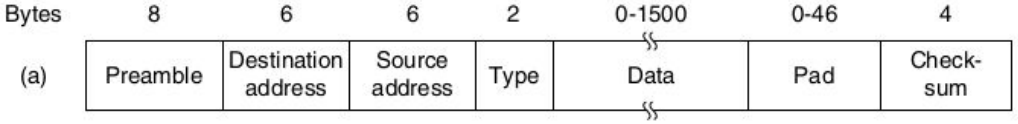
Per **evitare problemi di sincronizzazione** nel momento in cui si devono codificare/decodificare segnali fisici (quindi 0 e 1), si ricorre all'uso della codifica Manchester.

La strategia consiste nell'**inviare un segnale di clock separato al ricevitore**. Una nuova linea di clock potrebbe rappresentare uno spreco per la maggior parte dei collegamenti di rete, visto che nel momento in cui abbiamo una nuova linea per inviare un segnale potremmo utilizzarla per l'invio di dati. L'idea quindi è di mixare il **segnale di clock col segnale dati** facendo uno **XOR** tra i due, in modo da non aver bisogno di una linea extra. Così facendo, il **clock** compie una **transizione per ogni bit**; in particolare, quando avviene uno XOR con uno **0** compierà una transizione **dal basso verso l'alto**, e viceversa in caso contrario (XOR con **1**) una transizione **dall'alto verso il basso**. Così facendo, **eliminiamo** la necessità di un **orologio** su cui controllare lo stato della trasmissione, in quanto ci accorgiamo immediatamente in base alla transizione compiuta se ci troviamo di fronte a uno 0 o a un 1. Lo svantaggio della codifica Manchester risiede però nel fatto che in questo modo la **banda** viene **dimezzata**.

Codifica Manchester differenziale

Per esempio, per inviare dati a 10 Mbps il segnale deve cambiare 20 milioni di volte al secondo. La Figura 4.16(b) mostra la codifica Manchester.
La codifica Manchester differenziale, mostrata nella Figura 4.16(c), è una variante della codifica Manchester elementare. Il bit 1 è indicato dall'assenza di una transizione all'inizio dell'intervallo; il bit 0 è indicato dalla presenza di una transizione all'inizio dell'intervallo. In entrambi i casi, c'è una transizione nel punto centrale. Lo schema differenziale richiede dispositivi più complessi ma offre una maggiore immunità ai rumori. Tutti i sistemi Ethernet adottano la codifica Manchester perché è più semplice. Il segnale alto ha una tensione di +0,85 Volt e il segnale basso ha una tensione di -0,85 Volt, con il valore DC pari a 0 Volt. Ethernet non utilizza la codifica Manchester differenziale, altre LAN invece sì (per esempio la 802.5 token ring).

Struttura del frame Ethernet



- **Preamble**: 8 bytes 10101010 per la **sincronizzazione**;
- **Destination address**: indirizzo di **destinazione**, 6 bytes. Il primo bit di questi segnala una comunicazione **multicast**; se **tutti i bits** sono a **1** si tratta di una comunicazione **broadcast**. Il **secondo bit** invece distingue **indirizzi locali da globali**;
- **Type**: specifica il tipo di **protocollo** o l'**uso del frame**;
- **Pad**: serve ad assicurare che la **lunghezza minima** del frame sia **almeno il tempo di roundtrip**;
- **Checksum**: **CRC-32**, fa error detection ma **non error correction**.

MAC Address

Precisamente MAC-48, rappresentano uno **spazio di indirizzi** gestito dall'IEEE. I **primi 3 bytes** di un indirizzo MAC rappresentano il **produttore** (**OUI, Organizationally Unique Identifier**).

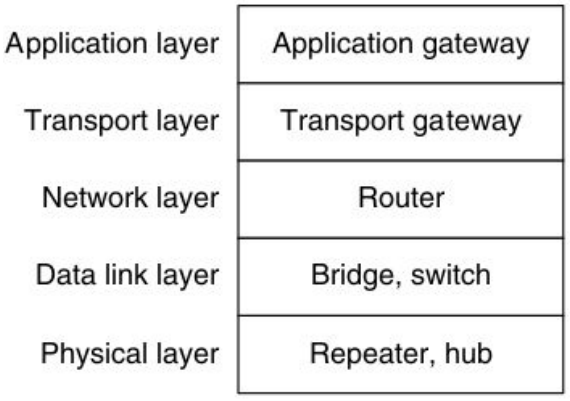
Collision detection

In caso di **collisione** si usa **ALOHA** in una speciale versione dinamica (**binary exponential backoff**) che funziona nel seguente modo: **aumenta esponenzialmente il tempo di attesa massimo** e fa ritrasmettere a caso fino a che la trasmissione non va a buon fine. Nella **versione** cosiddetta **truncated** del binary exponential backoff ci si **ferma dopo 10 collisioni**, mantenendo l'intervallo massimo a 1023 slots per altre 10 collisioni e poi eventualmente si rinuncia.

Problema di Ethernet

Il **problema principale** di **Ethernet** è che la sua efficienza è inversamente proporzionale al prodotto bandwidth*lunghezza, quindi, **quanto più aumentiamo la banda o la lunghezza, tanto più l'efficienza diminuisce**. Per questo motivo, la soluzione migliore è **limitare la lunghezza massima della rete**.

I "collanti" di rete



Dati i limiti di lunghezza dei simgoli componenti dei livelli di ogni rete (come appena visto per l'Ethernet), per **ogni livello** esistono dispositivi o modi che aiutano a **collegare più componenti**, per darci un'infrastruttura di rete ampia come quella che siamo abituati a conoscere.

Repeaters e hubs

Per quanto riguarda lo **strato fisico** troviamo questi due dispositivi:

- il **ripetitore ripete** (generalmente amplificandolo in potenza) il segnale;
- **l'hub** (che può essere anche repeater) **propaga** il segnale **da una porta a tutte le altre**.

Bridge/Switch

Fa parte del **livello data link** e quindi interagisce con la struttura dei frame. Crea una **rete più grande a livello logico**, rimuovendo i limiti di grandezza della rete e stabilendo diversi domini di collisione. Per fare ciò, lo switch deve ispezionare i pacchetti e filtrare il traffico; ha quindi bisogno di una **fase di learning**, in cui impara la configurazione di rete e gestisce il traffico corrispondentemente. Inoltre, essendo a livello data link, controlla mittente e destinazione dei frame.
Il processo messo in atto per imparare quanto detto si chiama **backward learning** e porta alla costruzione delle cosiddette **hash tables**, cioè **tabelle** in cui **ogni macchina** è **associata** alla sua (sotto) **rete di riferimenti** e che vengono costruite analizzando il flusso di dati e risposte, per poi fare broadcasting nella fase intermedia.
Negli switch è anche implementato un **timeout** (**di fading**) per ovviare ai **cambiamenti nella topologia di rete**, allo scadere del quale le **hash tables** vengono **cancellate** per essere ricostruite nuovamente.

Routers

I routers fanno parte dello **strato network**, che si occupa di **portare a destinazione i pacchetti** su una rete complessa. Per portare un pacchetto a destinazione attraversando il **percorso** migliore ci serve un **concetto di distanza**, che inizialmente possiamo approssimare contando il numero di hops, cioè di stazioni incontrate nel cammino. Si possono anche ottenere concetti di distanza più sofisticati, pesando ad esempio ogni cammino tenendo conto del tempo di trasmissione.

Flooding

Il flooding costituisce un **metodo di routing molto potente**, e consiste nel **ritrasmettere ogni pacchetto a tutte le linee** (esclusa quella di origine). Ovviamente, per non creare problemi, il flooding ha bisogno di **controllo**, e a tal proposito una delle tecniche è rappresentata dall'**hop counting**: si associa cioè un **numero massimo di hop ad ogni pacchetto**, dopo il quale il pacchetto muore.
Una tecnica alternativa a questa è quella del **tracking**: si tiene cioè **traccia dei pacchetti che sono già stati trasmessi** per non ritrasmetterli nuovamente; si possono anche tenere liste separate per ogni router, tenendo un contatore speciale per la lista dei pacchetti 0-N che sono stati già ricevuti.
Nonostante queste tecniche, rimane l'ovvio **svantaggio** che il flooding genera una **enorme quantità di pacchetti**.
I **vantaggi** del flooding sono però **notevoli**: questa tecnica è infatti in grado di **scegliere sempre la strada migliore** e costituisce il **più robusto sistema** di routing possibile. Queste caratteristiche rendono il flooding utilissimo nei casi in cui il carico di rete non è molto alto, o quando è fondamentale che un messaggio arrivi sempre nel minor tempo possibile (es. ambito militare).

Distance vector routing

L'idea alla base di questa tecnica di routing (utilizzata in ARPANET, la prima versione di Internet) è che **ogni router** ha una **tabella** contenente informazioni su **quanto veloce è la connessione ad un altro router e qual è la via migliore per raggiungerlo**. Ogni router chiede quindi ai **propri vicini** le loro **tabelle**, che usa, insieme al **tempo impiegato per riceverle**, per **costruire la sua tabella** selezionando i percorsi migliori.
Il principale vantaggio e il principale svantaggio di questa tecnica sono dati dal fatto che è **tanto veloce a recepire cambiamenti della rete in positivo, quanto lenta a fare lo stesso in caso di peggioramenti**.

Link State Routing

Per risolvere il problema appena visto, viene introdotta una nuova tecnica, chiamata Link State Routing, che funziona nel seguente modo: quando ha un'**informazione completa** sui suoi vicini, ogni nodo costruisce un **pacchetto** che contiene questa informazione e la manda in **broadcast** (**flooding con refresh periodico**: si spreca più banda ma garantisce più robustezza) a tutti gli altri. In questo modo, **ogni nodo** riceverà le informazioni locali degli altri, e sarà quindi in grado di costruire una **mappa completa della rete**, calcolando i percorsi migliori.

Quality of Service

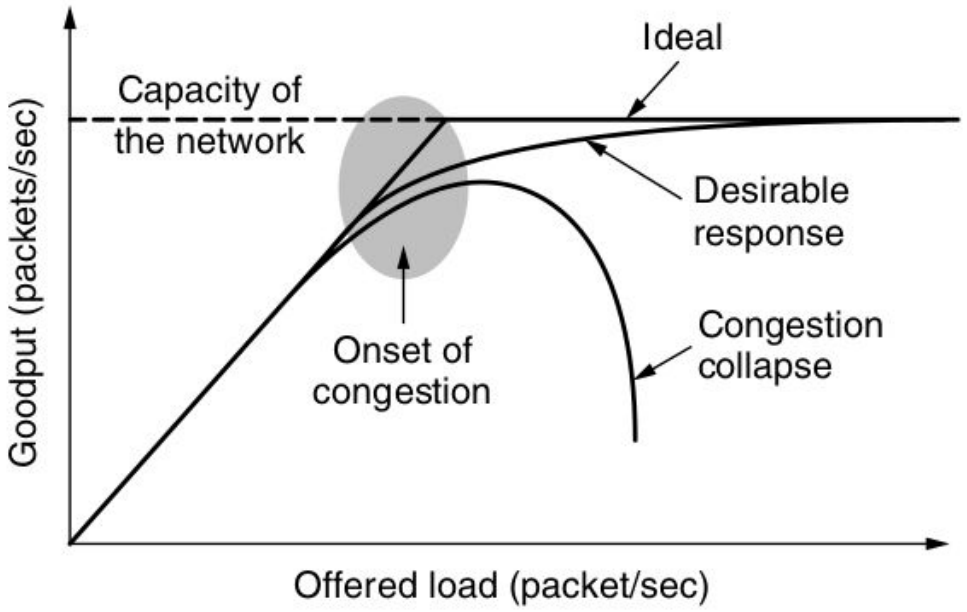
La qualità del servizio è tipicamente data da **4 parametri principali** che dettano la qualità del servizio offerto:

- **Reliability** (affidabilità);
- **Bandwidth** (banda);
- **Delay** (ritardo di trasmissione);
- **Jitter** (grado di variazione nei tempi di arrivi dei pacchetti).

L'affidabilità di un servizio si può garantire in modo relativamente semplice, attraverso error control e error correction, mentre i problemi riguardanti gli altri parametri derivano da alcune cause, che devono essere risolte se si vuole mantenere una buona QoS.

La congestione

Una delle principali cause di problemi in questo senso è la congestione, cioè ciò che accade quando la **capacità della linea** o di qualche stazione si **satura**; per questo motivo, è importante fare **congestion control**.



Quello da affrontare è in realtà un problema molto complesso, in quanto **aggiungere capacità ad una rete può** addirittura **portare ad una diminuzione della performance** (paradosso di Braess).

I choke packets

Il **ricevente** può segnalare casi di **congestione** inviando un **pacchetto speciale** (choke packet) a chi sta inviando dati, dicendogli di rallentare; tipicamente, non appena riceve un choke packet, un **host dimezza il suo data rate**. Per uscire dal choke si usa il **fading**: se non riceviamo choke packets dopo un certo periodo di tempo torniamo ad incrementare la nostra velocità di trasmissione. Allo stesso modo, per non diminuire troppo il data rate, una volta **ricevuto un choke** si **ignorano i successivi** provenienti dalla stessa direzione.

Choke hop-by-hop

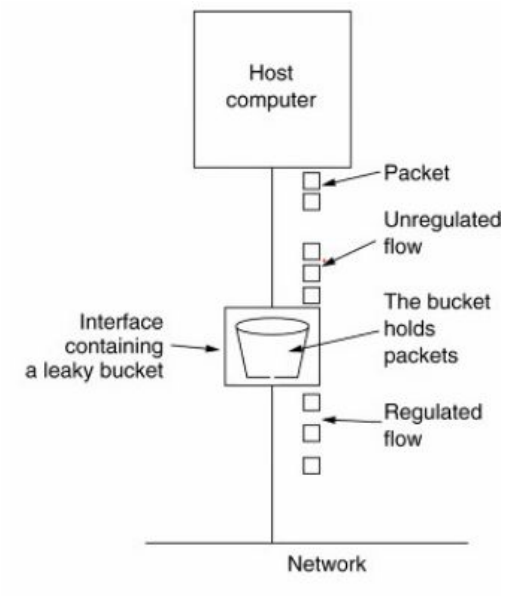
In una rete molto grande, una richiesta di choke può impiegare troppo tempo a muoversi tra gli host; per ovviare a questo problema si usa una variante del choke, chiamata choke hop-by-hop. Questo tipo di choke ha la particolarità di **non rallentare solo il suo destinatario, bensì tutti i router che attraversa**.

I buckets

Rappresentano dei **filtri**, utili a garantire buone proprietà in termini di QoS (in particolare limitando la congestione) e che generalmente sono gestiti da chi invia i dati.

Leaky Buckets

I leaky buckets rappresentano un tipo di filtro che garantisce un **data rate massimo costante**, **evitando** quindi **i burst** che potrebbero sovraccaricare la rete.



Il leaky bucket ha però il **vantaggio/svantaggio** di avere un **data rate massimo costante**, nonostante alle volte, in caso di traffico intenso, sarebbe meglio **aumentarlo**.

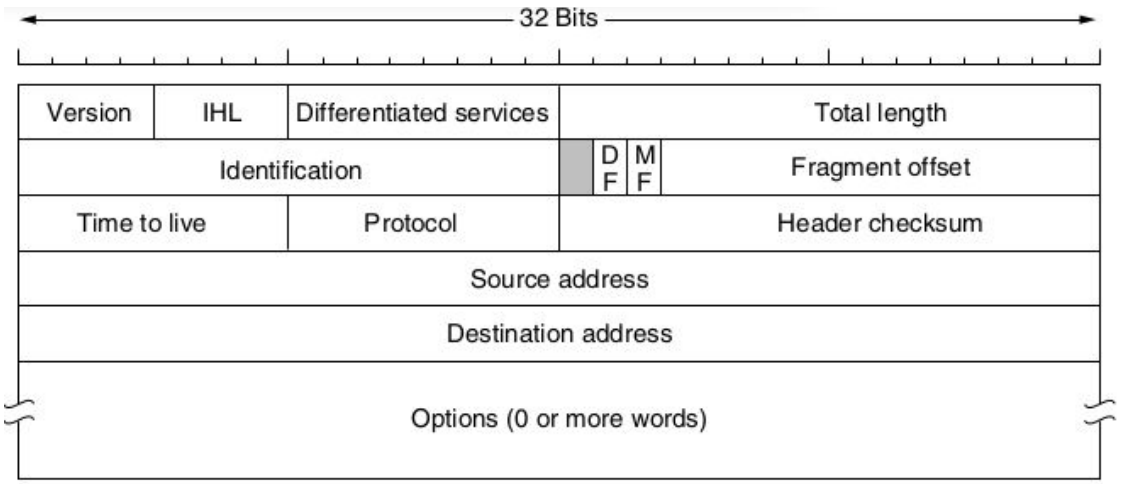
Token Buckets

Per ovviare a questo problema esiste appunto il **token bucket**, che genera un **token ogni certo intervallo di tempo**; i **pacchetti** che arrivano possono **uscire** dal secchio **solo se "bruciano" un token** disponibile. In questo modo, se il traffico è lento per un certo periodo ed è successivamente seguito da un **burst**, sarà più facilmente gestibile, consumando i **token accumulati** durante il periodo di tranquillità.

Strato di rete

Protocollo IP

Header IP



Ha una **parte** di **lunghezza fissa 20 byte** e una parte a **lunghezza variabile**. Analizziamo i vari campi:

- **Version**: specifica la **versione** del protocollo in uso;
- **IHL (IP Header Length)**: specifica la **grandezza dell'header**;
- **Differentiated services**: **tipo di servizio**, utile per indicare la QoS ma spesso **ignorato**;
- **Total length**: **lunghezza totale** compresi i dati;
- **Identification**: identifica se i **dati** sono stati **frammentati** in più pacchetti;
- **Evil bit**: flag standardizzato con valore 1 nel caso in cui un pacchetto abbia **intenti malevoli** (joke);
- **DF (Don't Fragment)**: flag che impone la **non-frammentazione** dei dati;
- **MF (More Fragment)**: flag che segnala l'**ultimo frammento**;
- **Fragment offset**: il **numero d'ordine del frammento**;
- **Time to live (TTL)**: **età massima del pacchetto** dopo la quale il pacchetto muore;
- **Protocol**: indica il **protocollo di più alto livello** che sta usando IP;
- **Header Checksum**: calcolato tramite **somme in complemento a 1**;
- **Source address**: **indirizzo** di chi **invia**;
- **Destination address**: **indirizzo** di chi **riceve**;
- **Options**: parte variabile per **eventuali opzioni**.

Limiti di IP

IP risulta debole e quasi impreparato per l'utilizzo che ne viene fatto oggi, in quanto le **misure** prese per la **sicurezza** sono quasi **inesistenti**. Il motivo di ciò risiede nel fatto che è stato originariamente pensato per un **modello closed world** (rete militare

del DoD USA), in cui si ha il controllo completo del protocollo e gli attacchi interni sono altamente improbabili. Per questo motivo, IP non sembra adatto a modelli **open-world**, in cui l'eventualità di subire **attacchi interni** non è così remota e la gestione della congestione del traffico (altro punto debole di IP) è fondamentale.
Altri limiti di IP sono costituiti dal **TTL limitato a 255 hops** (problema risolvibile dal router quindi non molto preoccupante) e dalla **lunghezza degli indirizzi** source e destination (32 bits) il cui spazio globale è fisso e non estendibile (cioè **stanno per finire** gli indirizzi a disposizione).

Indirizzi IP

Per gestire al meglio il **numero limitato di indirizzi IP** a disposizione, si sono suddivisi in una **struttura gerarchica**: invece di assegnare un solo indirizzo alla volta, assegnamo interi **blocchi di indirizzi** alle varie organizzazioni. Per definire la grandezza di questi blocchi, all'inizio di Internet si è usato il metodo del **classful addressing**, che ha consentito la creazione di **blocchi di taglie diverse** a seconda della grandezza della rete richiesta dalle singole organizzazioni. In particolare, si sono definite tre classi (blocchi) di indirizzi: **classe A** (**16,7 milioni di indirizzi**), **classe B** (**65536 indirizzi**) e **classe C** (**256 indirizzi**). Il problema principale che ha fatto sì che gli indirizzi IP si stiano esaurendo in fretta, era dato dal fatto che molte organizzazioni per le quali blocchi di classe C sarebbero stati ideali hanno preferito acquistare blocchi di classe B, utilizzando pochissimi indirizzi.

CIDR

Una delle **possibili soluzioni** al problema degli indirizzi IP risiede nel **Classless InterDomain Routing**, che prova a superare il concetto di classi di indirizzi utilizzando **blocchi a lunghezza variabile**. Il CIDR permette ai router una gestione abbastanza efficiente, usando le cosiddette **aggregate entries**: se ci sono cioè varie classi di indirizzi che devono essere indirizzati allo stesso router, possono essere combinate in un'unica entrata se hanno un prefisso comune; se però c'è un'organizzazione con lo stesso prefisso di altre che deve però essere mandata ad un altro router, si usa la regola del longest matching, cioè l'entrata con il prefisso di rete più lungo ha la priorità.

NAT

Sta per **Network Address Translation**, e ha alla base l'idea di **simulare una sottorete** usando **un solo indirizzo IP**. In pratica, la rete all'interno funziona con degli indirizzi IP invisibili all'esterno, dove invece la rete appare come un singolo indirizzo IP. Quando un **pacchetto esce** dalla rete, il suo **indirizzo IP (interno)** viene **sostituito** dall'**unico indirizzo** del NAT. Nel **verso opposto**, invece, il NAT usa le **porte del livello di trasporto** per fare **multiplexing a livello di rete**: in pratica, abusa del concetto di porta e fa sì che uno spazio delle porte (1-65536) possa essere associato a molti indirizzi IP invece che a uno solo. Così facendo va però a mescolare livelli diversi (blub) e trasforma Internet in una rete

connection-oriented, con tutti gli svantaggi conseguenti: se il NAT box crasha, tutte le sue connessioni andranno perse.
Il NAT dispone inoltre di alcuni indirizzi riservati per le reti interne che non possono essere usati come normali indirizzi Internet.

IPv6

La **soluzione definitiva** al problema della scarsità di indirizzi IP risiede in una **nuova versione** di IP, cioè **IPv6**. In questa versione, gli **indirizzi** sono da **16 bytes** invece che da 4, garantendo così uno spazio di indirizzi infinitamente più ampio rispetto a prima. IPv6 **perde** però il campo **checksum**, diventando di fatto un protocollo **inaffidabile**: l'idea è che il controllo dell'errore, se serve, verrà fatto dai protocolli degli strati più alti.

ICMP

Internet Control Message Protocol, è il protocollo che serve a gestire **eventi inaspettati**; i messaggi ICMP vengono incapsulati dentro IP.

Message type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo and echo reply	Check if a machine is alive
Timestamp request/reply	Same as Echo, but with timestamp
Router advertisement/solicitation	Find a nearby router

DHCP

Se volessimo inviare un pacchetto IP, dovremmo tener conto che stiamo per attraversare strati diversi, che usano **spazi di indirizzi diversi** (es. IP e MAC). Per **passare da uno all'altro** si usa il protocollo **DHCP** (**Dynamic Host Configuration Protocol**). In particolare, quando una macchina vuole sapere il suo indirizzo IP, invia un pacchetto DHCP di tipo **DISCOVERY** e il gestore DHCP di rete risponde con l'indirizzo richiesto.

ARP

Nel caso invece in cui ci interessa passare **da indirizzo IP a indirizzo MAC**, la **soluzione** adottata è opposta rispetto a quella appena vista col DHCP; la corrispondenza si gestisce infatti in modo distribuito e **non più centralizzato**.
Se vuole inviare un messaggio ad una macchina con un certo indirizzo, un host trasmette attraverso un **pacchetto broadcast** in cui **chiede** chi è il **proprietario** dei

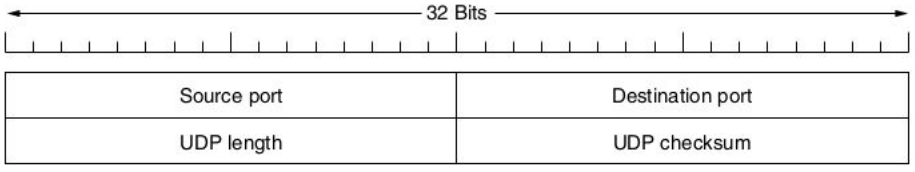
quell'indirizzo. La trasmissione broadcast arriva a tutte le macchine della rete e **ognuna controlla il proprio indirizzo IP**. Solo l'host con l'indirizzo richiesto risponde inviando il proprio indirizzo data-layer.
Il protocollo utilizzato per fare questa domanda e ottenere una risposta si chiama **ARP (Address Resolution Protocol)**. Quasi tutte le macchine di Internet lo utilizzano, in quanto utilizzare ARP al posto dei file di configurazione rende tutto più semplice. Il gestore del sistema non deve fare altro che assegnare a ogni macchina un indirizzo IP e decidere le maschere di sottorete, mentre ARP si occupa di tutto il resto.
Varie **ottimizzazioni** permettono ad ARP di funzionare in modo più efficiente. Tanto per cominciare, una volta eseguita l'operazione ARP, il **computer memorizza in cache il risultato**, casomai dovesse essere necessario ricontattare lo stesso computer, risparmiando il messaggio broadcast la volta successiva. Inoltre, in **ogni pacchetto ARP** che una macchina invia c'è in **piggyback** la **corrispondenza IP-data layer della macchina**; in questo modo, quando A chiede tramite ARP chi ha un certo indirizzo IP e B risponde, anche B saprà già la corrispondenza IP-data layer di A (così, nel caso in cui B debba nuovamente rispondere ad A non ci sarà bisogno di un'altra richiesta ARP).
Una conseguenza di questa politica è che, tipicamente, quando una **macchina entra in una rete**, invia un pacchetto ARP chiedendo il **suo stesso indirizzo IP**; così facendo, se nessuno risponde, tutti hanno una possibilità di **memorizzare** la **nuova corrispondenza** IP-MAC; se invece qualcuno risponde, significa che **due macchine nella rete hanno lo stesso indirizzo IP** e si è quindi verificato un errore di gestione.
Infine, bisogna considerare che se c'è nuova informazione su altre assegnazioni IP-MAC, questa ha priorità sulla vecchia; è così garantita la gestione dinamica della rete.

Strato di trasporto

La principale differenza di questo strato rispetto a quello di rete sta nel fatto che qui si fa **multiplexing delle singole risorse network** (cioè le macchine, identificate dai rispettivi indirizzi IP) tramite le porte. L'**associazione IP+porta** è chiamata **socket**, cioè la divisione in "rete logica" di una singola unità di rete.

UDP

L'applicazione più semplice di questo strato è lo **User Datagram Protocol**.



Come si può notare dal suo header, UDP aggiunge l'**informazione minimale** per lo strato transport (cioè la **porta** di chi **manda** e di chi **riceve**) e fa il controllo dell'errore tramite **checksum** (**opzionale**).
UDP è quindi un protocollo **connectionless** e si usa in tutte quelle situazioni in cui serve inviare **messaggi brev**i o dove non serve avere controllo dei dati ma si preferisce la **velocità**.

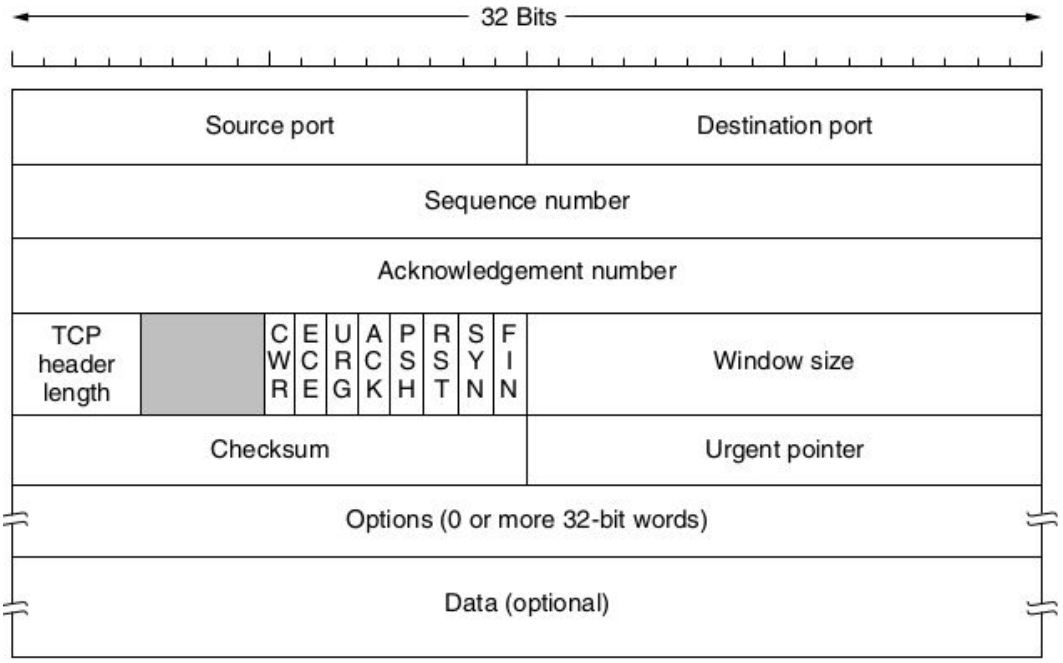
DNS

Un esempio d'uso di UDP è il **DNS (Domain Name System)**, cioè il protocollo che si occupa di **associare dei nomi agli indirizzi IP**. Questo protocollo funziona in modo **gerarchico**, utilizzando i resolver dei TLD (domini di primo livello) quando non c'è informazione disponibile. (livello applicazione).

TCP

Transmission Control Protocol, rappresenta il duale di UDP: mentre UDP è connectionless, TCP è **connection-oriented**, e mentre UDP è unreliable, TCP è **reliable**. Le connessioni TCP sono **full-duplex e point-to-point**.

L'header TCP



- **Source port**: porta di **origine**;
- **Destination port**: porta di **destinazione**;
- **Sequence number** e **Acknowledgement number**: entrambi da 32 bits, servono per il **controllo di flusso**;
- **TCP header length**: **lunghezza dell'header**, variabile a causa delle eventuali opzioni;
- **URG** e **PSH**: flag che indicano **pacchetti ad alta priorità**;
- **ACK**: flag che indica che c'è un **acknowledgement** (piggybacking);

- **RST**: flag che indica **problemi nella connessione** e fa il **reset** (o rifiuta nuove aperture);
- **SYN**: flag che serve ad **aprire una connessione** (con **3-way handshake**);
- **FIN**: flag che serve a **concludere una connessione** (tramite handshaking) solo da parte di chi lo invia (chi lo riceve può continuare a trasmettere dati);
- **Window size**: gestisce la **taglia delle sliding window** (TCP usa sliding windows ad ampiezza variabile; go back N di default con una opzione per il selective repeat);
- **Checksum**: **controllo dell'errore** (semplici somme);
- **Urgent pointer**: se il flag URG è attivo, qui è indicato il **numero d'ordine dell'ultimo byte urgente**;