



UNIVERSITÀ
DI PARMA

DIPARTIMENTO DI SCIENZE MATEMATICHE, FISICHE ED INFORMATICHE
Corso di Laurea in Informatica

Introduzione

RETI DI CALCOLATORI - a.a. 2023/2024
Roberto Alfieri

Contenuti

- ▶ Definizioni, topologie, canali di comunicazione e mezzi trasmissivi, gli standard
- ▶ Reti locali, reti geografiche, Internet
- ▶ La storia, Il modello ISO/OSI
- ▶ I livelli TCP/IP: livello rete, trasporto e applicazione
- ▶ Sicurezza
- ▶ Contenuti del corso

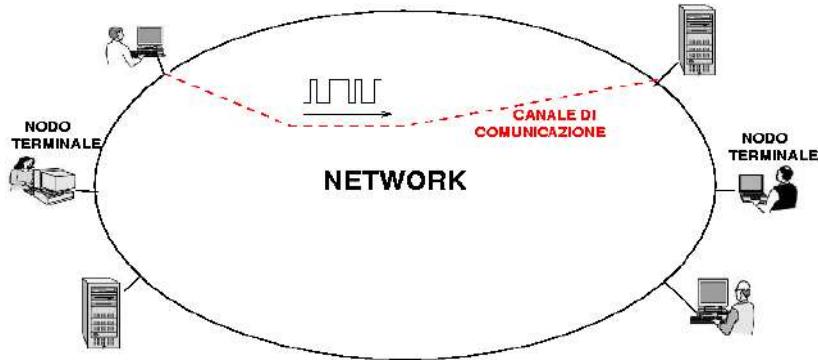
RIFERIMENTI

- ▶ *Reti di Calcolatori, A. Tanenbaum, ed. Pearson*
- ▶ *Reti di calcolatori e Internet, Forouzan , Ed. McGraw-Hill*

Cosa sono le Reti di Calcolatori

Una rete di calcolatori è un insieme di nodi di elaborazione

- autonomi tra loro
- connessi mediante un opportuno sistema di comunicazione
- in grado di interagire mediante scambio di messaggi al fine di consentire alle applicazioni in esecuzione sui nodi di comunicare tra loro



La comunicazione avviene attraverso canali individuati sulla rete e ha 5 componenti:

- **Messaggio:** informazione da trasferire
- **Mittente:** dispositivo che spedisce i messaggio
- **Destinatario:** dispositivo che riceve il messaggio
- **Mezzo di trasmissione:** cammino fisico percorso dal messaggio
- **Protocollo:** insieme di regole che governano la comunicazione

Applicazioni : una tassonomia

modello client-server (User to resource)

Servizi che consentono all'utente di accedere a risorse remote

- ▶ Accesso a risorse remote: sistemi, applicazioni, storage, stampanti, ..

modello peer to peer (user to user)

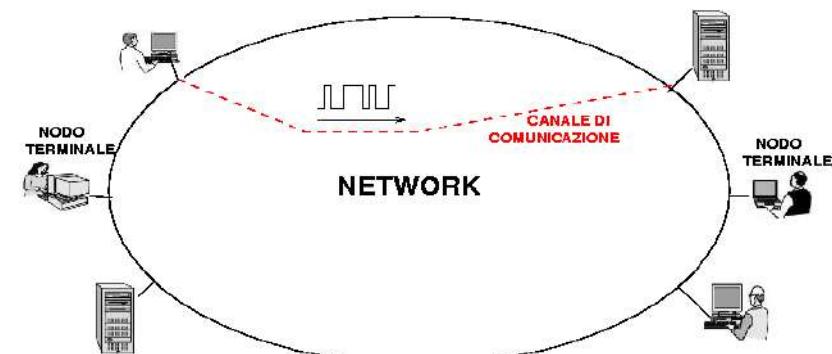
Servizi collaborativi per 2 o più utenti remoti

- ▶ File sharing, audio/video conferenza, VoIP

modello multi tier e architetture middleware (resource to resource)

Una applicazione viene distribuita su diversi host/device che cooperano via rete.

- ▶ Sistemi distribuiti (Modularità e affidabilità)
- ▶ Accelerazione delle prestazioni (HPC)
- ▶ Internet of Things



Metriche per la valutazione del canale di comunicazione

La qualità dei canali di comunicazione è stabilita dai seguenti parametri primari:

- **Aampiezza di Banda Digitale:** Quantità di dati (bit) che possono essere trasmessi nell'unità di tempo. Si misura in bit/sec
- **Ritardo (latenza) :** Tempo di trasferimento di un bit da un terminale all'altro (secondi)
- **Jitter :** Variazione del ritardo
- **Affidabilità:** Intolleranza agli errori di trasmissione (Percentuale di bit persi)
- **Sicurezza:** capacità di opporsi ad accesso non autorizzato, danno o violazioni della rete

Le applicazioni hanno sensibilità diverse rispetto ai parametri primari.

Ad esempio:

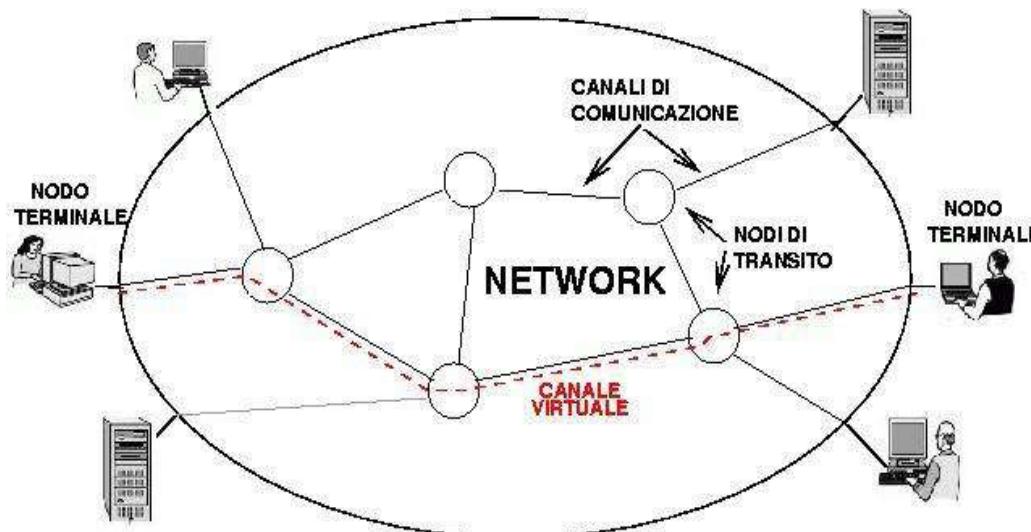
Application	Reliability	Delay	Jitter	Bandwidth
E-mail	High	Low	Low	Low
File transfer	High	Low	Low	Medium
Web access	High	Medium	Low	Medium
Remote login	High	Medium	Medium	Low
Audio on demand	Low	Low	High	Medium
Video on demand	Low	Low	High	High
Telephony	Low	High	High	Low
Videoconferencing	Low	High	High	High

Struttura fisica della Rete

La rete è un insieme di nodi interconnessi tra loro.

Due nodi comunicano mediante una **connessione fisica** se è presente un **canale fisico (link)** che li collega direttamente. Una catena di canali fisici forma una **connessione logica** su di un percorso (**path**) statico o dinamico detto **canale virtuale**.

In una connessione i nodi possono essere **terminali** (punto di arrivo di una connessione logica), di **transito** (nodi con funzionalità di **commutazione** dei dati tra un canale fisico e il successivo) o con **entrambe** le funzioni.

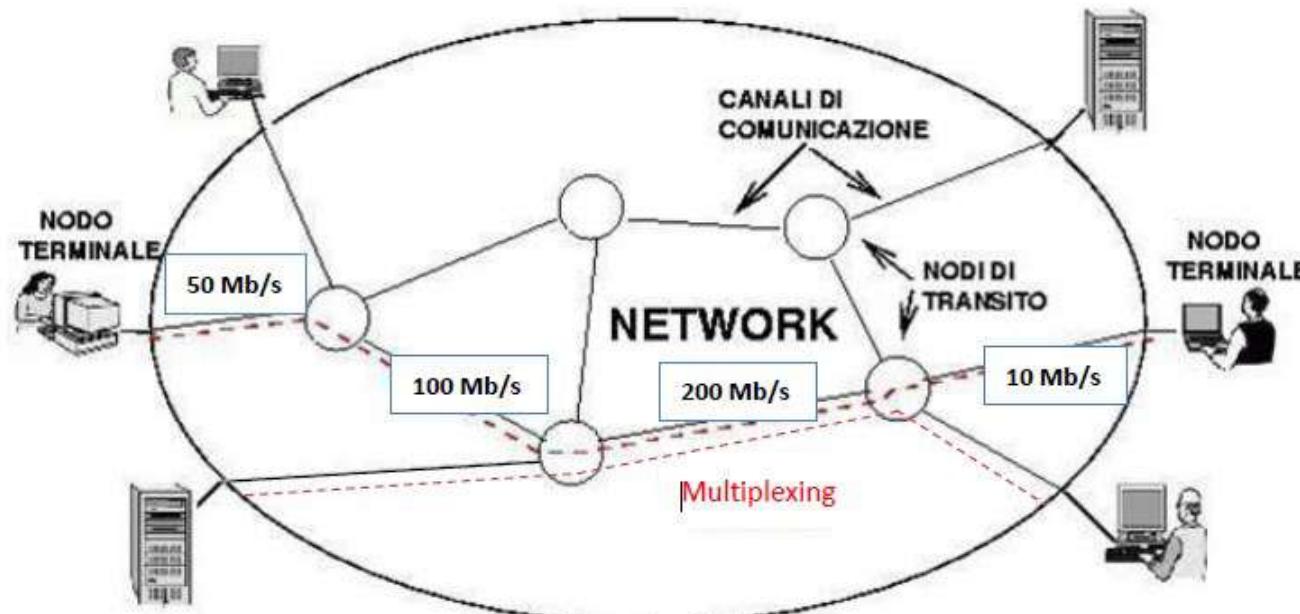


Aampiezza di banda e throughput

Ogni link ha la propria ampiezza di banda (dipende dal mezzo trasmittivo e dalla tecnica di codifica) e latenza (dipende principalmente dalla distanza).

L'effettiva veleità di trasmissione sul canale virtuale (throughput) non potrà superare l'ampiezza di banda del link più lento.

E' possibile attivare più canali virtuali contemporaneamente sullo stesso link utilizzando una tecnica di condivisione del canale (multiplexing). L'ampiezza di banda del link viene quindi condivisa tra i diversi canali virtuali.

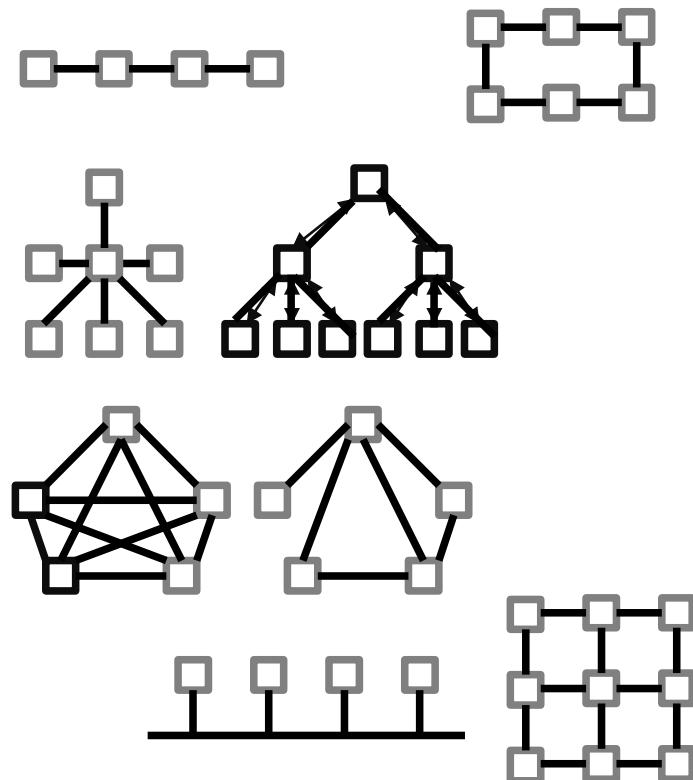


Topologia della rete

La topologia di rete è il modello geometrico finalizzato a rappresentare le relazioni di connettività, fisica o logica, tra i nodi della rete.

Principali topologie:

- Lineare aperta o ad anello
- A stella
- Ad albero
- Completamente o parzialmente magliata
- A bus
- A griglia



Topologia della rete : alcune definizioni

Link: è il canale di comunicazione che interconnette fisicamente due nodi.

Path: è la catena di link che compone un canale virtuale tra due nodi.

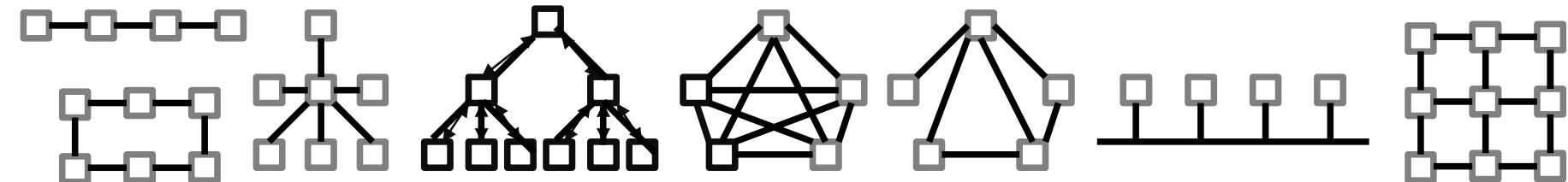
Nota: Per ogni coppia di nodi esistono uno o più path possibili. Attraverso opportune metriche possiamo attribuire un costo ad ogni path e determinare il path a costo minimo tra due nodi.

Hops: è il numero di link da attraversare.

Diametro: è la distanza (hops) tra i due nodi più lontani

Grado: numero massimo di link connessi ad un nodo

Scalabilità: Capacità della topologia di sostenere un numero crescente di nodi.



Modelli di utilizzo del canale

Le comunicazioni sui canali possono essere classificate in base a

Direzione:

- **Simplex** (monodirezionali, esempio TV)
- **Half-Duplex** (Entrambe le direzioni, non contemporaneamente, esempio walkie-talkie)
- **Full-Duplex** (Entrambe le direzioni contemporaneamente, esempio telefono)

Destinazione:

- **Unicast** (Un singolo destinatario)
- **Broadcast** (Tutti i nodi appartenenti ad una rete o sottorete)
- **Multicast** (Un sottoinsieme dei nodi della rete)

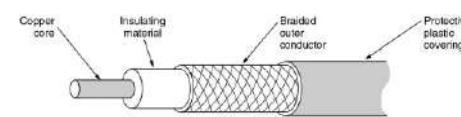
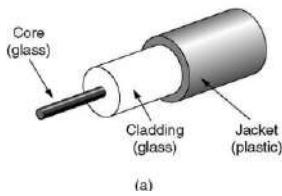
Canali di Comunicazione (link) e mezzi trasmissivi

I mezzi trasmissivi possono essere:

- **Punto-Punto:** comunicazione tra 2 nodi
- **Multi-Accesso :** comunicazione tra N nodi (Bus)



Ad esempio la fibra ottica e il doppino telefonico sono punto-punto, mentre il cavo coassiale e il wireless possono essere multi-accesso.

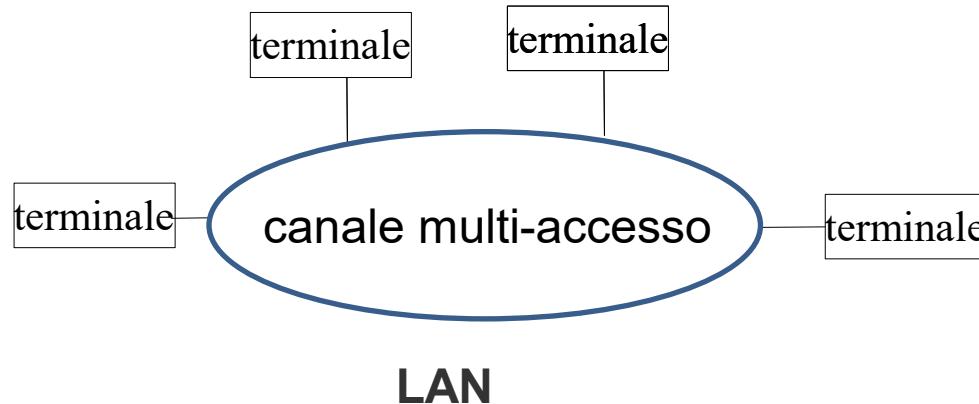


Local Area Network (LAN)

I mezzi trasmissivi multi-accesso possono essere utilizzati per realizzare un particolare tipo di rete detta LAN (Local Area Network) che include tutti i nodi che condividono lo stesso canale multi-accesso.

Le reti LAN supportano tutti i modelli di destinazione:

- Unicast, Broadcast e Multicast

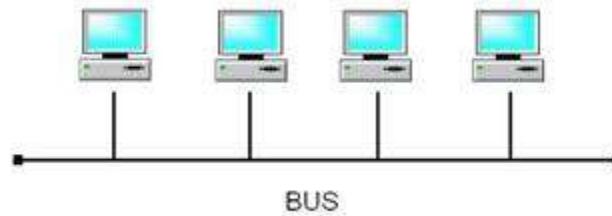


Le reti LAN dispongono di un opportuno **protocollo** per disciplinare l'accesso al canale e per gestire dell'indirizzamento (comunicazioni unicast, broadcast o multicast)

Reti locali: Ethernet

Ethernet è la tecnologia dominante per le reti LAN.

Nella versione Ethernet 2.0 (1980) l'ampiezza di banda era di 10Mb/s su cavo coassiale (topologia a BUS).



Nel 1995 esce lo standard denominato “Fast Ethernet” a 100Mb/s e nel 1999 arriva il Gigabit Ethernet a 1 Gb/s.

La topologia diventa a stella con l'introduzione di opportuni apparati di rete (HUB e switch), che consentono di realizzare una infrastruttura multi-accesso utilizzando mezzi trasmissivi punto-punto:

i dati ricevuti in una porta dell'apparato di rete vengono replicati su tutte e altre porte.
La topologia può anche essere ad albero connettendo più hub/switch in cascata.



Reti Geografiche (WAN)

Per realizzare reti che si estendono su grandi distanze geografiche (WAN - Wide Area Network) si possono utilizzare semplici link punto-punto

- ad esempio la connessione ADSL tra casa e la centrale telefonica



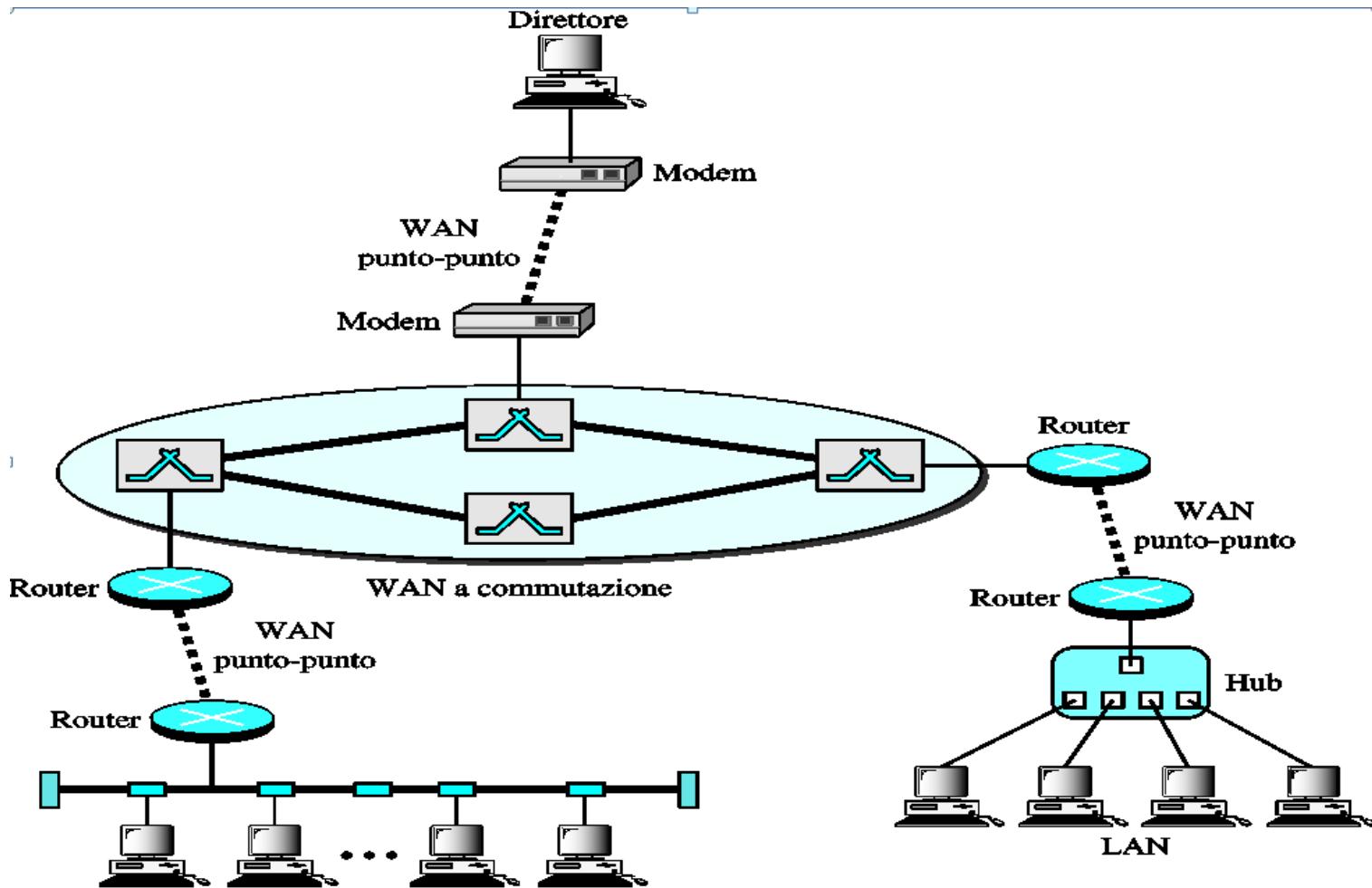
oppure infrastrutture di commutazione fornite da terze parti, come la rete ATM dei providers di telefonia.

- ad esempio la connessione tra 2 sedi remote di una azienda



Interconnessione di reti: internet

Dal punto di vista dell'infrastruttura fisica della rete internet (con la i minuscola) è l'unione di reti LAN e WAN interconnesse tra loro, con topologia magliata.



Commutazione di Circuito e di Pacchetto

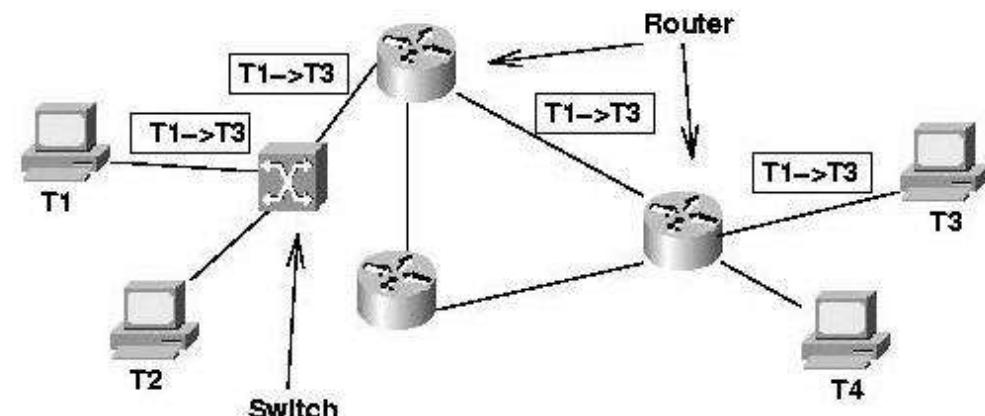
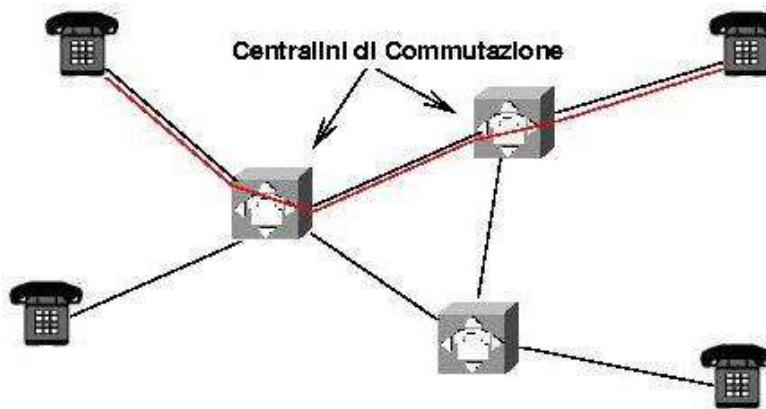
Il processo per individuare ed utilizzare il percorso su cui far transitare i messaggi su un canale virtuale è detto commutazione. Esistono due modelli di commutazione:

Commutazione di circuito (tipico delle reti telefoniche)

- Viene individuato il percorso tra i due terminali e creato un circuito fisico temporaneo.
- Esiste un ritardo iniziale dovuto al tempo necessario per instaurare il circuito.
- I terminali scambiano i dati (come se fosse un collegamento diretto).
- Il canale viene chiuso al termine della comunicazione.

Commutazione di Pacchetto (tipico delle reti dati)

- I dati della comunicazione sono frazionati in “pacchetti” con una lunghezza massima stabilita
- I nodi di transito (router, switch, ..) hanno il compito di instradare ogni pacchetto.
- Il destinatario riassembra i pacchetti e ricostruisce il messaggio.



Gli inizi: Le reti proprietarie

Negli anni 70 **IBM** dominava il mercato dei calcolatori con i propri “Mainframe” (prevalentemente della serie 370, con sistema operativo VM).

Nel 1974 rilascia per la prima volta un sistema operativo con un protocollo di rete denominato SNA (system Network Architecture).

DIGITAL Equipment Corporation (DEC) rilasciò nel 1978 un “mini” elaboratore che ebbe un grande successo: il VAX 11/780 (sistema operativo VAX/VMS).

Nel 1982 VAX/VMS incluse il protocollo di rete proprietario DECnet.

Negli anni 80 si diffusero a livello mondiale 2 reti indipendenti con tecnologie proprietarie: la rete BITnet basata su SNA degli elaboratori IBM e la rete HEPnet basata su DECnet. Le due reti non erano in grado di comunicare tra loro.

Standardizzazione delle reti

Esistono molti costruttori e fornitori di reti, ognuno con le proprie impostazioni.

La necessità dell'interoperabilità richiede la presenza di standard condivisi per ogni aspetto della rete.

Gli standard vengono proposti da diversi organismi internazionali. I principali sono:

ITU (International Telecommunication Union) ha il compito di standardizzare le telecomunicazioni, principalmente telefoniche. ITU-T è un settore di ITU che si occupa di telefonia e scambio dati.

ISO (International Standard Organization) definisce standard per una vasta gamma di argomenti. Sono stati emanati più di 17000 standard, incluso lo standard OSI (Open System Interconnect).

IEEE (Institute of Electrical and Electronics Engineers) è il principale ente professionale del mondo. Pubblica periodici, organizza conferenze e emana standard nel campo dell'ingegneria elettrica e dei computers. Il comitato IEEE 802 ha standardizzato molti tipi di LAN.

ISoc (Intenet Society) è nata nel 1992 su iniziativa di Vint Cerf and Bob Kahn per governare la crescita di internet. Gli standard sono emanati da **IETF (Intenet Engineer Task Force)** attraverso gli **RFC (Request For Comment)**.

W3C (World Wide Web Consortium) è un consorzio industriale guidato da Tim BernesLee (creatore del Web) che sviluppa protocolli e linee guida per la crescita a lungo termine del Web (HTTP, HTML, privacy, ecc.)

Il modello ISO/OSI

A partire dal 1976 la ISO (International Organization for Standardization) ha dato il via a lavori per giungere ad una serie di standard unificati per la realizzazione di reti di calcolatori aperte.

ISO ha proposto un modello di riferimento detto **Open System Interconnection Reference Model** (OSI-RM) che nel 1983 è diventato standard internazionale ISO 7498, detta comunemente ISO-OSI.

ISO-OSI è basato sul concetto centrale di una **architettura a strati**.

L'architettura a strati ha alcuni grandi vantaggi:

- scomponete il problema in sotto-problemi posti a diversi **livelli**, più semplici da trattare
- rende i vari livelli indipendenti, comunicanti mediante una interfaccia standard
- strati diversi possono essere sviluppati da enti diversi.

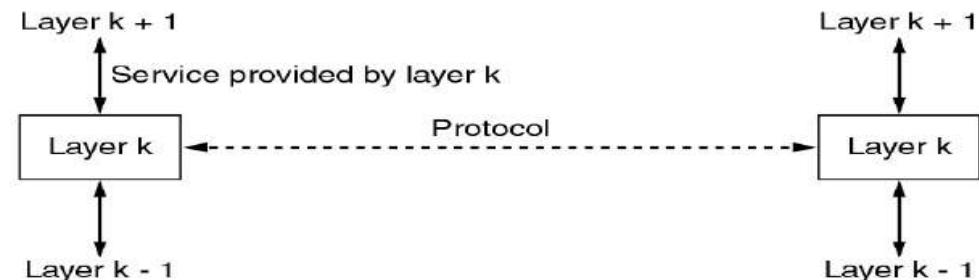
Strati ISO/OSI

OSI scomponete la comunicazione in **7 strati (o livelli)**. Lo scopo di ciascun strato è quello di fornire servizi agli strati superiori, mascherando come questi servizi sono implementati.

I nodi terminali devono avere tutti i livelli implementati, mentre i nodi di transito utilizzano solo i livelli più bassi, in base alla loro funzione.

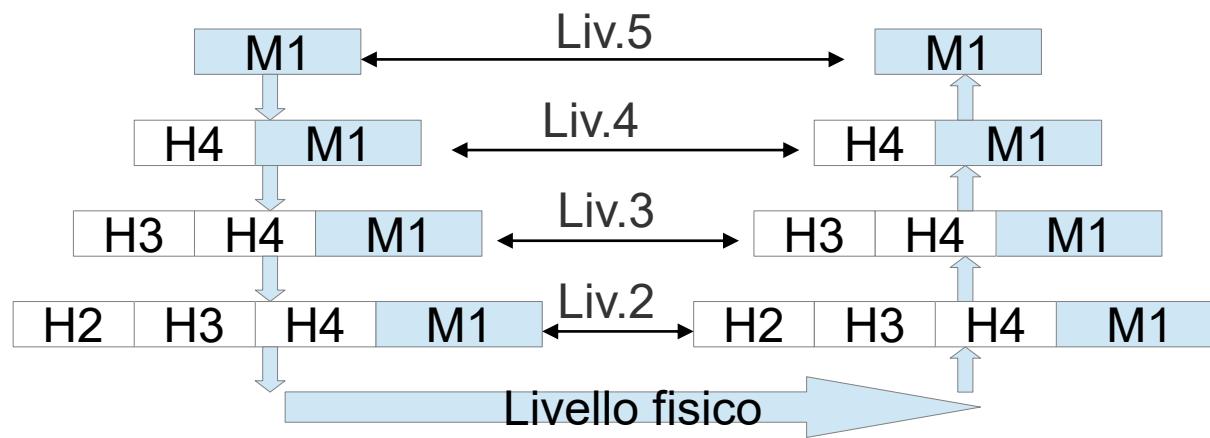


Ogni strato comunica logicamente con il pari strato del nodo remoto (peer), mentre fisicamente comunica con lo strato superiore ed inferiore dello stesso nodo.



Flusso dei dati

Il messaggio M viene frazionato in segmenti (M1, M2, ..) che vengono spedito in rete.
Ogni strato aggiunge una intestazione in cui inserisce propri dati di servizio:

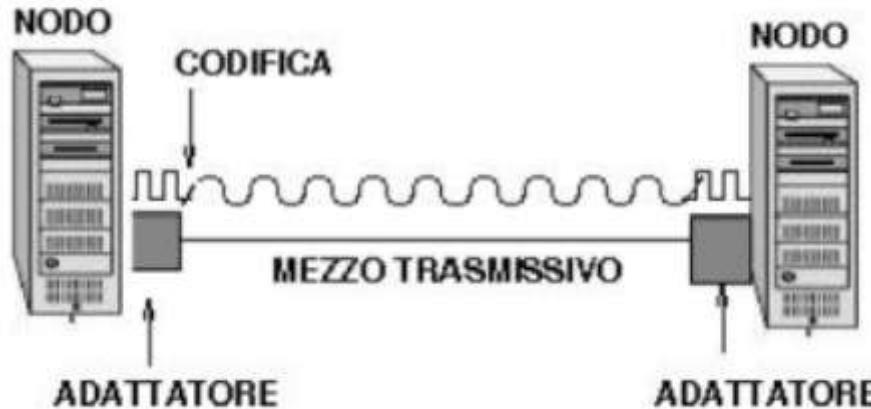


Strato 1 di OSI: Fisico

Scopo dello strato fisico è la trasmissione di bit “grezzi” sul canale di comunicazione. Deve attivare, mantenere e disattivare la connessione fisica per conto dello strato 2.

Per fare questo deve specificare le caratteristiche:

- meccaniche (forma di prese e spine, numero di contatti)
- elettriche (voltaggio e caratteristiche elettriche dei segnali associati all'interfaccia)
- funzionali (significato dei vari segnali, schemi di codifica dei bit in grandezze fisiche)



Strato 2 di OSI: Linea

Lo strato di linea deve

- attivare, mantenere e disattivare la connessione fisica su una linea (link) per conto dello strato 3
- rendere affidabile il collegamento diretto di un canale fisico

Le funzioni tipicamente svolte dallo strato 2 sono le seguenti:

- strutturazione del flusso di dati in unità di dialogo, denominati trame o frames
- controllo e gestione degli errori di trasmissione
- controllo di flusso (congestione)
- controllo di sequenza (ordine)

Per svolgere queste funzioni viene aggiunta una intestazione (Header) intesta ad ogni frame contenente informazioni di servizio.



Strato 3 di OSI: Rete

Scopo dello strato di rete è di far giungere le unità di informazione, dette pacchetti (**packets**), al destinatario determinando il percorso (**path**) attraverso la rete

Si occupa dunque del problema della commutazione. Nelle reti di calcolatori si usa la **commutazione di pacchetto** e la funzione svolta dallo strato 3 viene detta **routing**

Occorre un modo per individuare i destinatari: è necessario uno **schema di indirizzi**.

Se si vuole un'unica rete lo schema di indirizzi deve essere universale.

Strato 4 di OSI : Trasporto

Scopo dello strato di trasporto è **fornire un canale sicuro end-to-end (tra un processo mittente e un processo destinatario)**, svincolando gli strati superiori (applicativo) da tutti i problemi di rete

Tipiche funzioni

- **frammentare** i dati forniti dagli strati superiori (fragmenting/reassembling)
- Fornire una libreria di API per la programmazione

Può avere molte altre funzioni fra cui

- controllo dell'errore
- controllo di flusso
- gestione di dati prioritari
- ecc..

Non tutti le applicazioni hanno bisogno delle stesse funzioni, per cui si possono definire diverse **Classi di Trasporto**

Strato 5 di OSI: Sessione

Suddivide il dialogo fra le applicazioni in unità logiche (dette appunto **sessioni**)

Una sessione deve essere individuata, eventualmente interrotta e ripresa per fare fronte a vari eventi catastrofici: **perdita di dati, caduta della linea, momentaneo crash** di uno dei due interlocutori

Permette la chiusura ordinata (soft) del dialogo, con la garanzia che tutti i dati trasmessi sono arrivati a destinazione

Anche gli strati di sessione hanno molte funzionalità e possono essere più o meno completi a seconda delle richieste

Strato 6 di OSI: Presentazione

Adatta il **formato dei dati** usato dai due interlocutori preservandone il significato

La descrizione del tipo di dati usati per una applicazione e del loro formato si dice una sintassi

Ogni interlocutore ha una sua Sintassi locale per la rappresentazione dei dati e durante il dialogo bisogna concordare una Sintassi di trasferimento

E' stato definito un linguaggio detto Abstract Syntax Notation 1 (ASN 1) per descrivere e negoziare le sintassi

Strato 7 di OSI: Applicazione

Lo strato di Applicazione è l'utente della rete di calcolatori e pertanto non deve offrire servizi a nessuno

Rappresenta il programma applicativo (Applicazione) che per svolgere i suoi compiti ha bisogno di comunicare con altre applicazioni remote

Critica del modello OSI

Alla fine degli anni 80 sembrava che i protocolli OSI avrebbero dominato la scena mondiale. Questo non e' successo per 4 motivi:

Poca tempestività

- ▶ Prima c'e' la fase di ricerca, poi gli investimenti delle industrie.
- ▶ Le implementazioni OSI arrivano quando oramai TCP/IP era ampiamente diffuso.

Tecnologia scadente

- ▶ La scelta di 7 layer era più politica che tecnica.
- ▶ Due livelli quasi vuoti (presentation e session) e due troppo pieni (data link e network)

Implementazioni carenti

- ▶ Prime implementazioni lente, complicate ed enormi
- ▶ Al contrario TCP/IP era semplice veloce e Open (parte di Unix)

Incapacità politica

- ▶ Sviluppi dominati dalle Telecom, Ministeri, Comunità Europea e USA.
- ▶ Percepito con un insieme di standard imposti da burocrati.

ARPA.net e TCP/IP

1957: il Dipartimento della Difesa degli US crea l'Advanced Research Projects Agency (ARPA)

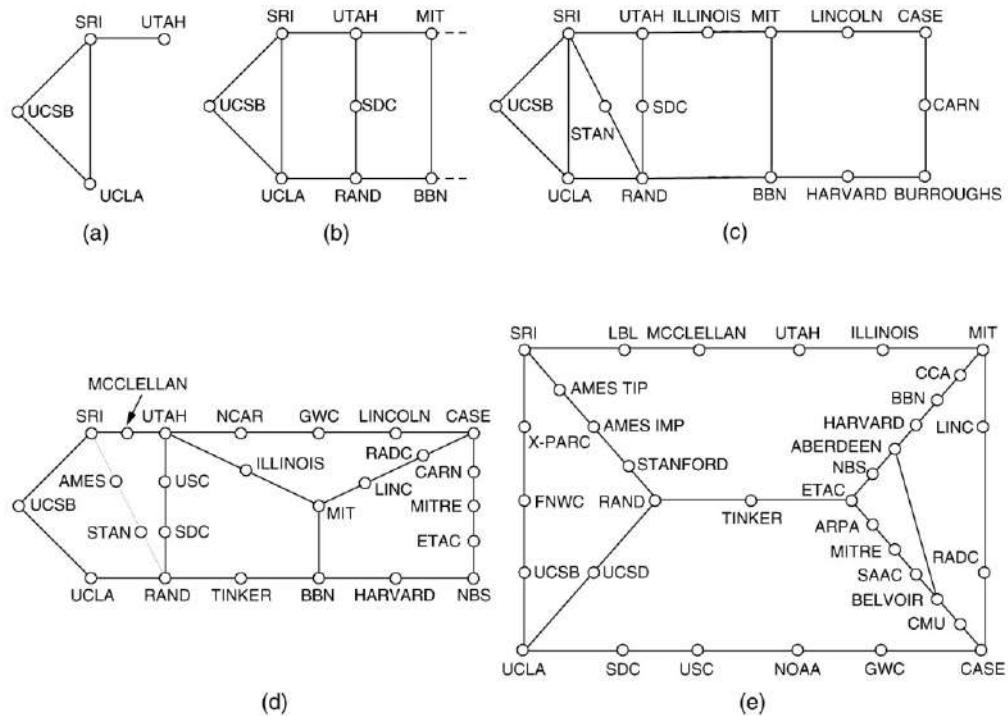
1970: nasce ARPAnet che collega alcune Università americane, utilizzando il protocollo host-to-host NCP (Network Control Protocol)

1974: Sulla base di un progetto finanziato da ARPA gli scienziati Vint Cerf e Bob Kahn pubblicano "A Protocol for Packet Network Intercommunication" in cui viene definita l'architettura del TCP/IP

1983: Il TCP/IP è adottato da ARPAnet

1986: La National Science Foundation (NSF) statunitense inizia lo sviluppo di NSFNET che oggi fornisce la maggior parte della spina dorsale dei servizi per Internet. Inizia la crescita esponenziale di Internet.

Crescita di Arpanet dal 1969 (a) al 1972 (e)



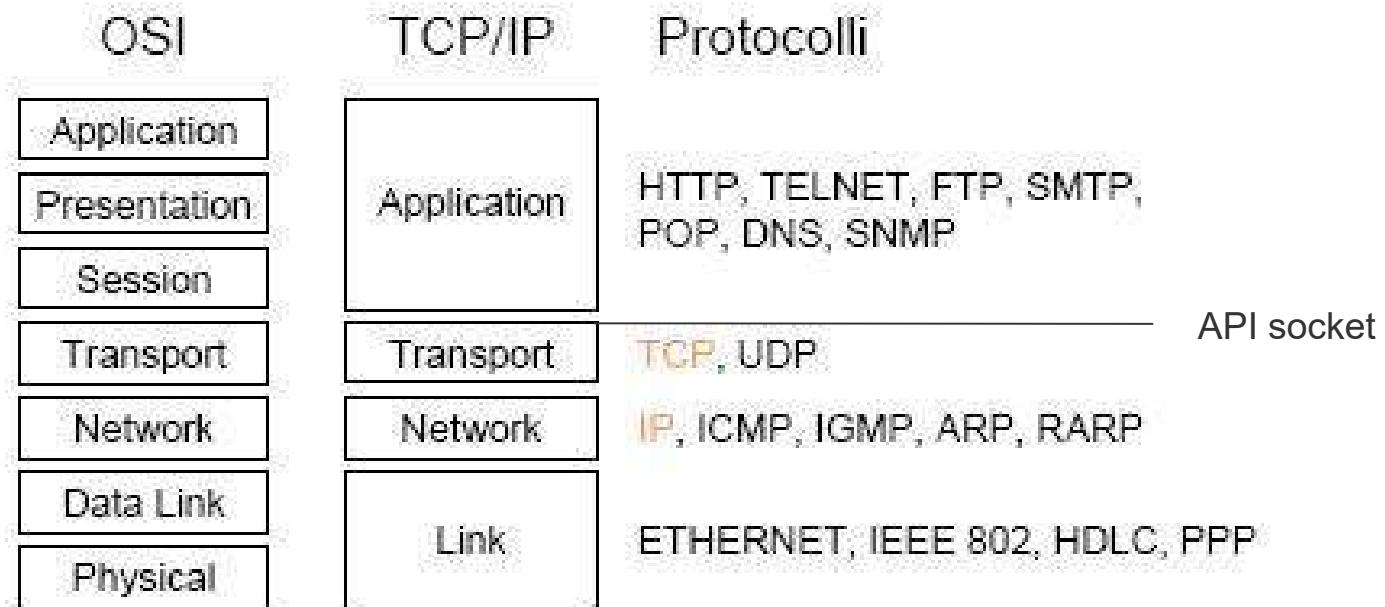
ISO/OSI e TCP/IP

Per alcuni anni si è ritenuto che TCP/IP fosse una soluzione transitoria e che prima o poi si sarebbe adottato OSI.

Il governo USA nominò una commissione per valutare la transizione e nel 1991 DECnet rilasciò una nuova versione (DECnet phase V) compatibile OSI, nella speranza di contrastare la crescita di TCP/IP.

Ma oramai le installazioni di TCP/IP erano così numerose da rendere troppo costosa la migrazione.

ISO/OSI rimane solo una architettura di riferimento.



Strati 1 e 2 di TCP/IP: fisico e collegamento

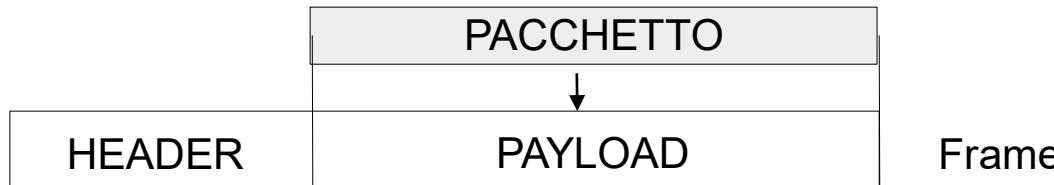
Per questi strati il modello TCP/IP non definisce nessun protocollo. Sono strati strettamente legati all'hardware di rete e vengono generalmente implementati nel device-driver della scheda e comunicano con i livelli superiori mediante una interfaccia standard.

Abbiamo 2 famiglie di protocolli:

- I protocolli WAN Esempi: HDLC e PPP.
- I protocolli LAN Esempio: Ethernet.

I protocolli dello stato fisico assegnano ad ogni interfaccia un indirizzo specifico, denominato **indirizzo fisico**. Ad esempio Ethernet utilizza indirizzi di 6 byte del tipo : 08 00 20 00 70 DF .

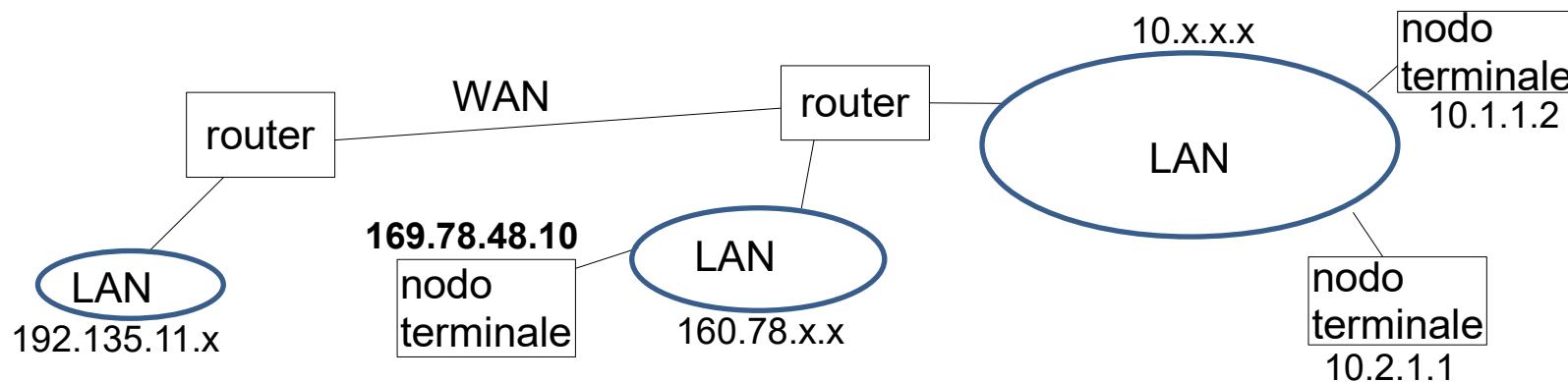
Tutti questi protocolli ricevono il **Payload** (carico utile) dal livello rete (pacchetto) che imbustano in una **Frame** in cui viene aggiunta una intestazione (**Header**) con campi necessari per il rilevamento degli errori, lo smistamento al livello superiore, l'indirizzamento.



Strato 3 di TCP/IP: rete

Lo strato rete ha il compito di mettere in comunicazione nodi appartenenti a reti LAN o WAN diverse. Richiede l'esistenza di particolari nodi di transito, detti **Router**, al confine tra 2 reti, con il compito di "ruotare" da una rete all'altra solamente i pacchetti che ne hanno necessità.

La suite di protocolli TCP/IP implementa questa funzionalità nel **protocollo IP**, che assegna ad ogni nodo un indirizzo logico univoco, come ad esempio 160.78.48.10. I bit della sequenza sono suddivisi in 2 parti: la prima parte denominata Network è comune a tutti i nodi dello stesso rete LAN (esempio 160.78) , mentre la seconda parte, denominata Host (esempio 48.10), distingue i nodi all'interno della LAN.



I router si coordinano tra loro tramite specifici protocolli (**link state** o **distance vector**) per individuare in modo dinamico il percorso migliore della comunicazione.

Lo strato 4 di TCP/IP: trasporto

Questo strato fornisce una connettività diretta tra 2 nodi indipendentemente dal percorso fisico di collegamento, svincolando gli strati superiori da tutti i problemi di rete.

Servizi Offerti:

Pacchettizzazione (fragmenting/reassembling): il flusso di dati da spedire viene frazionato in Segmenti che diventano il Payload del pacchetti di livello rete.

Multiplexing/demultiplexing: Ad ogni applicazione verrà associato un numero di porta univoco sul nodo, in modo da distinguere i pacchetti provenienti dallo stato inferiore e dirottarli sull'applicazione corretta.

Diversi tipi di servizio, in base alle necessità:

Connection-Less, fornito dal **protocollo UDP**. Senza garanzia di ricevimento, senza riscontro, utilizzato da applicazioni che devono scambiarsi rapidamente brevi messaggi.

Connection-Oriented, fornita dal **protocollo TCP** in cui viene attivato un canale virtuale tra le due parti. Il TCP fornisce la garanzia di consegna senza errori e il controllo del flusso.

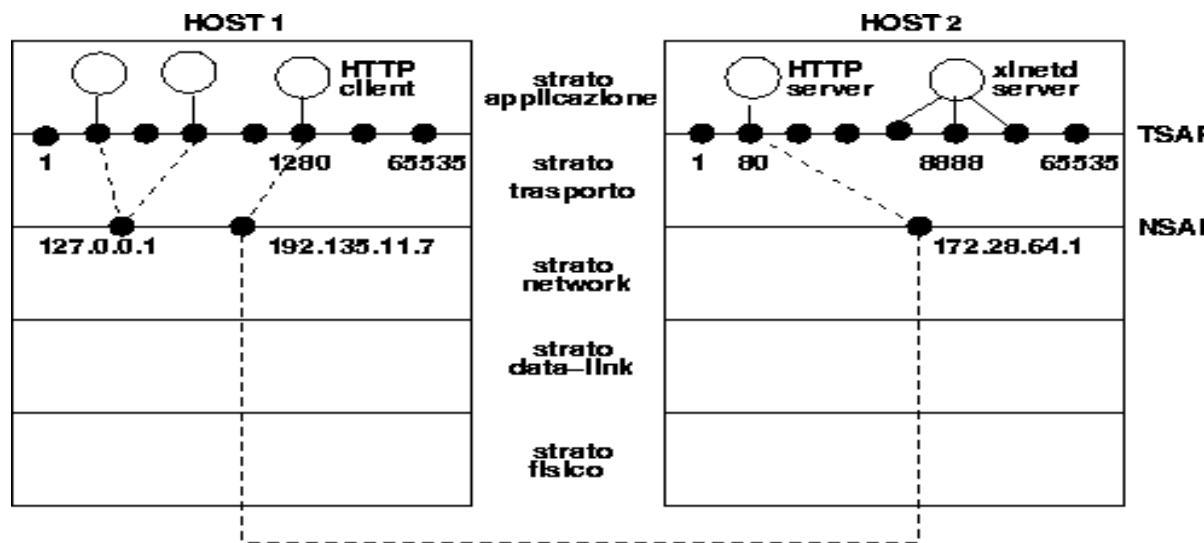
API: Le applicazioni del nodo potranno essere implementate per mezzo di una opportuna libreria di API, scegliendo il tipo di servizio più adatto alle proprie esigenze

Strato 4 di TCP/IP: demultiplexing

Visto che e' possibile avere piu' applicazioni di rete sullo stesso nodo, il livello di trasporto fornisce un meccanismo di multiplexing e demultiplexing basato sul concetto di porta.

Ad ogni applicazione verra' associato un numero di porta univoco sul nodo, in modo da distinguere i pacchetti provenienti dallo stato inferiore e dirottarli sull'applicazione corretta.

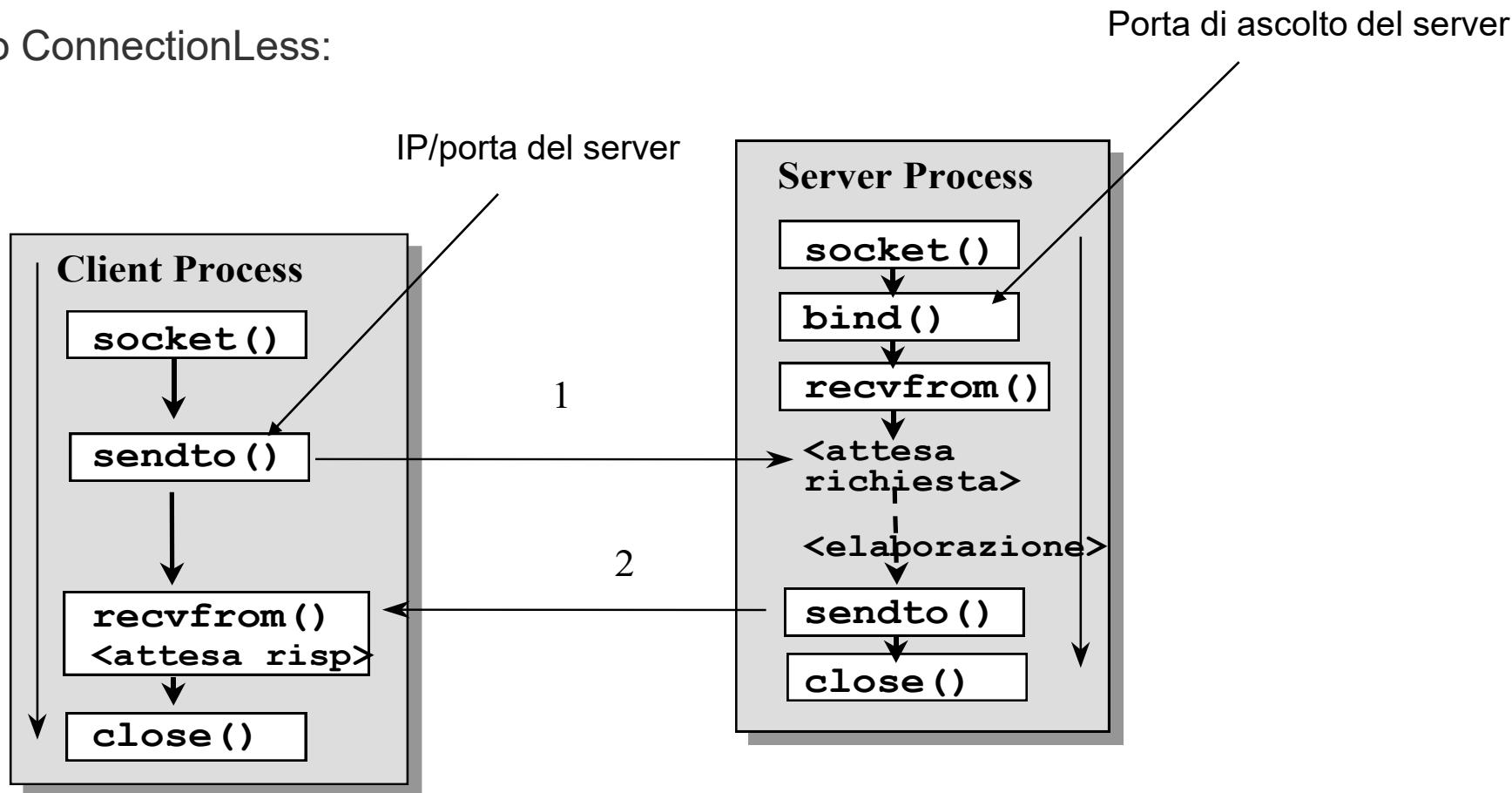
La porta e' un identificativo numerico (**indirizzo di porta**) che rappresenta il punto di arrivo di una connessione su di un host. La coppia (IPaddr, Port) identifica quindi univocamente un estremo di una connessione ed è detta **Socket**.



Strato 4 di TCP/IP : API

Attraverso le Berkeley Socket Library lo strato di trasporto di TCP/IP fornisce le API per implementare un modello di comunicazione **Client/Server** di tipo ConnectionLess o ConnectionOriented.

Esempio ConnectionLess:



Strato 5 di TCP/IP: Applicativo

Il modello client-server richiede che sia ben identificabile e indirizzabile il server. Le porte TCP o UDP assegnate ad applicativi con ampia diffusione vengono assegnate da un organismo internazionale (IANA) e prendono il nome di [Well Known Ports](#). Contrariamente le porte dei client vengono determinate dinamicamente dal sistema operativo al momento dell'accesso al server.

Esistono alcuni applicativi “storici” di TCP/IP quali:

telnet (23/TCP) per il servizio di terminale remoto a linea di comando
smtp (25/TCP) per il servizio di posta elettronica
ftp (20/TCP e 21/TCP) per il trasferimento di file

Altri importanti applicativi si sono aggiunti successivamente, tra cui:

dns (53/UDP) per la traduzione dei nomi
http (80/TCP) per il trasferimento di ipermedia (WWW).

Ogni applicativo assegna ai propri oggetti un **indirizzo specifico** mnemonico.

Esempio di questi indirizzi sono www.unipr.it per il servizio Web e alfieri@unipr.it per il servizio di posta elettronica.

Sicurezza delle reti

L'utilizzo pervasivo delle reti anche in ambiti critici come l'economia, la finanza e la salute ha portato ad una attenzione crescente sulla sicurezza nell'utilizzo delle reti e ad una conseguente revisione delle architetture di rete a tutti i livelli.

A livello architetturale viene definito un perimetro delle risorse da difendere (Rete locale o singolo PC) che viene protetto con strumenti di controllo e limitazioni (riduzione della superficie d'attacco) rispetto a possibili attacchi informatici provenienti dall'esterno, introducendo strumenti come Firewall, Intrusion Detection/Prevention Systems e, più recentemente, strumenti di Intelligence con la CyberSecurity.

A livello dei protocolli un ruolo fondamentale è dato dall'introduzione di **strumenti crittografici** per rafforzare servizi di sicurezza come l'autenticazione, la riservatezza e il non ripudio. L'utilizzo della crittografia sta portando ad una graduale sostituzione dei protocolli di rete sia a livello applicativo (HTTPS, IMAPS, LDAPS, ecc) che ai livelli sottostanti (SSL, IPsec, ecc).

Contenuti del corso

Lezioni:

- **Il livello fisico** e la telefonia.
- **Il livello link** e Ethernet
- **Il livello rete**, IPv4, protocolli di servizio, IPv6, protocolli di routing.
- **Il livello trasporto**, UDP, TCP, programmazione di rete, gestione della congestione
- **Il livello applicativo**, DNS, Posta elettronica e WWW.
- **Sicurezza delle reti**, crittografia applicata, la sicurezza nei protocolli.

Laboratorio:

- Net-tools e Net-sniffing
- Net-programming
- WWW server e client
- Networking con Docker
- Crittografia applicata con OpenSSL
- Virtualhost SSL, programmazione SSL, password cracking



UNIVERSITÀ
DI PARMA

DIPARTIMENTO DI SCIENZE MATEMATICHE, FISICHE ED INFORMATICHE
Corso di Laurea in Informatica

Il Livello Fisico

RETI DI CALCOLATORI - a.a. 2023/2024
Roberto Alfieri

Il livello Fisico: sommario

PARTE I

- ▶ Scopo dello strato Fisico
- ▶ Il canale di comunicazione
- ▶ I mezzi Trasmissivi
 - Trasmissioni su rame, il Doppino, il cablaggio strutturato
 - Trasmissioni su Fibra Ottica
 - Trasmissioni via Etere
- ▶ Le codifiche dei bit: banda base e banda passante

PARTE II

- ▶ Il sistema Telefonico, FDM, TDM, DSL, la telefonia mobile

RIFERIMENTI

- ▶ *Reti di Calcolatori, A. Tanenbaum, ed. Pearson*
- ▶ *Reti di calcolatori e Internet, Forouzan , Ed. McGraw-Hill*

Scopo del livello Fisico

Il livello fisico si occupa del trasferimento dei bit tra nodi che si affacciano allo stesso canale fisico (punto-punto o multi-accesso).



Il trasferimento avviene utilizzando un **mezzo trasmissivo** su cui i bit vengono **codificati** trasformandoli in una forma di energia (tipicamente segnali elettromagnetici come luce, tensione, onde e.m.). I nodi sono dotati di un **adattatore** che ha il compito di codificare i bit in energia e decodificare i bit in arrivo.



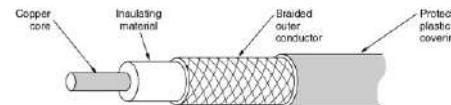
Mezzi trasmissivi

Esistono 3 categorie principali di mezzi fisici per la realizzazione di un canale:

1) Elettrico

Cavi coassiali di rame (MultiAccesso)

Doppini telefonici (Punto-Punto)



2) Ottico

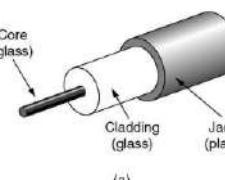
Fibre ottiche (canali Punto-Punto)

3) Wireless

Onde radio omnidirezionali (MultiAccesso)

Ponti radio (Punto-Punto)

Satelliti (punto-punto o MultiAccesso)



La scelta del mezzo dipende dalle caratteristiche del canale quali:

- l'attenuazione
- la sensibilità ai disturbi esterni
- la velocità di trasmissione
- il ritardo di propagazione (latenza)
- il costo e la maneggevolezza nell'impiego.

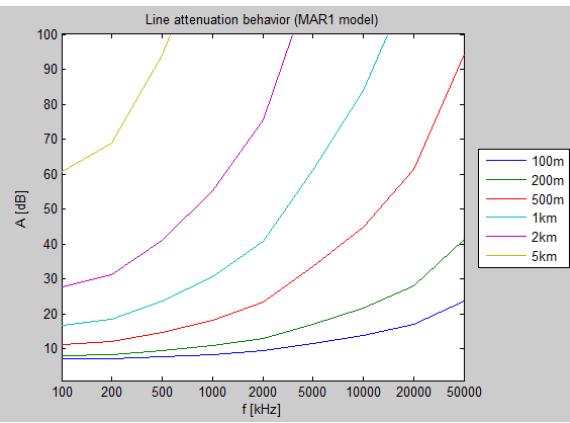
Banda passante

Per codificare un segnale sul mezzo trasmittivo si utilizza un intervallo di frequenze contenute in una banda che ha una determinata larghezza H , espressa in Hertz. Generalmente l'attenuazione dipende dalle frequenze utilizzate dal segnale inviato.

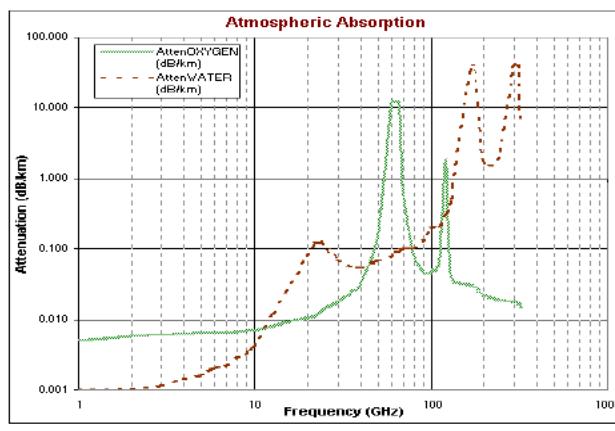
Per ogni mezzo trasmittivo è necessario individuare una banda di frequenze, detta **banda passante**, che verrà utilizzata per la codifica dei dati, in cui l'attenuazione è più contenuta e possibilmente costante.

La **banda passante** è determinata dalle caratteristiche fisiche del mezzo, ma può essere limitata in modo artificioso mediante opportuni "filtri passa banda"

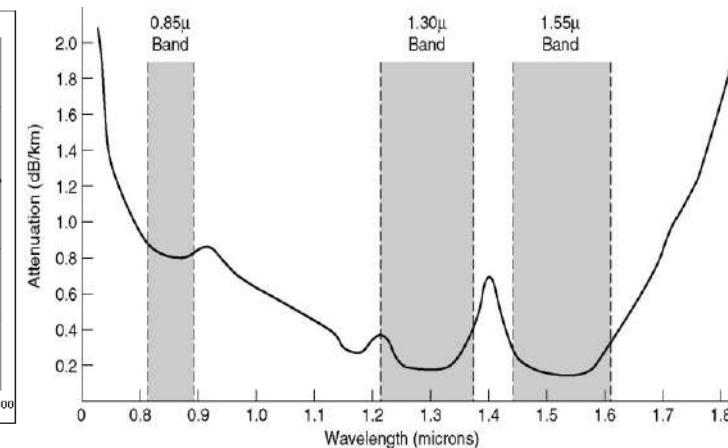
rame



etero



fibra ottica



Nota: Le onde vengono misurate in base alla loro frequenza f (Hz) o lunghezza λ (metri).

Le 2 grandezze sono legate dalla velocità di propagazione (v): $v = \lambda \cdot f$

Al esempio per gli infrarossi usati nelle fibre ottiche: $\lambda=1\mu m$ $f=v/\lambda = 2 \cdot 10^8 m/s / 1 \mu m = 200 THz$

Deterioramento del segnale Attenuazione

Diminuzione del segnale sulla lunghezza del mezzo

Determina la massima distanza raggiungibile

Dovuto ad una perdita di energia

Si misura in Decibel (db) = $10 \log_{10} P2/P1$ (P1=trasmittente, P2=ricevente)

E.g: un **dimezzamento** della potenza del segnale corrisponde a

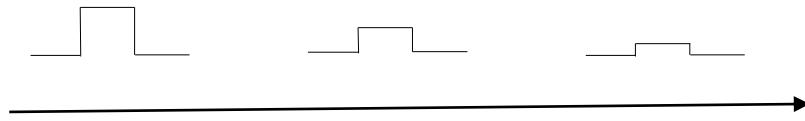
$$10 \log_{10} P2/P1 = 10 \log_{10} 0.5 = -3\text{db}$$

E.g: Un **raddoppio** della potenza, dovuto ad amplificazione, corrisponde a

$$10 \log_{10} P2/P1 = 10 \log_{10} 2 = 3\text{db}$$

- Nei mezzi guidati è lineare con la distanza (esempio fibra ottica -0,24 db/Km)

- Nei mezzi omnidirezionali (etero) si aggiunge l'attenuazione isotropica



Deterioramento del segnale Distorsione

Se la banda di frequenze utilizzata non ha valori costanti di attenuazione si aggiunge il fenomeno della distorsione del segnale, dovuto alla maggiore attenuazione di alcune frequenze (tipicamente le frequenze più alte).



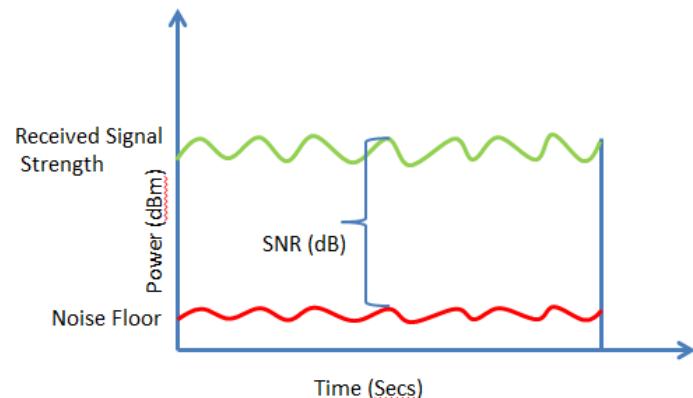
Deterioramento del segnale rumore e disturbo

Rumore: al Segnale (con potenza S) si sovrappone Rumore termico (con potenza N), sempre presente, dovuto al movimento delle molecole del mezzo.

Il rapporto segnale/rumore (Signal Noise Ratio - SNR) si misura in db ($10 \log_{10} S/N$)

Esempio ADSL:

- ▶ < 5db : la linea non si sincronizza ($S/N < 3$)
- ▶ 5-10 db : linea scadente
- ▶ 10-15 db : linea mediocre
- ▶ 15-22 db : linea buona
- ▶ 23-38 db : linea ottima
- ▶ 29-35 db: linea eccellente



Disturbo: proveniente da elementi esterni

- ▶ Diafonia (crosstalk): quando il disturbo proviene da canali adiacenti

Velocità massima di trasmissione di un canale

La **Banda Passante** incide sulla velocità massima con cui possiamo spedire bit sul canale (**ampiezza di banda digitale B**, espressa in bit/sec)

La relazione tra Banda Passante analogica **H** e l'ampiezza di banda digitale **B** su di un canale ideale (privo di rumore) è stabilita da **Nyquist**:

$$B = 2H \log_2 V \text{ b/s}$$

V è il numero di simboli del segnale. Ad esempio, se usiamo la fibra ottica con una sorgente luminosa possiamo inviare due simboli: luce oppure buio (V=2)

Questa legge teorica ci porterebbe ad aumentare il numero di simboli per ottenere velocità virtualmente illimitate.

In presenza di rumore abbiamo però un tasso massimo fissato dal rapporto Segnale/Rumore e definito dal teorema di **Shannon**:

$$B = H \log_2 (1+S/N) \text{ indipendente dal numero di simboli del segnale}$$

E.g: in un canale con $H=3\text{KHz}$ e un rapporto S/N di 30db (cioè 1000) abbiamo

$$B = 3K \log_2 (1001) = 30 \text{ Kb/s}$$

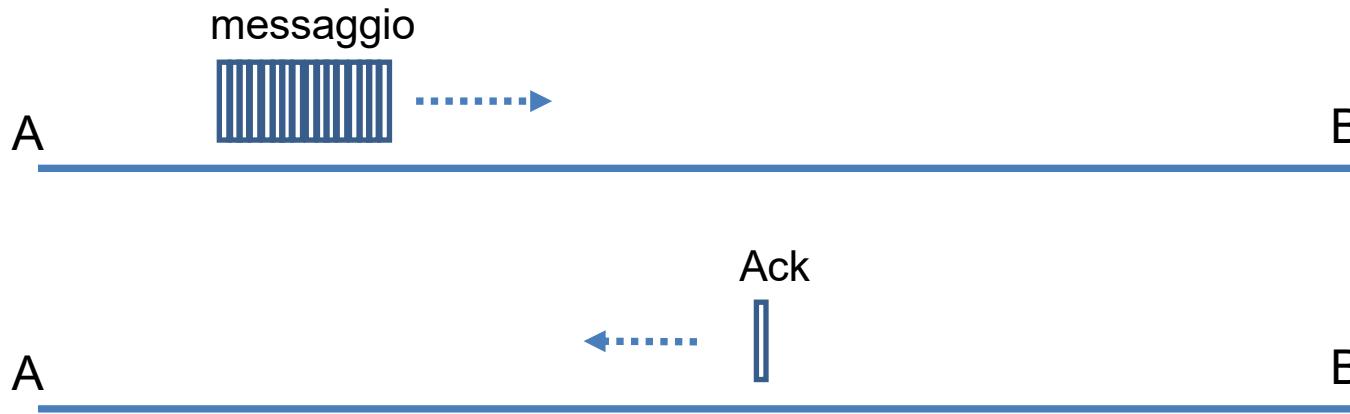
Uso di entrambi i teoremi: applichiamo Shannon per stabilire l'ampiezza di banda digitale **B**, quindi Nyquist per determinare il numero di simboli ottimale $V = 2^{(BitRate/2H)}$

Esempio precedente: $B = 2H \log_2 (V) = 30K \text{ bps}$ $V = 2^{(30K/6K)} = 2^5 = 32$

Tempo di consegna e RTT

Il **tempo di consegna** è il tempo necessario per trasferire un **messaggio** (sequenza di bit) dal mittente al destinatario.

Il **Round Trip Time (RTT)** è il tempo tra l'invio di un messaggio e la ricezione di un riscontro (ACK).



Questi tempi sono determinati dalla somma di diverse latenze introdotte dal mittente, da eventuali nodi di transito, dal mezzo trasmisivo e dal destinatario.

Tempi della comunicazione

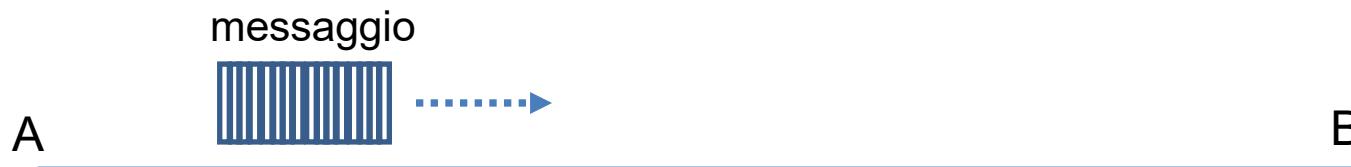
I principali componenti che incidono su questi tempi sono:

- ▶ **Tempo di trasmissione:** tempo che impiega un messaggio ad uscire dall'interfaccia di rete. Dipende dal numero di bit e dalla velocità di trasmissione (b/s)
 $t = \text{numero di bit} / \text{velocità di trasmissione}$

- ▶ **Tempo di propagazione:** tempo che impiega un bit a percorrere il mezzo
 $t = l / v$ dove $l \rightarrow$ lunghezza del mezzo , $v \rightarrow$ velocità di propagazione nel mezzo
 $v = c/n$ $c \rightarrow$ velocità della luce = 300000 Km/s (3×10^8 m/s)
 n è l'indice di rifrazione del mezzo es: aria $n = 1,0003$, acqua $n=1,33$, Fibra Ottica $n=1,5$

La velocità di propagazione nell'**aria** è circa 3×10^8 m/s

La velocità di propagazione nei mezzi guidati (**fibra ottica e rame**) è circa 2×10^8 m/s (valore approssimato da utilizzare negli esercizi)



Tempi della comunicazione

Tempo di preparazione del mittente: tempo necessario al mittente per la preparazione del dato da spedire (ad esempio tempi di codifica e compressione).

- ▶ **Tempo di riempimento del pacchetto (real time streaming):** se abbiamo un flusso continuo di dati (esempio applicazioni multimediali real-time) i primi bit inseriti in un pacchetto devono attendere il completamento del pacchetto prima di essere inviati.

Tempo di attraversamento dei nodi di transito

- ▶ **Tempo di elaborazione** (o inoltro) è introdotto dal nodo di transito ed è dovuto al processamento software e/o hardware dei dati. Dipende dalle caratteristiche del nodo.
- ▶ **Tempo di attesa.** Se un nodo di transito utilizza delle code di trasmissione si introduce un tempo di attesa necessario per lo smaltimento della coda.
 - ▶ Il tempo dipende dal carico della rete ed è trascurabile se la rete è scarica.

Tempo di elaborazione del destinatario (esempio tempi per la decodifica e decompressione)

Misuriamo il bit sul mezzo trasmissivo

Lunghezza e tempo di trasmissione del bit (Ethernet su rame)

1 bit:

$$\text{A } 10\text{Mb/s} \quad t = 1/10\text{M} = 10^{-7} \text{ s} = 100\text{ns} \quad l = 2 \times 10^8 \times 10^{-7} = 20 \text{ m}$$

$$\text{A } 100\text{Mb/s} \quad t = 1/100\text{M} = 10^{-8} \text{ s} = 10\text{ns} \quad l = 2 \times 10^8 \times 10^{-8} = 2 \text{ m}$$

$$\text{A } 1\text{Gb/s} \quad t = 1/1000\text{M} = 10^{-9} \text{ s} = 1\text{ns} \quad l = 2 \times 10^8 \times 10^{-9} = 20 \text{ cm}$$



1KByte (8000 bit) :

$$\text{A } 10\text{Mb/s} \quad t = 800\mu\text{s} \quad l = 20 \text{ m} \times 8000 = 16\text{Km}$$

$$\text{A } 1\text{Gb/s} \quad t = 8\mu\text{s} \quad l = 20 \text{ cm} \times 8000 = 160\text{m}$$

Tempo di propagazione:

$$\text{Cavo in rame di una rete locale (100m)} \quad t = 100 \text{ m} / 2 \cdot 10^8 \text{ m/s} = 0.5 \mu\text{s}$$

$$\text{Fibra Ottica Roma - NewYork (6.600Km)} \quad t = 6,6 \cdot 10^6 \text{ m} / 2 \cdot 10^8 \text{ m/s} = 33 \text{ ms}$$

$$\text{Satellite geostazionario (h=35.800Km x 2)} \quad t = 71,6 \cdot 10^6 \text{ m} / 3 \cdot 10^8 \text{ m/s} = 238 \text{ ms}$$



Diagramma spazio-tempo

Il diagramma spazio-tempo ci consente di rappresentare graficamente i tempi coinvolti nella spedizione.

Esempio di trasmissione di un messaggio da 100 Byte a 1 Gb/s su cavo di rame di 100m
Consideriamo solo i tempi di trasmissione e di propagazione, e trascuriamo il tempo di trasmissione del riscontro (ACK).

Tempo di Trasmissione:

$$t_{\text{trasm}} = 800 \text{b} / 1 \text{Gb/s} = 0.8 \mu\text{s}$$

Tempo di Propagazione:

$$t_{\text{prop}} = 100 / 2 \times 10^8 \text{s} = 0.5 \mu\text{s}$$

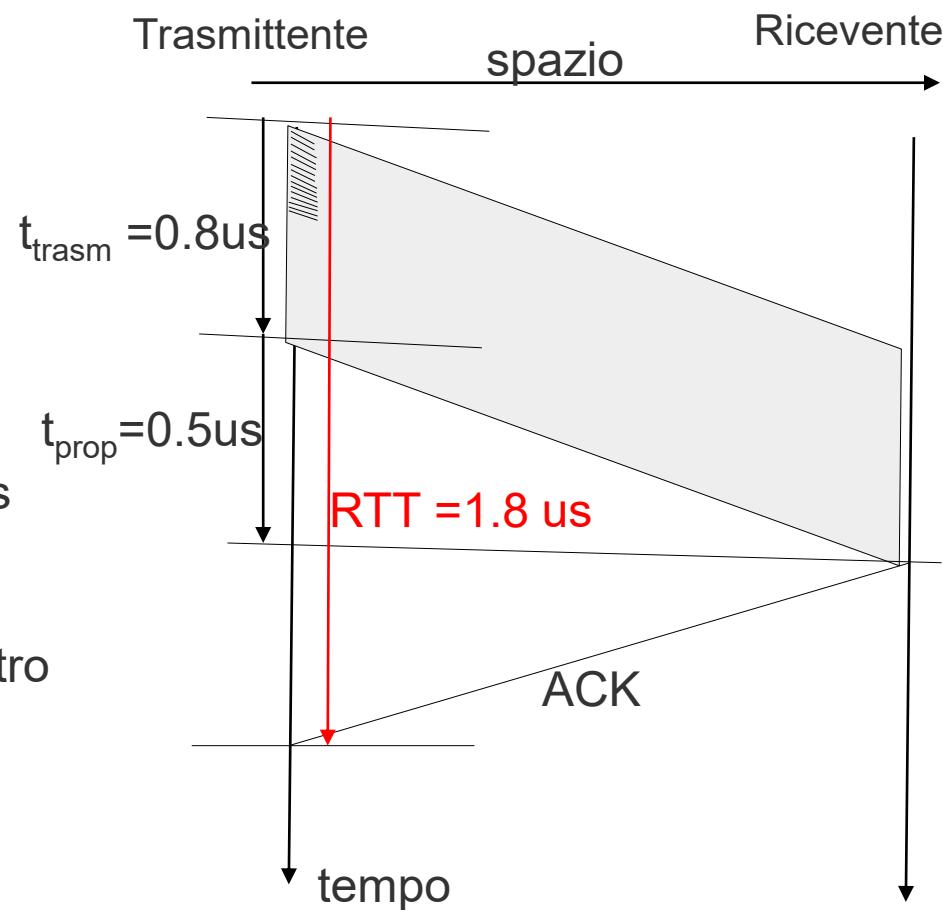
Tempo di consegna:

$$t_{\text{consegna}} = t_{\text{trasm}} + t_{\text{prop}} = 0.8 + 0.5 \mu\text{s} = 1.3 \mu\text{s}$$

Round-Trip Time (RTT):

E' il tempo di consegna + tempo del riscontro

$$\text{RTT} = t_{\text{consegna}} + t_{\text{prop}} = 1.8 \mu\text{s}$$

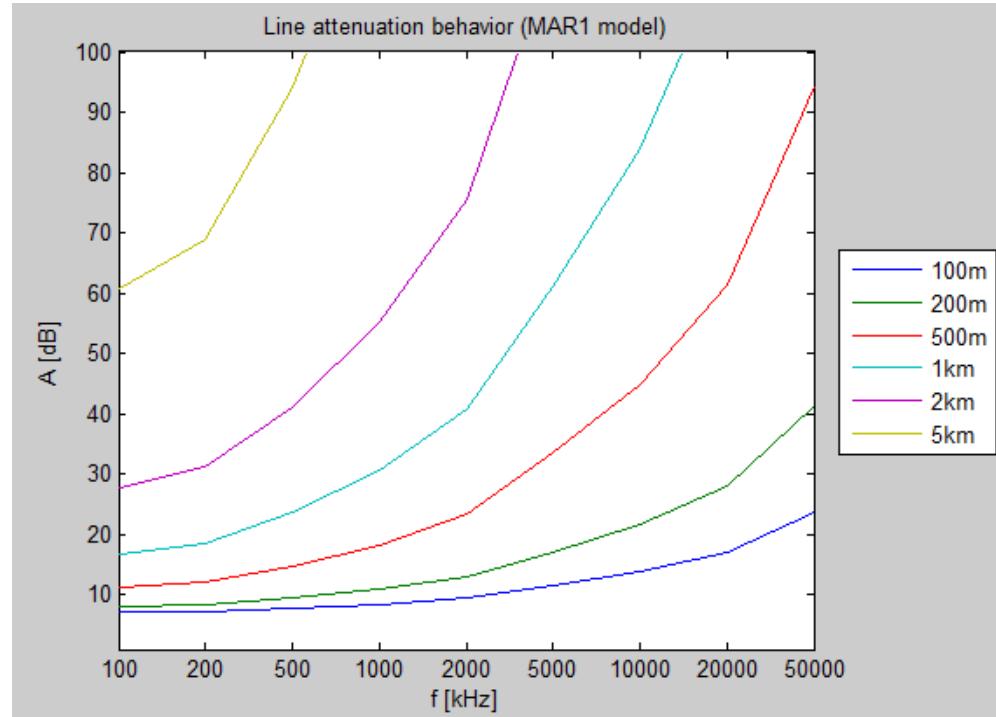


Cavo in rame

E' un mezzo trasmissivo a basso costo, ma l'attenuazione del segnale cresce rapidamente con la frequenza e con la distanza. Per questo motivo è largamente utilizzato nelle reti locali.

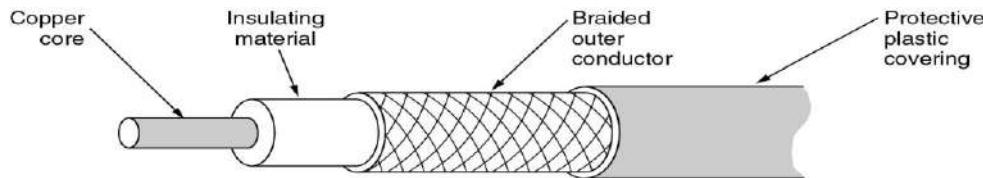
Esistono 2 principali tipi di cavi:

- ▶ Cavo coassiale
- ▶ Doppino



Cavo Coassiale (Coaxial Cable)

Due cavi di rame concentrici e separati da un materiale isolante.
La parte esterna è realizzata con una calza di conduttori sottili.

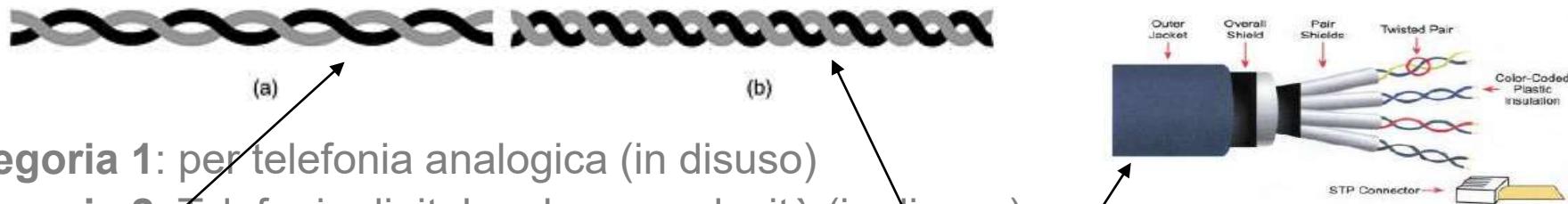


Il cavo coassiale denominato RG58 veniva utilizzato negli anni 80-90 come canale Multi-accesso e Bidirezionale (half duplex) per le reti locali.
Attualmente poco utilizzato nell'ambito delle reti di calcolatori.

Doppino (Twisted Pair)

Coppia di fili di rame avvolti non schermati (UTP - Unshielded Twisted Pair).

Viene realizzato con diversi standard qualitativi a seconda del tipo di utilizzo:



Categoria 1: per telefonia analogica (in disuso)

Categoria 2: Telefonia digitale a bassa velocità (in disuso)

Categoria 3: Banda 16MHz. Utilizzato per Telefonia digitale (attualmente in uso)

Categoria 4: Banda 20MHz (scarsamente utilizzato)

Categoria 5: Banda 100MHz

Categoria 6: Banda a 250MHz

Categoria 6a: Banda a 500MHz

attualmente in uso per le reti locali con cavi composti da 4 coppie e connettori RJ45

Categoria 7: Banda a 600MHz - STP - (Shielded Twisted Pair), 4 coppie schermate singolarmente

Attenuazione: Max 100-200 metri

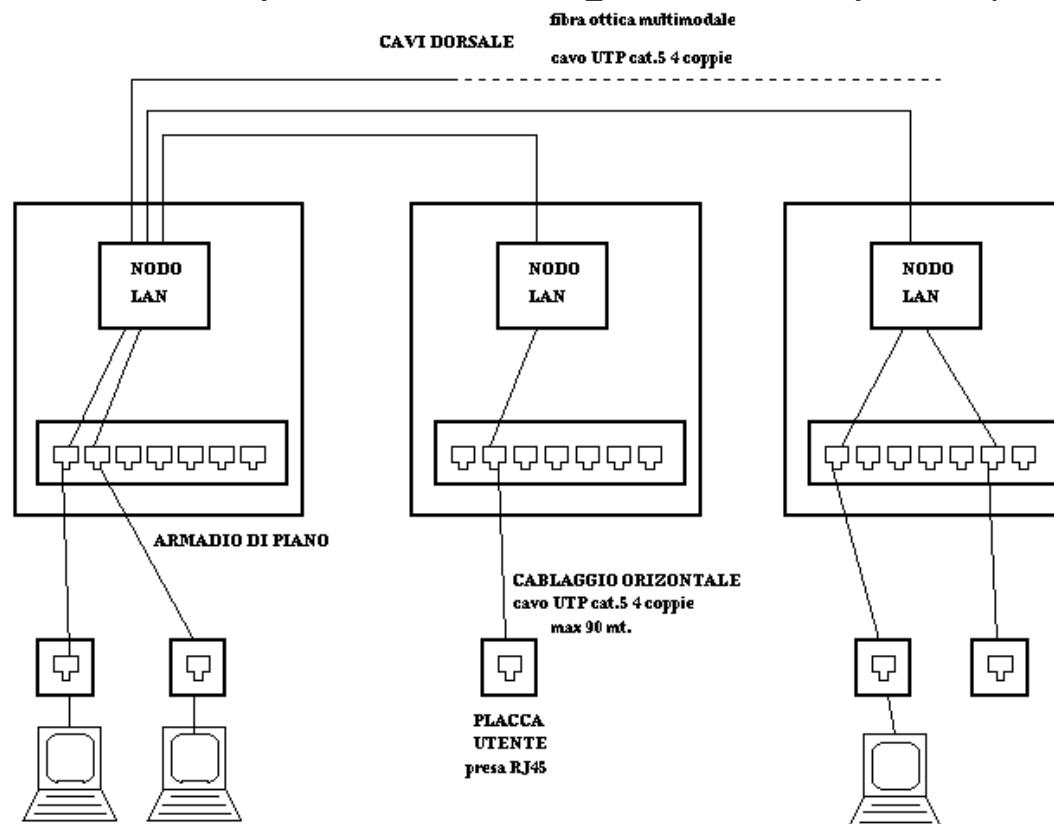
Caratteristiche: Banda passante fino a 500MHz → Max 10Gb/s (10GbaseT)

Cablaggio strutturato

La distribuzione capillare in edificio del cablaggio per la telefonia e per i dati fa oramai parte dell'infrastruttura di un edificio.

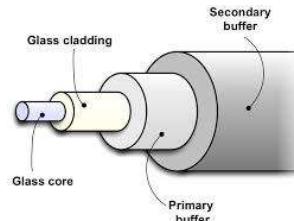
Il cablaggio strutturato vuole essere una soluzione architettonale flessibile per fonia e dati: entrambi i sistemi possono avere una topologia ad albero con cablaggio terminale in doppino telefonico almeno di **categoria 5** (cablaggio orizzontale).

Si usa fibra ottica multimodale per connettere gli armadi di piano (cablaggio verticale)



Fibra Ottica

Fibre di vetro che trasportano impulsi di luce su fibre flessibili del diametro di qualche decina di micron (1 micron = 10^{-6} m)



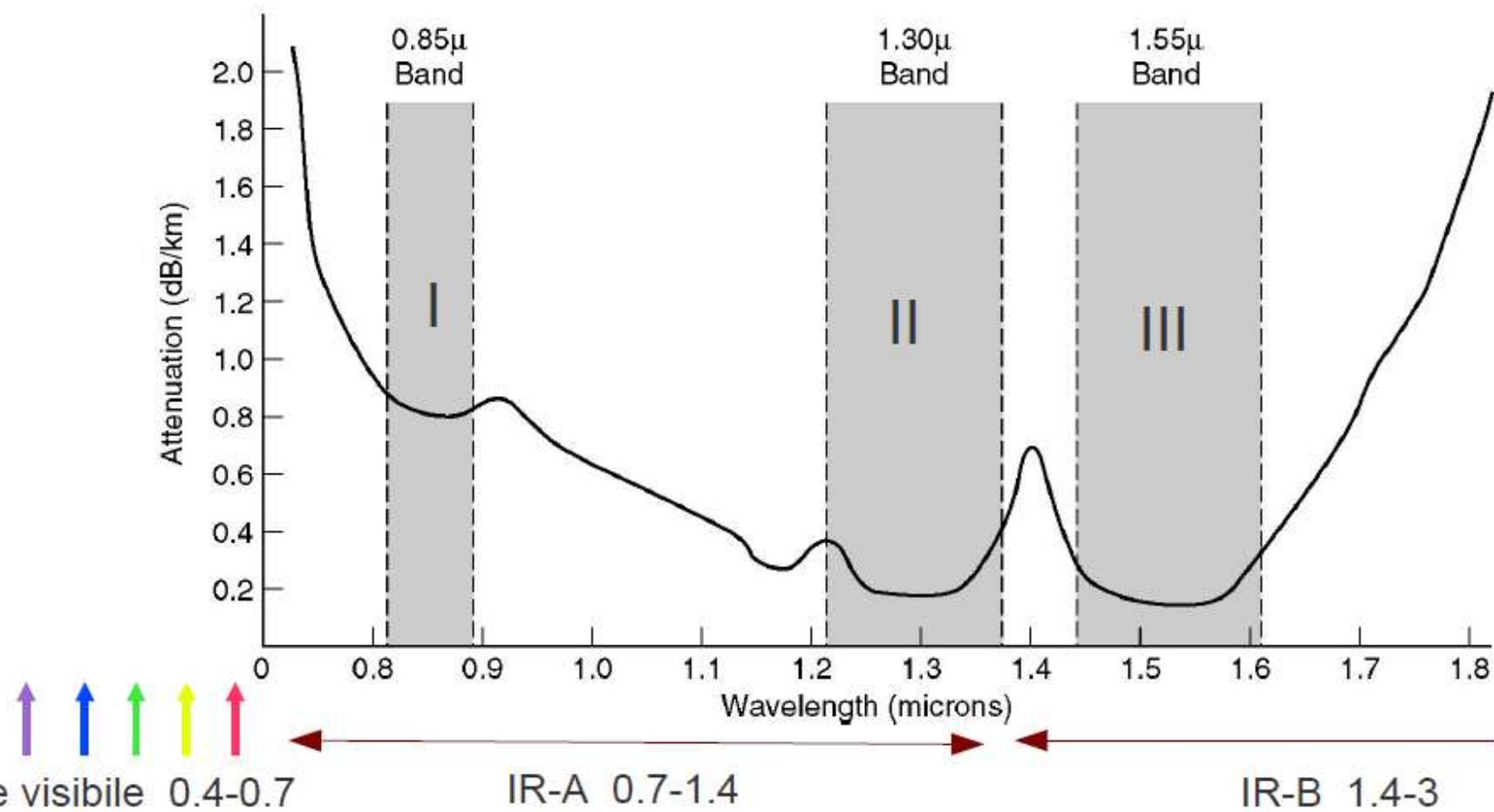
Caratteristiche:

- ▶ 3 componenti: sorgente luminosa, mezzo di trasmissione, rilevatore.
- ▶ Modulazione
 - OOK (On Off Keying) : un impulso di luce indica il valore 1, l'assenza indica il valore 0.
 - SCM (SubCarrier Modulation): modulazione di una portante (AM, FM, ..)
- ▶ Ottimo rapporto Segnale/Rumore (60 – 65 dB).
- ▶ Elevata banda trasmisiva (25000-30000 GHz, fino a 50 Tbps !)
- ▶ Bassa attenuazione
 - 1 - 3 dB/Km per F.O. multimodali
 - 0.4 - 0.2 dB/Km per F.O. monomodali (ovvero meno del 5% per Km)
- ▶ E' immune da disturbi e.m.
- ▶ Sicurezza (difficile inserirsi in una comunicazione)
- ▶ Minor dimensione e peso rispetto al rame
- ▶ Maggior costo di installazione, connettorizzazione e dei dispositivi attivi.

Trasmissione Ottica

Bande (I II e III) comprese tra 250THz e 300 THz (InfraRosso, 0.8-1.6 us)

Ricerche in corso per contenere la dispersione cromatica (attenuazione cambia con la lunghezza d'onda)

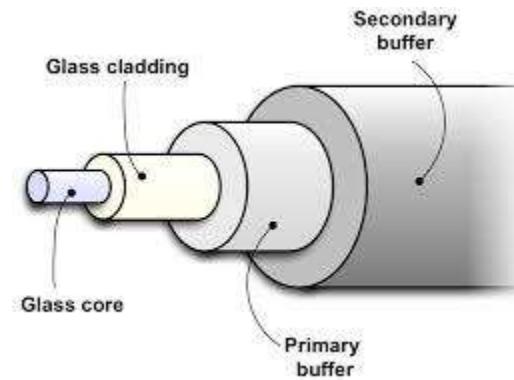


Fibre Monomodali e Multimodali

Esistono 2 tipi di fibre:

Fibre Multimodali

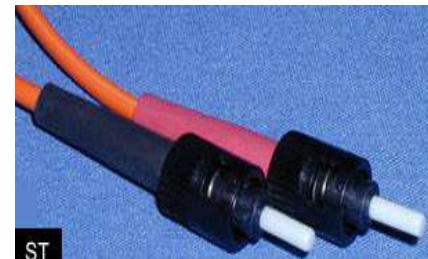
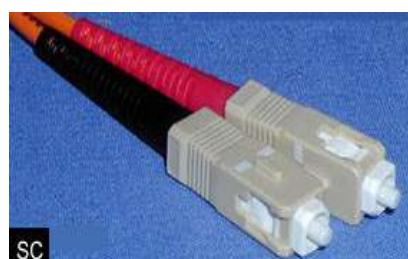
- I raggi colpiscono le pareti con diversi angoli (mode)
- 50/125 (core 50 micron/cladding 125 micron) e 62.5/125
- Luce generata con LED
- Finestra di utilizzo I e II



Fibre Monomodali

- Percorso rettilineo , senza rimbalzi
- 10/125 (core 10 micron/cladding 125 micron)
- Luce generata con un fascio Laser
- Finestra di utilizzo II e III
- Maggiore costo, Minore attenuazione (possono trasmettere dati a 50Gbps per 100Km).

I connettori piu' utilizzati sono LC, SC e ST:



International Undersea Fiber Systems

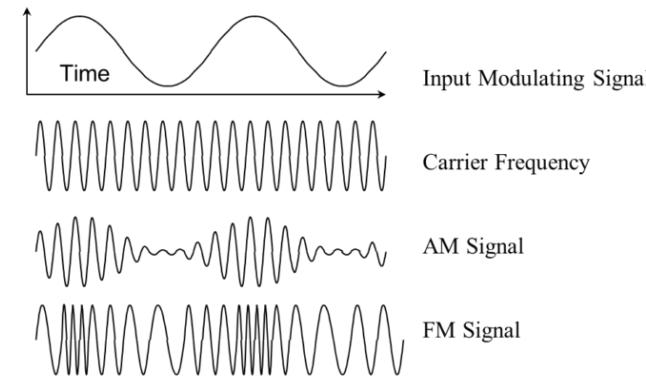
Alcatel-Lucent 

Optical fibre submarine network

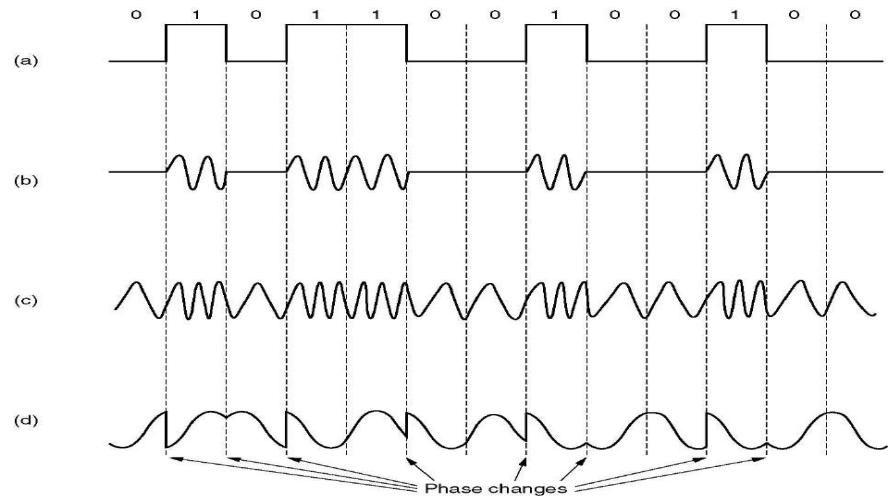


Onde Elettromagnetiche

Le onde e.m. possono essere utilizzate per trasmettere informazioni senza l'utilizzo di un mezzo fisico guidato. La trasmissione dei dati avviene modulando una frequenza portante (carrier), come avviene per le trasmissioni radio AM o FM (Modulazione di Ampiezza o di Frequenza)



Per trasmissione dati (a) viene modulata l'ampiezza (b), la frequenza (c) o la fase delle onde (d) di una portante, ma codificando valori discreti.



The Electromagnetic Spectrum

Onde Radio (10^4 - 10^9 Hz) : sono facili da generare, possono viaggiare per lunghe distanze e attraversano facilmente gli edifici. Il limite è la ridotta ampiezza di banda. Le Onde Radio (LF, MF e HF) seguono il terreno, VHF e UHF viaggiano in linea retta e rimbalzano contro gli ostacoli.

Microonde (10^9 - 10^{11} Hz - $\lambda \div 30\text{cm}-1\text{mm}$) Viaggiano in linea retta e faticano ad attraversare gli edifici.

Infrarossi (10^{11} - $4 \cdot 10^{14}$ Hz - $\lambda \div 1\mu\text{m}$) : Non attraversa gli ostacoli solidi. Sono utilizzati per periferiche a breve distanza (telecomandi) e per gli impulsi nelle fibre ottiche (near infrared).

Luce visibile ($4 \cdot 10^{14}$ - $8 \cdot 10^{14}$ Hz $\lambda \div 0.5\mu\text{m}$)

La luce Ultravioletta, i raggi X e i raggi Gamma ($> 8 \cdot 10^{14}$ Hz): funzionerebbero anche meglio, ma sono difficili da generare, da modulare, non si propagano bene attraverso i muri e sono dannose.

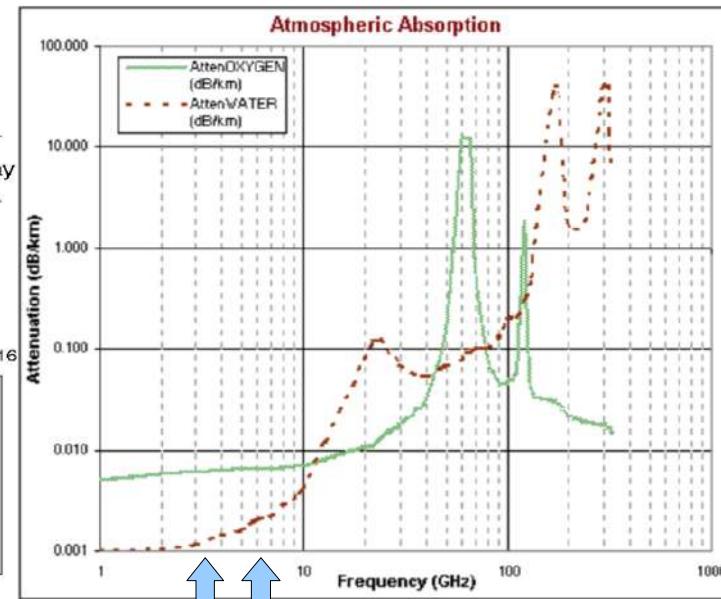
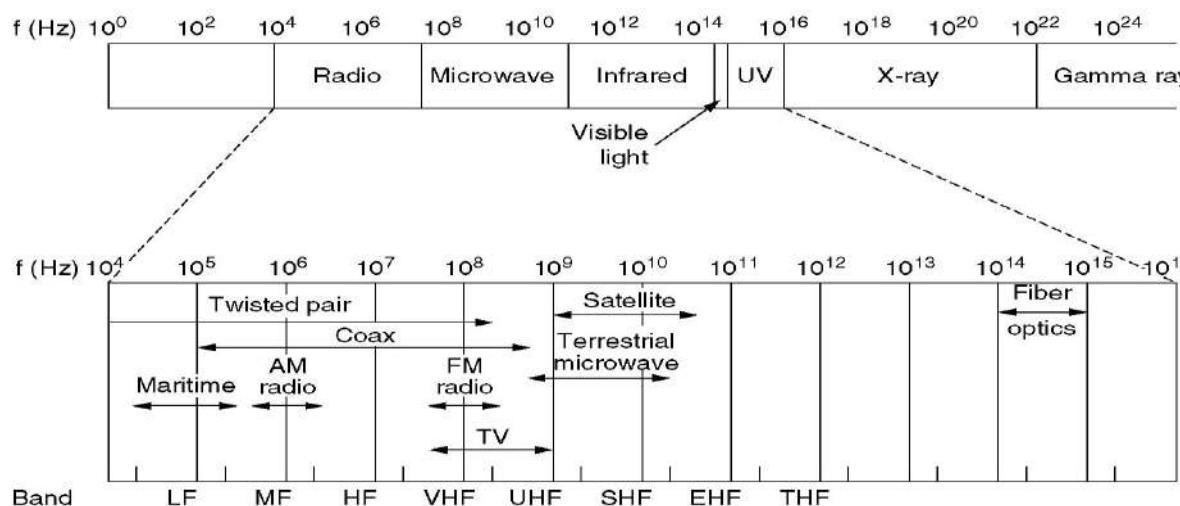


Figure 1: Atmospheric attenuation as a function of carrier frequency.

Utilizzo dello spettro e.m.

Al crescere della frequenza aumenta l'ampiezza del canale, ma peggiora l'interazione con l'ambiente.

A tutte le frequenze le onde sono soggette a disturbi (motori ed altri dispositivi elettrici) e ad interferenze con altre trasmissioni dati via etere. Per questo motivo i governi regolano e limitano l'utilizzo delle trasmissioni via radio mediante opportune licenze.

Molti governi hanno mantenuto libere alcune bande di frequenza, note come bande **ISM** (Industriale Scientifica Medica), che possono essere utilizzate da chiunque, senza licenza, a patto di rispettare **limiti di potenza** per limitarne le interferenze.

Principali apparati che utilizzano le Bande ISM:

- Telefoni CordLess, Forni a Microonde, Radiocomandi per cancelli automatici, LAN Wireless e Bluetooth.

Le bande ISM definite a livello mondiale sono:

902-928 MHz – 2.4-2.4835 Ghz – 5.725-5.875 GHz

http://it.wikipedia.org/wiki/Banda_ISM

Principali utilizzi per Trasmissione Dati

Ponti Radio

Connessioni punto-punto **terrestri** (decine di Km) e **satellitari** (migliaia di Km). I ponti radio utilizzano per le trasmissioni frequenze nel campo dei GHz (Microonde da 2.5 GKz a 23GHz) e quindi lunghezze d'onda dell'ordine del centimetro, per cui le antenne impiegate sono necessariamente del tipo parabolico. Viene utilizzata una larghezza di banda che può arrivare fino a qualche GHz.

Reti Locali

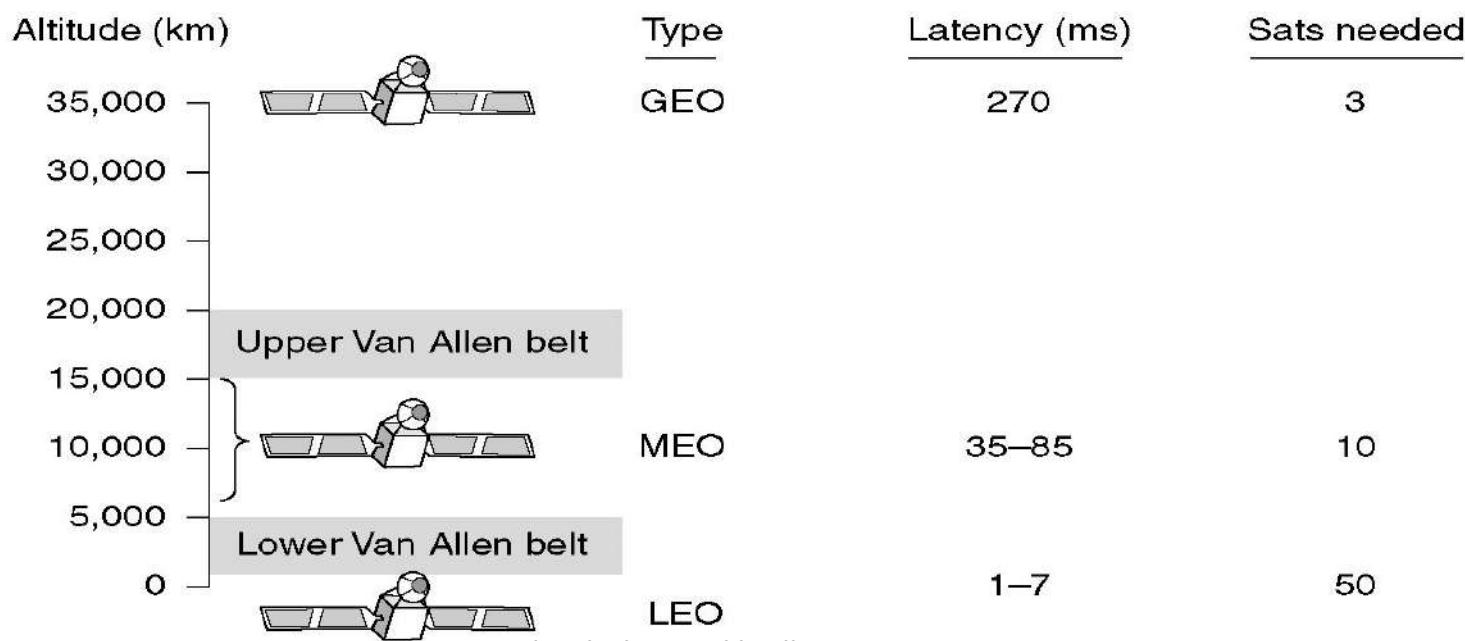
Connessioni omnidirezionali utilizzata all'interno degli edifici per realizzare reti multiaccesso. Si utilizzano microonde lunghe (2-5 GHz) per minimizzare i problemi di attraversamento delle pareti. L'ampiezza del canale può arrivare fino a 50 Mb/s in un raggio di **100-200 mt** (WiFi) o **qualche Km** (WiMax).

Comunicazioni Satellitari

Nel 1962 il satellite **TELSTAR** realizzò per la prima volta una trasmissione satellitare (segnale televisivo tra US e Francia). Il satellite aveva un orbita di 2 ore e 37 minuti, con una copertura di 20 minuti per passaggio.

Il periodo dell'orbita di un satellite dipende dalla sua altezza. Ad una altezza di 35800 Km il periodo è di 24 ore per cui il satellite è **geostazionario**. Con soli 3 satelliti geostazionari è possibile realizzare una copertura totale della terra. La latenza (andata e ritorno) è di $t = \text{distanza}/\text{velocità} = 71600 \text{ Km}/3 \times 10^8 = 270 \text{ mS}$

Il tempo di latenza può essere ridotto utilizzando orbite più basse, richiedendo però un elevato numero di satelliti per una copertura globale.



Progetti Satellitari

Reti satellitari :

TV analogica e digitale

- ▶ Diversi satelliti GEO

Posizionamento Terrestre (GPS)

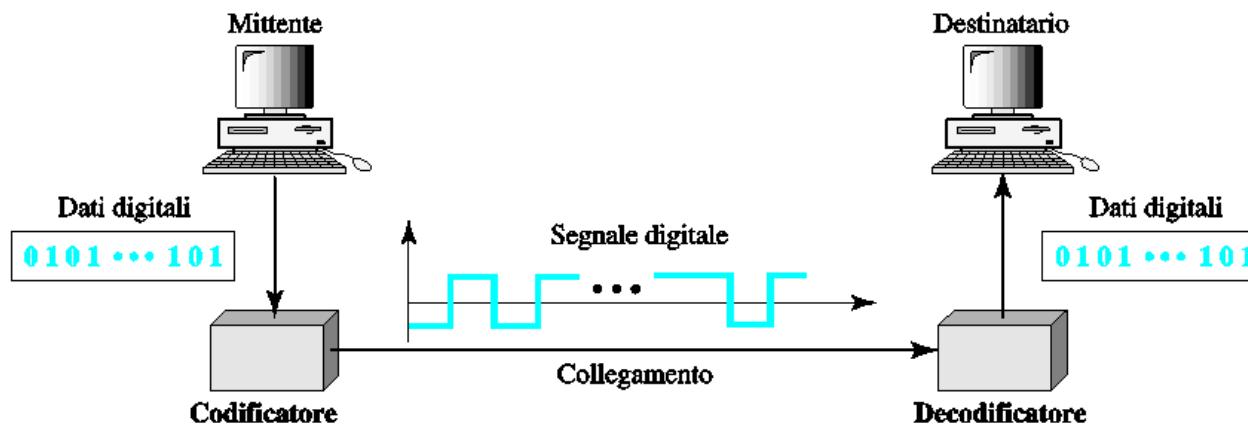
- ▶ 24 Satelliti MEO

Voce e Dati

- ▶ Iridium : 72 satelliti LEO, voce in tutto il pianeta
- ▶ Globalstar : 52 satelliti LEO, voce e dati in 120 paesi
- ▶ Starlink : 240 satelliti LEO nel 2020
- ▶ Inmarsat : 3 satelliti GEO , 50Mb/s
- ▶ Tooway : 2 satelliti GEO (Solo Europa) 22Mb/s

Modulazione digitale

La modulazione digitale è il processo di conversione dei dati digitali in segnali digitali che possono essere voltaggi, intensità di luce, o segnali elettromagnetici , secondo le caratteristiche della linea di comunicazione usata per il collegamento.



Il processo che assegna un segnale digitale a una sequenza di uno o più bit è detto codifica.

Codifiche dei bit

Principali tecniche di trasmissione:

Trasmissione in banda base

- Segnale modulato lasciando inalterata la frequenza

Trasmissione in banda passante

- Modulazione di una frequenza Portante

Il trasferimento dei bit avviene codificando sul canale due o più stati (simboli)

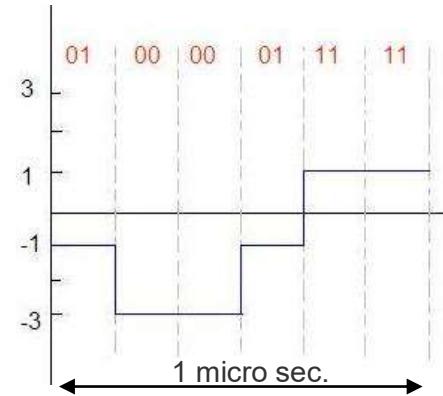
Il numero di simboli trasmessi in un secondo è detto **Baud-rate**.

Nell'esempio di figura (codifica con 4 simboli -3,-1,1,3)

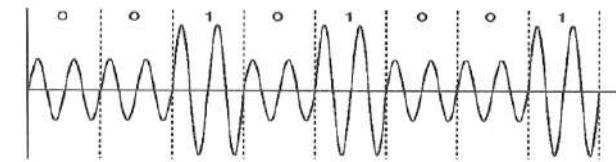
Boud-rate=6 Mbaud/s - Bit-rate=12 Mbit/s

Le onde quadre si deformano rapidamente al crescere della distanza (distorsione) , per cui le trasmissioni in Banda base sono adatte prevalentemente per Reti Locali.

Per lunghe distanze si utilizza la modulazione di un'onda sinusoidale, detta **portante**



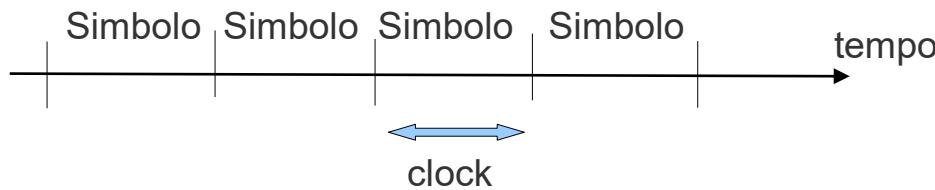
Modulazione in banda base



Modulazione (ampiezza) in banda passante

La trasmissione dei dati

La comunicazione nelle reti è generalmente seriale: si utilizza un solo canale (o eventualmente due per realizzare una connessione duplex con due canali simplex). I simboli vengono codificati all'interno di intervalli di tempo costante (**Clock**) che rappresentano il sincronismo condiviso tra trasmettitore e ricevitore.



Modalità Asincrona:

- ogni gruppo di bit è inviato in modo asincrono (è possibile l'assenza di segnale tra un gruppo e il successivo); ogni gruppo è preceduto da una sequenza di bit aggiuntivi che consentono al destinatario di ricostruire il sincronismo.

Questa è la modalità utilizzata nelle reti calcolatori.

Modalità Sincrona:

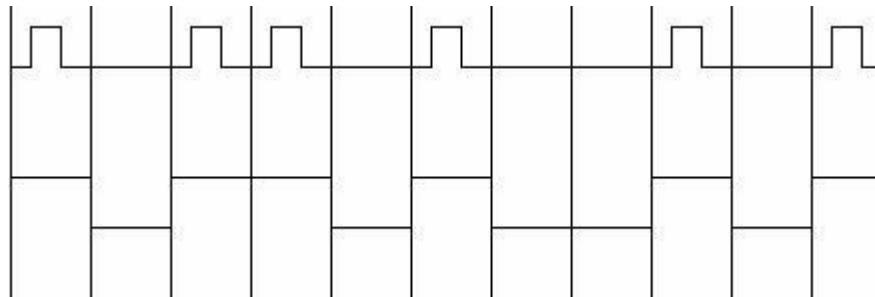
- Non sono previsti bit di sincronismo. Il segnale di Clock viene inviato parallelamente al canale dei dati, oppure il flusso non viene mai interrotto per non perdere il sincronismo.
- Generalmente questo è possibile solo per connessioni a breve distanza e alta velocità.

Schemi di codifica di linea: RZ e NRZ

Sequenza binaria

1 0 1 1 0 1 0 0 1 0 1

Return to Zero (RZ)



Non Return to Zero (NRZ)

NRZ non richiede circuiti complicati: i dati sono passati direttamente in uscita. E' robusto agli errori ma lunghe stringhe di zeri o uni potrebbero causare perdita di sincronismo.

RZ è più soggetto ad errori, ma non perde il sincronismo perché lo stato cambia ad ogni bit.

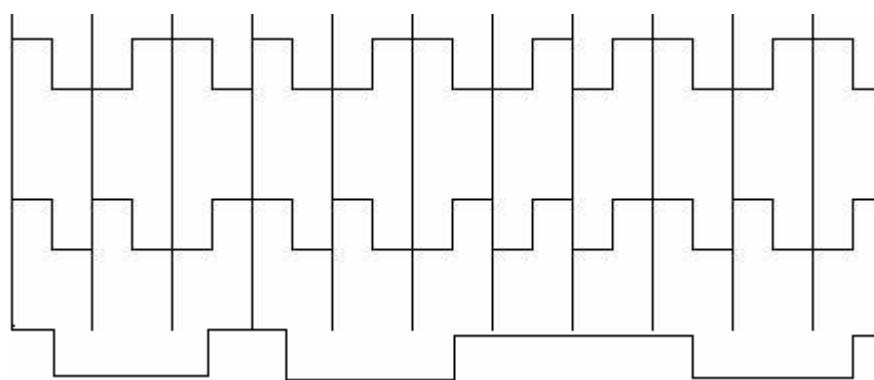
RZ e NRZ sono utilizzate nella PCM (trasmissione telefonica digitale).

Schemi di codifica di linea: NRZ-I e Manchester

Sequenza binaria

1 0 1 1 0 1 0 0 1 0 1

Manchester



Manchester differenziale

NRZ-I

NRZ-I cambia il simbolo di codifica in corrispondenza del bit 1, altrimenti rimane invariato.

La codifica **Manchester**, codificando i bit con le transizioni, è invece ideale per la gestione del sincronismo e per questo è utilizzata nel protocollo **Ethernet 10baseT**. Da notare che la codifica Manchester invia segnali ad una frequenza doppia.

Manchester differenziale combina la codifica Manchester (transizioni) e con NRZ-I (1 cambia il simbolo di codifica, 0 lo mantiene invariato).

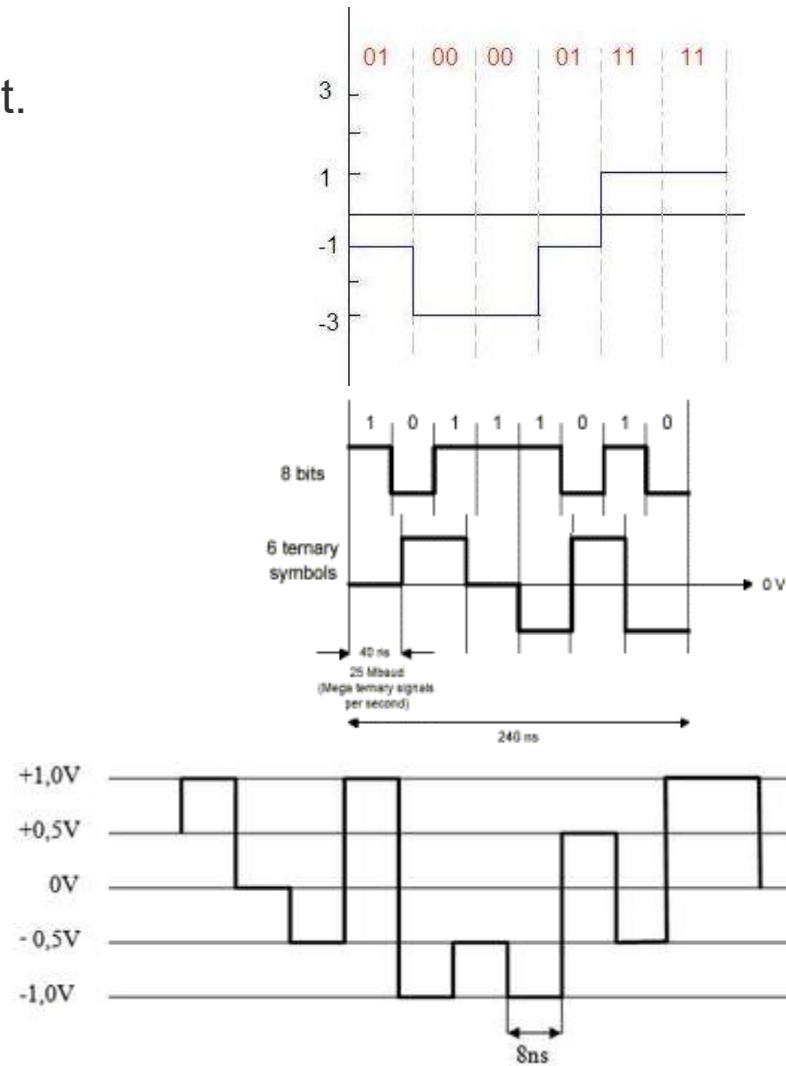
Codifiche di linea multi-livello

Se il canale ha un buon rapporto Segnale/Rumore, è possibile aumentare la velocità dei dati utilizzando schemi multi-livello. Alcuni esempi in uso:

2B1Q: utilizza 4 livelli di tensione per codificare 2 bit.
E' utilizzato da ISDN e alcune varianti di HDSL.

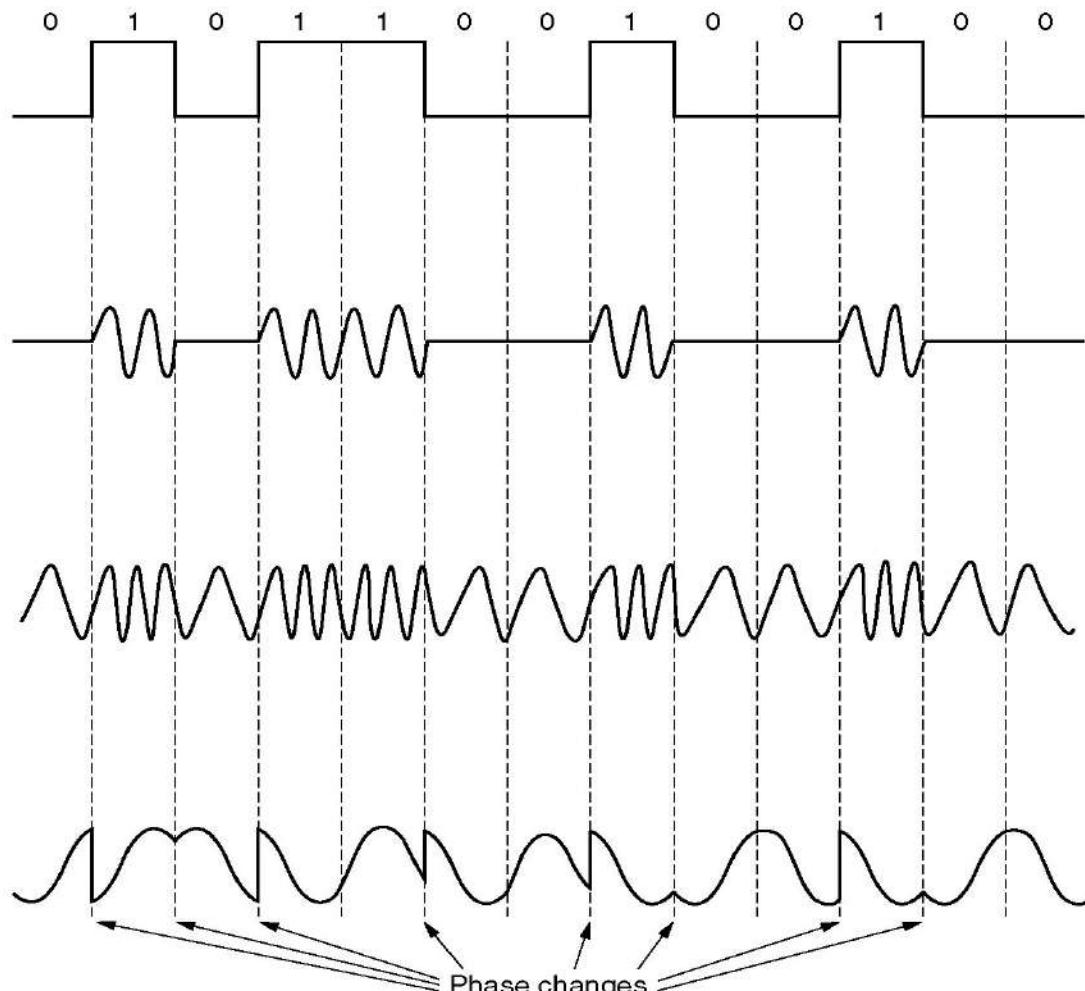
8B6T: sequenze di 8 bit sono codificate con sequenze di 6 simboli a 3 stati.
Utilizzato in 100baseT4

PAM5: utilizza 5 livelli di tensione
(0V serve per rilevare errori)
GigaBit Ethernet 1000baseT utilizza le 4 coppie del cavo UTP cat.5 e invia 125Mbaud (250Mbps) su ogni coppia



Modulazione di una frequenza portante

La codifica delle onde elettromagnetiche avviene **modulando** l'ampiezza, la frequenza o la fase (o una combinazione di questi metodi) di un'onda portante.



NRZ

[Modulazione di Ampiezza](#)

Esempio di utilizzo: fibre ottiche

[Modulazione di Frequenza](#)

[Modulazione di fase](#)

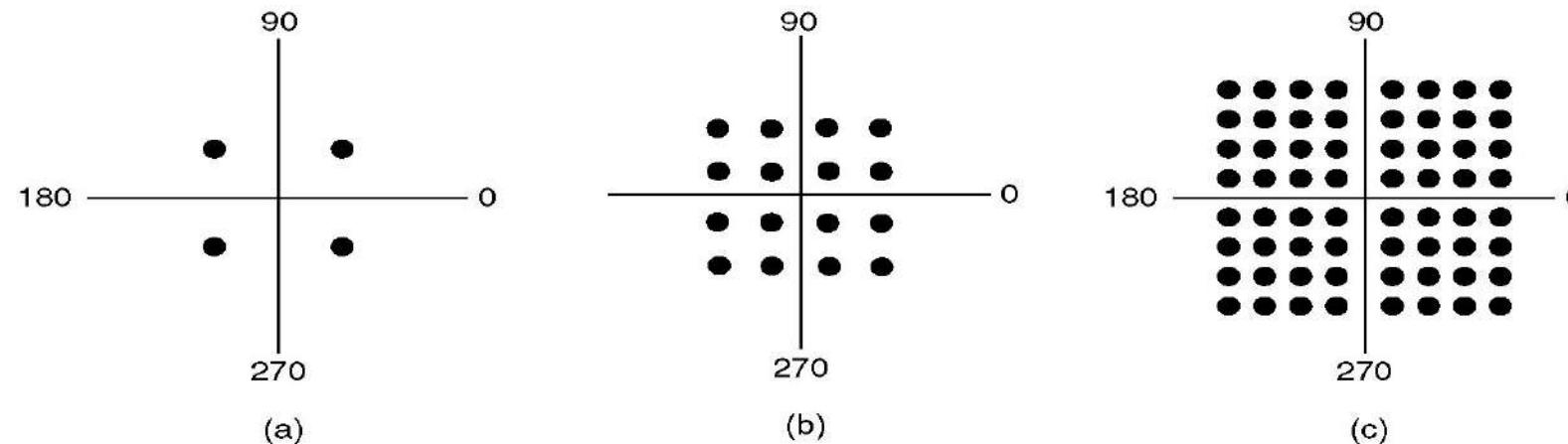
Diagrammi a costellazione

La modulazione combinata di ampiezza e fase è detta QAM (Quadrature Amplitude Modulation) . È la più efficiente e più usata.

Ogni simbolo è determinato da una coppia fase-ampiezza e viene rappresentato da un punto nel diagramma delle fasi. L'insieme dei punti formano una “costellazione”.

L'esempio sottostante riporta le costellazioni utilizzate nelle reti WiMAX:

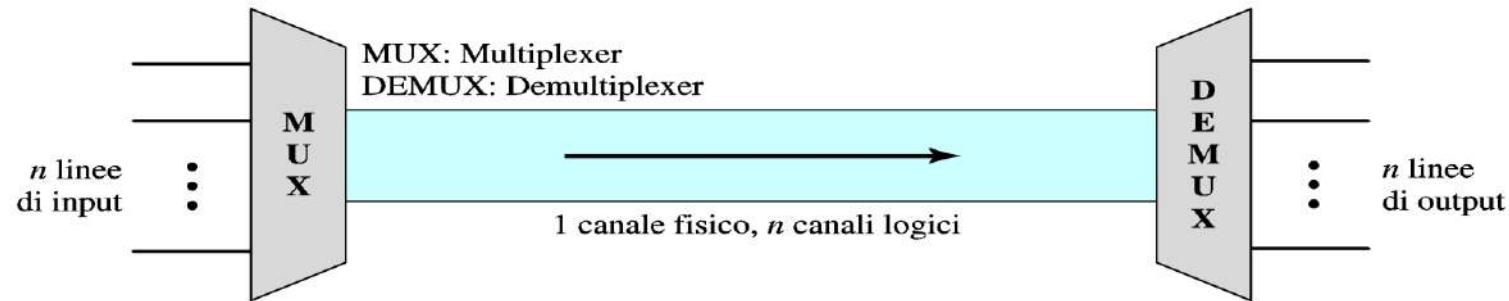
- (a) QPSK: modulazione di fase a 4 stati. (2 bit per ogni baud)
- (b) QAM16: modulazione di ampiezza e fase a 16 stati (4 bit per ogni baud)
- (c) QAM64: modulazione di ampiezza e fase a 64 stati (6 bit per ogni baud)



Multiplexing

Quando la larghezza di banda del canale trasmittivo è maggiore della larghezza di banda effettivamente necessaria, il canale può essere condiviso da più trasmissioni simultanee.

Il Multiplexing è la tecnica che permette la trasmissione simultanea di più segnali in un singolo canale.



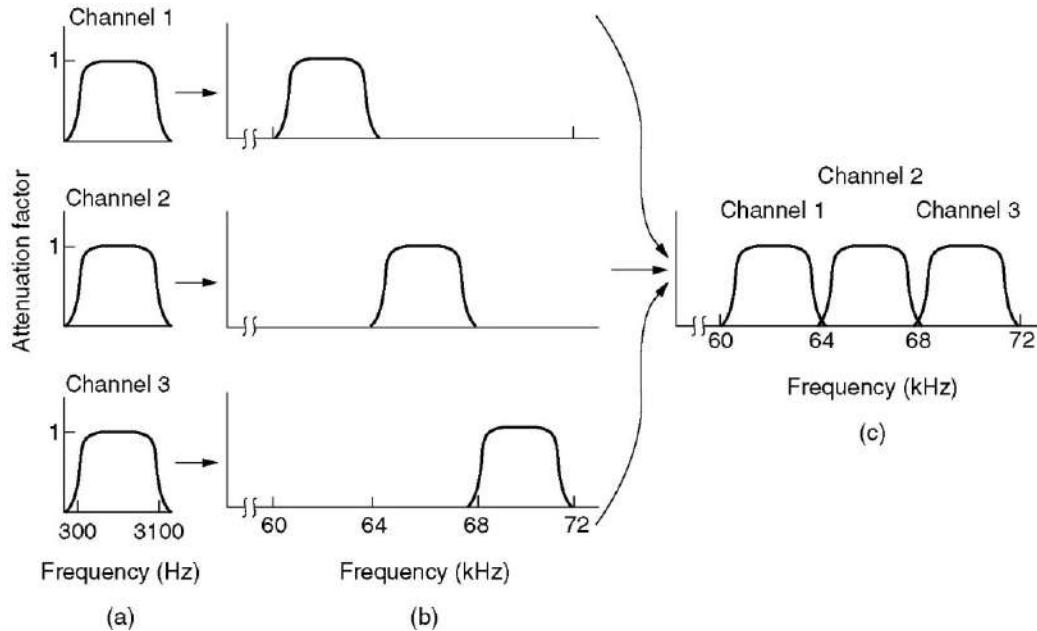
Esistono diverse tecniche per implementare il multiplexing.

Le principali sono:

- A divisione di frequenza **FDM**
- A divisione di tempo, **TDM**



Frequency Division Multiplexing



- (a) The original bandwidths.
- (b) The bandwidths raised in frequency.
- (c) The multiplexed channel.

Divide lo spettro in bande di frequenza. Ogni canale viene codificato modulando le frequenze all'interno di una banda.

Un esempio tipico sono le trasmissioni radiofoniche AM; lo spettro di frequenze utilizzato è di circa 1 MHz (da 500 a 1500 KHz). Ogni stazione radio opera all'interno di una banda di circa 10KHz, con una separazione tra i canali abbastanza grande per evitare interferenze.

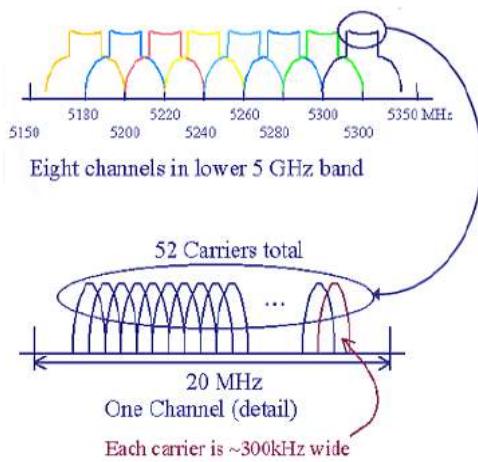
OFDM

OFDM (Orthogonal Frequency Division Multiplexing) è una tecnica FDM in cui le frequenze portanti tra loro ortogonali: le fasi di portanti adiacenti sono calcolate in modo da non interferire.

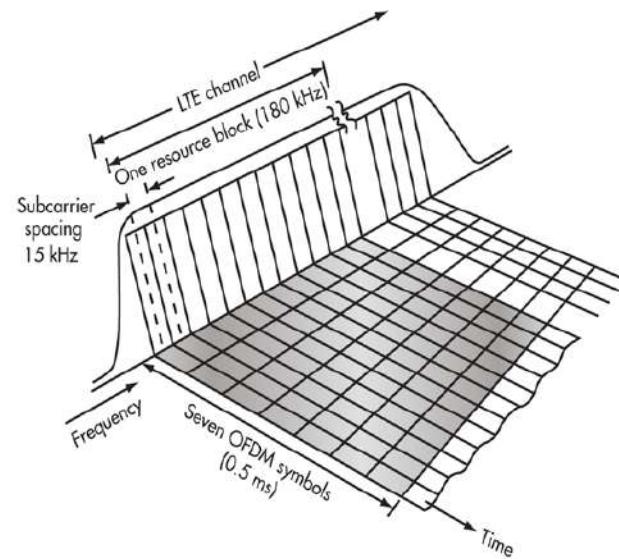
N portanti adiacenti compongono un canale che può essere utilizzato per una singola trasmissione in cui i dati vengono inviati in parallelo.

Le singole sotto-portanti (subcarrier) possono utilizzare codifiche diverse in base al profilo dell'attenuazione.

E' utilizzata nelle principali tecnologie per trasmissione dati quali ADSL, WiFi 802.11g e 802.11n e nei sistemi cellulari LTE.



OFDM in WiFi

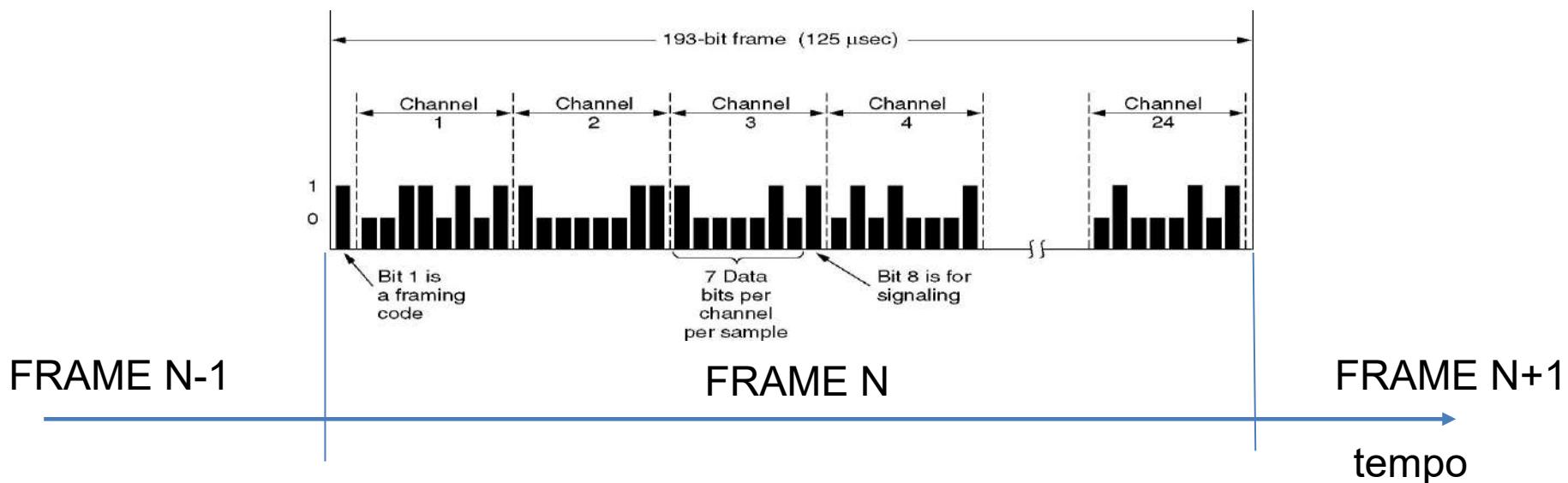


OFDM in LTE

Time Division Multiplexing (TDM)

Il TDM è tipico delle trasmissioni digitali. Il tempo viene diviso in frame di dimensione stabilita. Ogni frame ospita un numero definito di canali.

I dati di un linea di input vengono inseriti nel canale assegnato.





UNIVERSITÀ
DI PARMA

DIPARTIMENTO DI SCIENZE MATEMATICHE, FISICHE ED INFORMATICHE
Corso di Laurea in Informatica

Il sistema telefonico

RETI DI CALCOLATORI - a.a. 2023/2024

Roberto Alfieri

Il livello Fisico: sommario

PARTE I

- ▶ Scopo dello strato Fisico
- ▶ Il canale di comunicazione
- ▶ I mezzi Trasmissivi
 - Trasmissioni su rame, il Doppino, il cablaggio strutturato
 - Trasmissioni su Fibra Ottica
 - Trasmissioni via Etere
- ▶ Le codifiche dei bit: banda base e banda passante

PARTE II

- ▶ Il sistema Telefonico, FDM, TDM, DSL, la telefonia mobile

RIFERIMENTI

- ▶ *Reti di Calcolatori, A. Tanenbaum, ed. Pearson*
- ▶ *Reti di calcolatori e Internet, Forouzan , Ed. McGraw-Hill*

Il sistema telefonico

Il sistema telefonico (PSTN - Public Switched Telephone Network) è una rete specializzata per la trasmissione di uno specifico tipo di dato: la voce analogica.

Perché ci interessa:

- ▶ E' un modello di confronto per la rete di trasmissione dati.
- ▶ E' una infrastruttura di rete capillare consolidata da più di 100 anni di attività.
- ▶ In alcuni casi è utilizzata come canale di comunicazione per trasmissione dati.

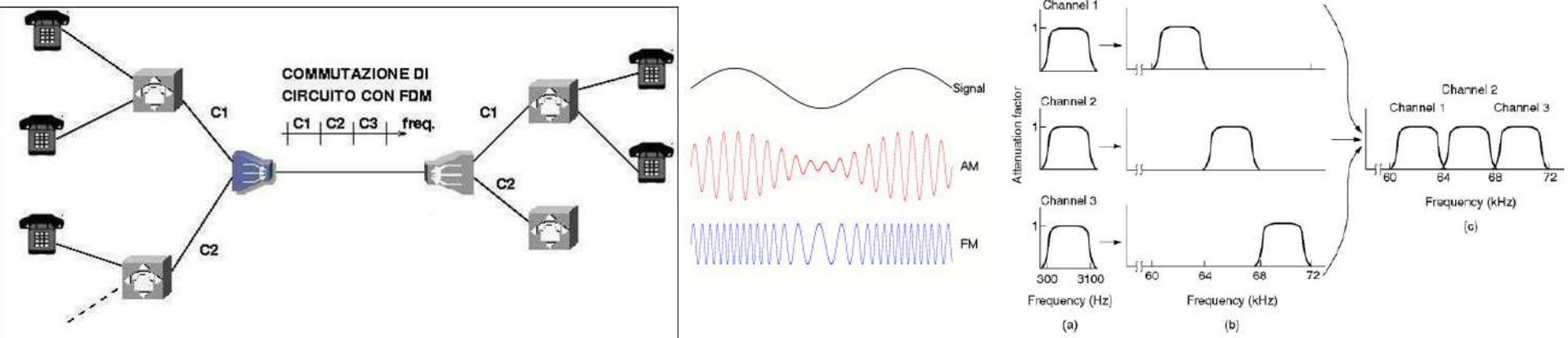
Il sistema telefonico analogico

Le prime reti telefoniche (**fino agli anni 60**) erano analogiche.

Ogni telefonata richiede una banda analogica di 4KHz tra l'utente e la centralina telefonica su un doppino telefonico in rame detto Ultimo Miglio (o Local Loop)

Nelle dorsali interne delle Telecom per poter trasportare più circuiti telefonici sullo stesso canale si poteva utilizzare la tecnica di multiplazione FDM (Frequency Division Multiplexing):

La banda di frequenze analogiche del mezzo trasmittivo veniva suddivisa in canali di almeno 4 KHz e ciascun canale poteva ospitare una telefonata modulando una frequenza portante all'ingresso, per poi demodularla uscita.



Il sistema telefonico digitale: PCM

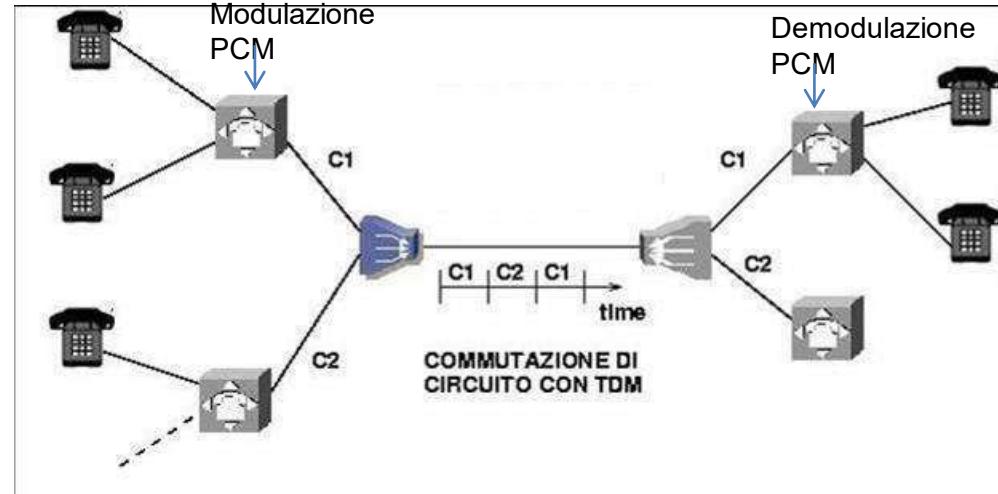
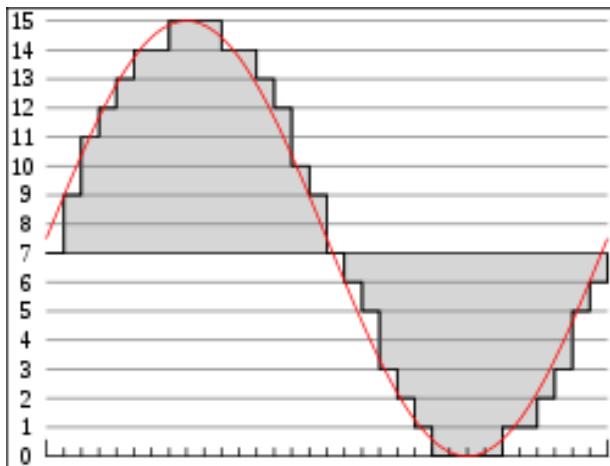
A partire dagli anni 60 le comunicazioni audio vengono gestite in modo digitale dalle compagnie telefoniche. La conversione in digitale è standardizzata con la tecnica di campionamento PCM (**Pulse Code Modulation**) eseguita nelle tratte della rete con un Codec che converte il segnale in forma digitale e lo riconverte in analogico in prossimità della destinazione.

Un canale analogico di 4 KHz richiede 8000 camp/sec ovvero uno ogni $125\mu\text{s}$

Ogni campionamento avviene generalmente su 8 bit di dati (o 7 dati + bit parità)

Si ottiene così un flusso di **64Kbit/sec** (o 56Kbit/sec).

In questo caso la multiplazione delle portanti avviene in Time Division Multiplexing, ovvero suddividendo il tempo del canale in slot che si ripetono ciclicamente.



Portanti TDM per canali PCM

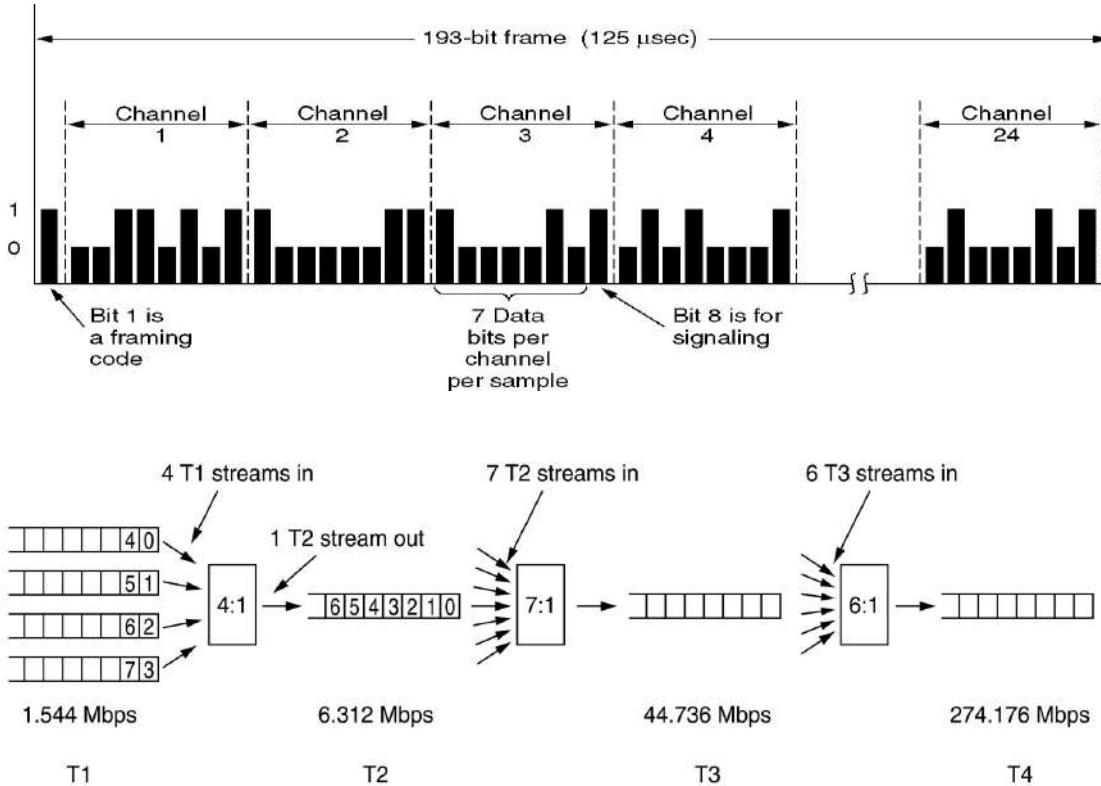
In Nord America e Giappone i canali PCM sono raggruppate con tecniche di multiplexing TDM su portanti in rame di tipo T (T1, T2, T3 e T4 ...):

T1 può portare 24 canali telefonici PCM da 64Kb/s (1.5Mb/s)

T2 trasporta 4 canali T1 (6.3Mb/s)

T3 trasporta 7 canali T2 (44.7 Mb/s)

T4 trasporta 6 canali T3 (274 Mb/s)



Le altre nazioni usano le portanti di tipo E (definite da CCITT) con una gerarchia a 32 (portante E1 a 2.048Mb/s) , 128, 512, 2048 e 8192 canali.

Digitalizzazione dell'ultimo miglio

Modem, ISDN e xDSL

La digitalizzazione non coinvolgeva l'utente che continuava a lavorare in analogico. Volendo utilizzare il sistema telefonico per la trasmissione dati era quindi necessario modulare i bit con segnali analogici assimilabili alla voce umana, inviarli nel sistema telefonico e demodularli in digitale al lato ricevente. Entrambe queste operazioni venivano eseguite da un apparato denominato Modem, il quale avendo a disposizione un canale di 4KHz poteva trasmettere fino a **56 Kbit/s** (teorema del campionamento di Shannon).

Con la tecnologia **ISDN** (anni 80/90) è stato possibile propagare il segnale digitale anche sull'ultimo miglio, portando in casa dell'utente un canale digitale da **64Kb/s**.

Negli anni 2000 per aumentare la velocità è stato introdotto la tecnologia **xDSL** che sfrutta la maggiore banda di frequenze dei cavi in **categoria 3**.

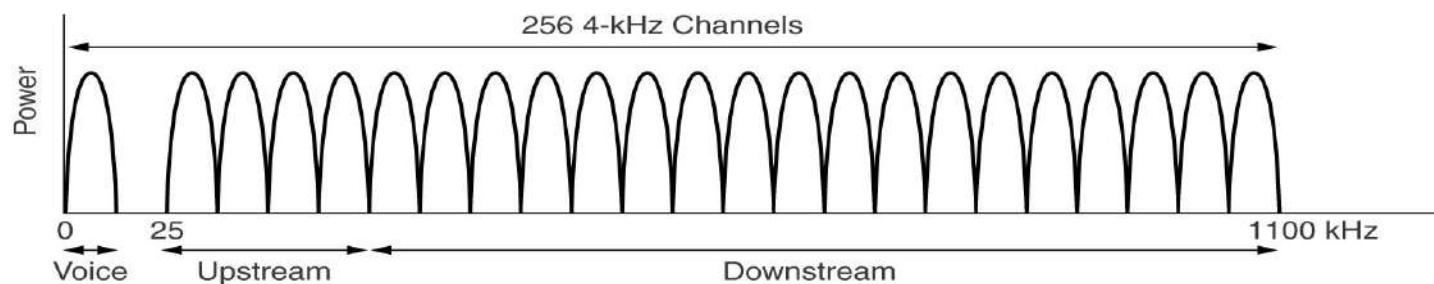
- Il segnale domestico non viene filtrato a 4 KHz ma a **1.1MHz**
- La banda viene suddivisa in 256 canali da **4.3KHz** (1.1 MHz / 256 canali) con tecnica OFDM.
- All'interno di ogni canale si usa la modulazione QAM con un rate di 4K baud.
- La qualità della linea viene costantemente monitorata per aggiustare la velocità di trasmissione, utilizzando costellazioni con più o meno punti.
- Massimo numero di bit per baud è 15 (32768-QAM)
- Massima velocità di un canale **60 Kb/s** (4K baud * 15 bit per baud)
- Massima velocità aggregata è quindi ~ **15 Mb/s** (60 Kb/s * 256 canali)

ADSL

ADSL (Asymmetric DSL) è un utilizzo specifico di xDSL, pensato per l'home computing, in cui il download è prevalente.

Con ADSL abbiamo:

- Canale 0 (0 – 4.3 KHz) per la fonia
- Canali 1-5 (4.3 - 25 KHz) non utilizzati per evitare interferenze fonia/dati
- Canali 6-30 (24) per upload. Max: $24 \times 60 \text{ Kb/s} = 1.4 \text{ Mb/s}$ (effettivi 0.5 Mb/s)
- Canali 31-255 (224) per download. Max: $224 \times 60 \text{ Kb/s} = 13.4 \text{ Mb/s}$ (effettivi 8 Mb/s)



ADSL, ADSL2 e ADSL2+

ADSL2 migliora le prestazioni (effettivi ~ 12 Mb/s) attraverso una diversa codifica con una maggiore efficienza.

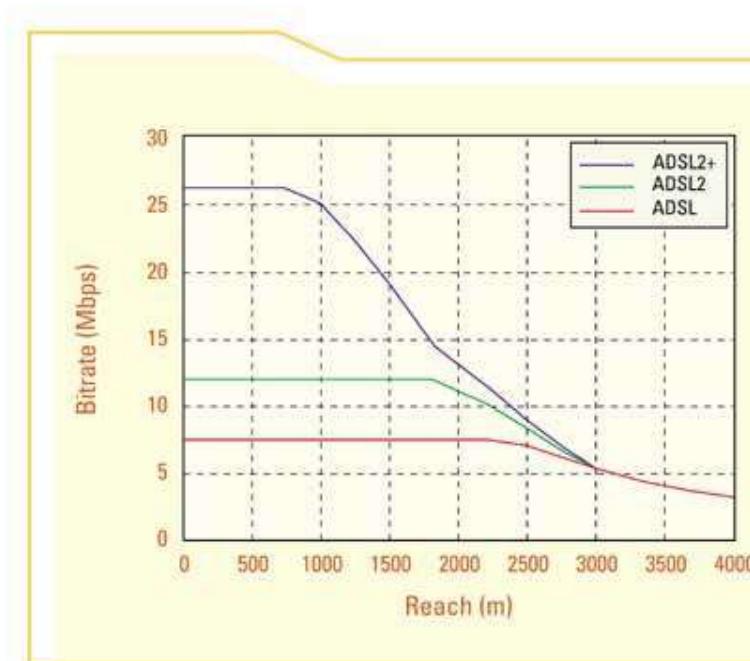
ADSL2+ è un nuovo standard che utilizza una banda doppia, di 2.2MHz. In questo caso la massima velocità è di ~ 26 Mb/s per brevi distanze

Tecnologia adattiva

In realtà solo chi abita in prossimità della centrale telefonica riesce ad arrivare ai 60Mb/s per canale perché il segnale decade rapidamente con la distanza. Dato che l'intera banda non ha una curva costante di segnale/rumore e attenuazione, ogni canale viene monitorato e modulato in modo indipendente.

Secondo le potenzialità di ogni canale il sistema decide il tipo di QAM da utilizzare.

Quando i bit utilizzabili scendono sotto una soglia minima il canale viene escluso.



FttX

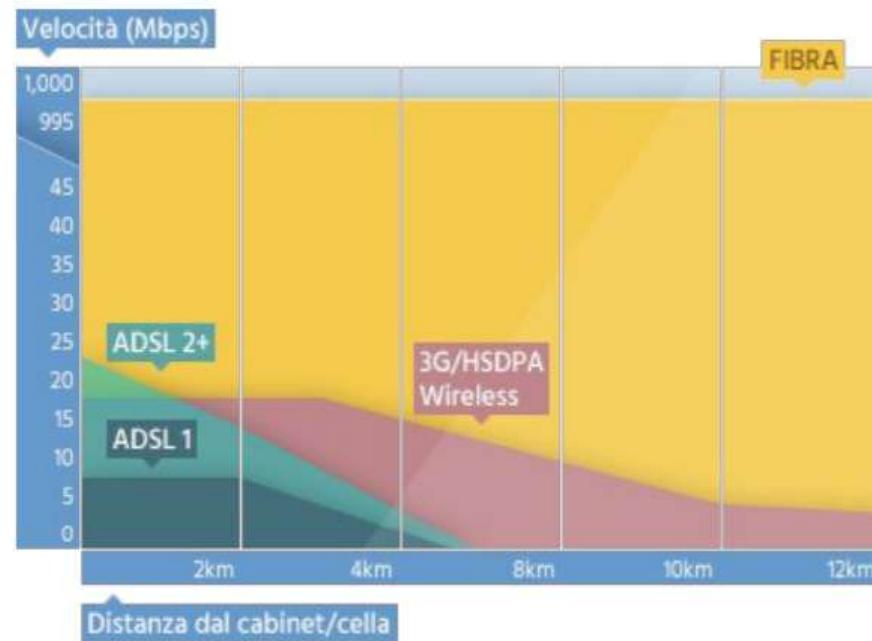
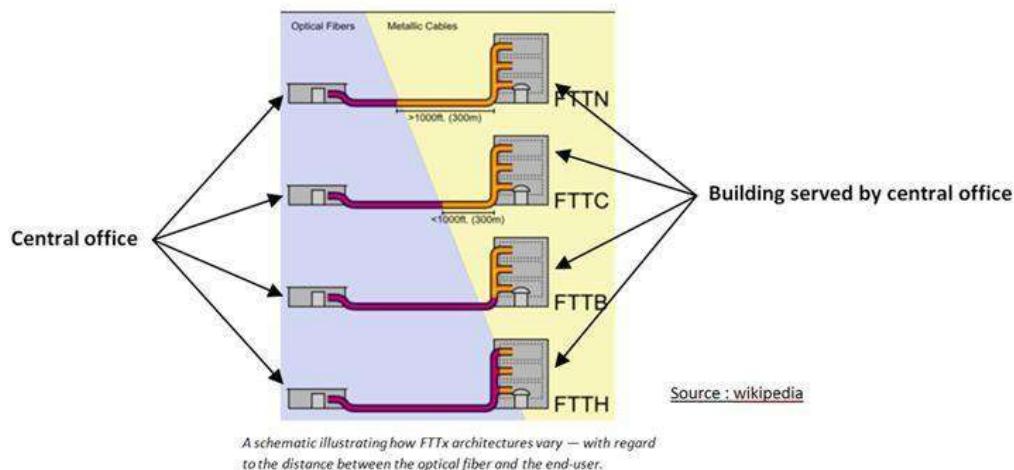
L'ultimo miglio in rame limita le prestazioni di ADSL.

Le compagnie telefoniche stanno sostituendo il rame con fibre ottiche che arrivano in prossimità dell'abitazione o in casa.

FttC (Fiber to the Cabinet, ovvero l'armadio in strada) può arrivare ad una banda di 35Mhz e una velocità di 300 Mb/s

FttH (Fiber to the Home) arriva a 1 Gb/s

Queste tecnologie vengono genericamente riferite come FttX.



<https://fibra.click/architetture/>

Il sistema telefonico mobile

Esistono diverse generazioni di telefoni cellulari, con diverse tecnologie:

1G: Standard analogici TACS (Europa) e AMPS (America)

2G: D-AMPS e GSM . Voce digitale

2.5G: GPRS e E-GPRS (EDGE). Trasmissione digitale a commutazione di pacchetto

3G: Standard UMTS, CDMA e HSDPA. Voce e dati digitali

4G: Standard LTE

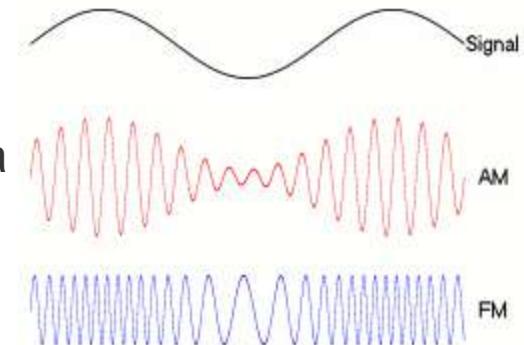
5G: (2020)

1G: AMPS e TACS

Voce analogica

Tecnologia nata nel 1982 denominata AMPS negli USA e TACS in EU

I segnali vocali analogici (3KHz) vengono modulati in frequenza e inseriti in canali di 30KHz gestiti con il metodo FDMA.



AMPS utilizza 832 canali full duplex, ciascuno dei quali composto da 2 canali simplex:

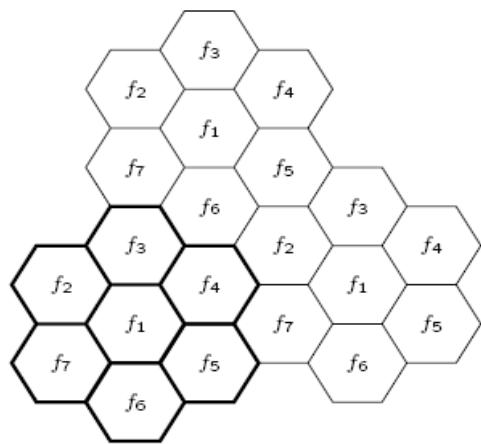
- 832 canali da 30KHz per trasmissione tra 824/849 Mhz (banda totale 25MHz)
- 832 canali da 30KHz per ricezione tra 869/894 Mhz (banda totale 25MHz)



1G: AMPS e TACS

Le celle

Ogni area geografica è divisa in celle di 10-20 Km di diametro.
Le celle sono organizzate in nuclei di 7 e hanno frequenze diverse.
In questo modo due 2 celle con le stesse frequenze sono distanziate da 2 celle diverse.
Questo significa che ogni cella ha mediamente un settimo degli 832 canali disponibili.



2G: D-AMPS

Voce digitale

D-AMPS. Progettato per coesistere con AMPS e utilizza gli stessi canali:

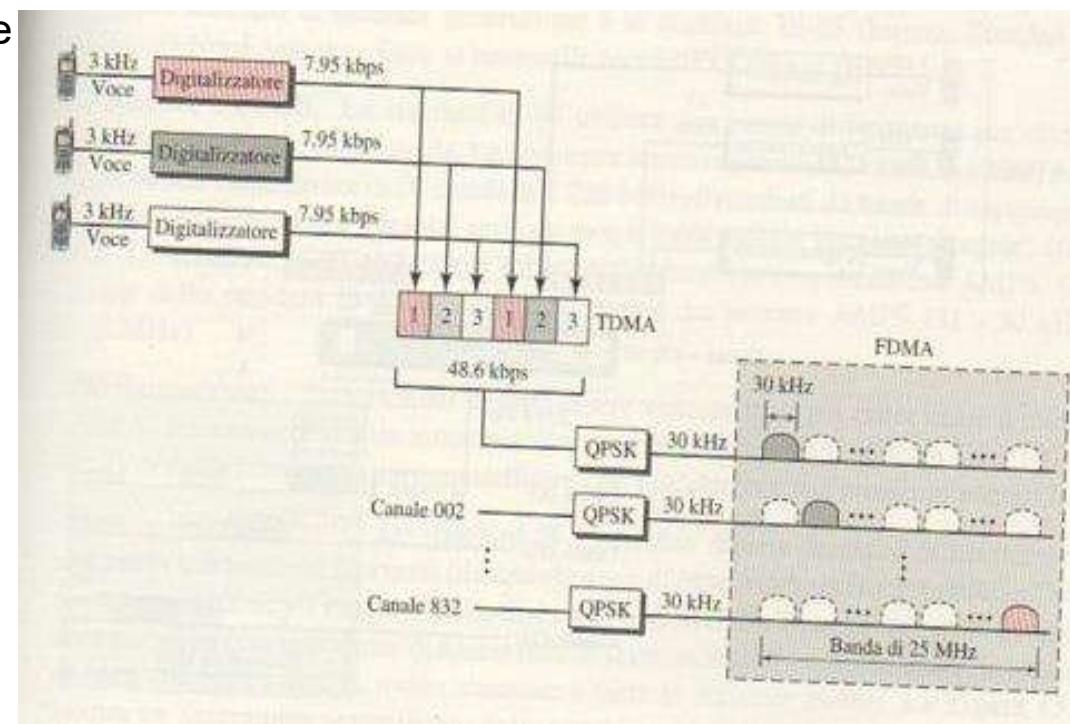
- 832 canali da 30KHz per trasmissione tra 824/849 Mhz (banda totale 25MHz)
 - 832 canali da 30KHz per ricezione tra 869/894 Mhz (banda totale 25MHz)
- più nuove frequenze a 1.9GHz. Utilizzato negli USA

I canali analogici da 3KHz vengono digitalizzati con PCM (56 Kbps) e compressi a 8 Kbps

Ogni coppia di frequenze da 30Khz viene condivisa da 3 utenti contemporaneamente in divisione di tempo (TDM).

Sulla coppia di frequenze vengono inviati 25 frame al secondo (40ms) e ogni frame è diviso in 6 slot temporali in cui vengono alternati in TDM i 3 flussi digitali che vengono modulati con QPSK (modulazione di fase).

D-AMPS utilizza quindi multiplexing TDM entro un multiplexing FDM.

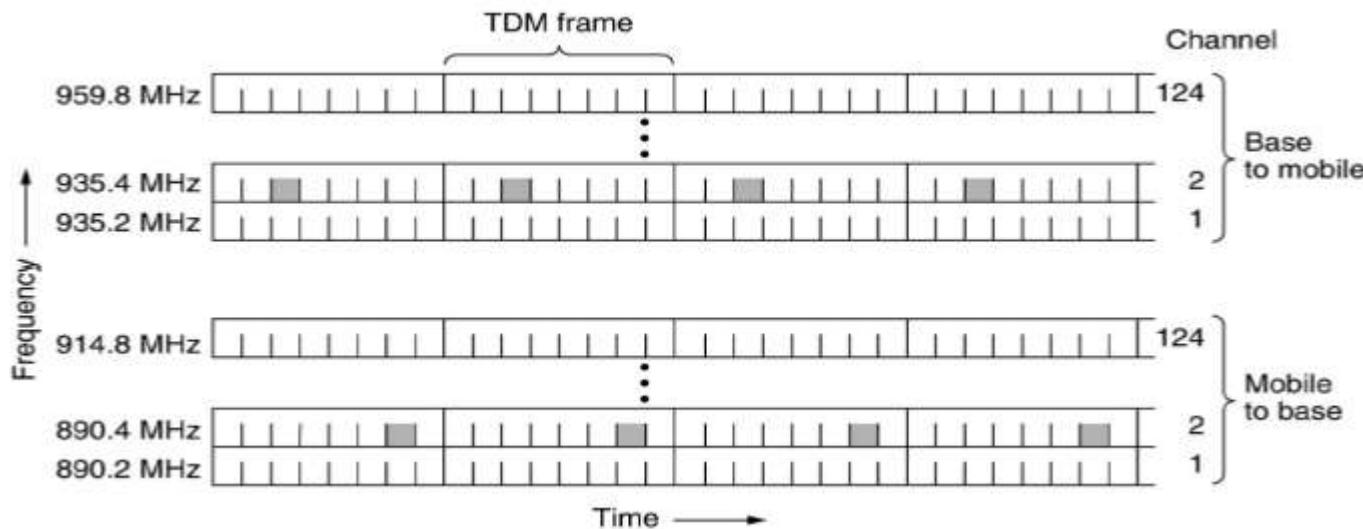


2G: GSM

Voce digitale

GSM. Utilizzato nel resto del mondo. Come D-AMPS utilizza FDM+TDM

- FDM: 124 canali simplex di 200KHz
- TDM: su ogni canale si susseguono Frame di 4.6 msec, suddivisi in 8 slot temporali ciascuno di 148 bit per 8 connessioni separate.



2.5G: GPRS e EDGE

Trasmissione digitale a commutazione di pacchetto

GPRS 2.5G in attesa di 3G.

Velocità teorica 171Kbps (reale 30-80 Kbps)

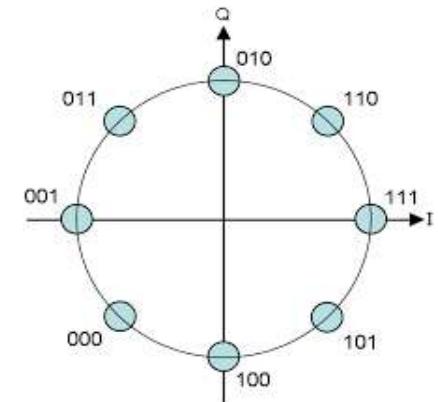
E' una sovrastruttura sopra D-AMPS e GSM per trasportare pacchetti IP raggruppando più slot.

Enhanced GPRS (EDGE)

Velocità incrementata introducendo una nuova modulazione, la 8-PSK (modulazione di fase a 8 simboli).

Prestazioni:

EDGE (E) fino a 473,6 Kbps teorici



3G: UMTS e HSDPA

Voce e dati digitali

Successore di terza generazione (3G) del GSM di cui utilizza l'infrastruttura, ma la tecnologia di trasmissione e' W-CDMA

CDMA (Code Division Multiple Access) e' una tecnica a diffusione dello spettro (Spread Spectrum) che permette di raggiungere velocità di trasmissione per utente superiori rispetto alla tecnica di accesso mista TDMA/FDMA utilizzata nella rete GSM/GPRS.

I protocolli HSPA come HSDPA e HSPA+ sono stati introdotti nello standard UMTS per migliorarne le prestazioni attraverso l'utilizzo di un numero maggiore di simboli di codifica.

Prestazioni

UMTS (3G): fino a 384 Kb/s in download, 128 Kb/s in upload, latenza 150 ms.

HSDPA (H) : fino a 14 Mb/s in download, 5,7 Mb/s in upload, latenza 100 ms.

HSPA+ (H+): fino a 43 Mb/s download, 11 Mb/s upload, latenza 50 ms.

Dal 2022 la rete 3G è dismessa in Italia

<https://www.tim.it/assistenza/info-consumatori/news/2021/dismissione-rete-3g>

4G: LTE

LTE (Long Term Evolution) è la generazione per i sistemi di accesso mobile a banda larga standardizzata nel 2008 e introdotta in Italia nel 2013.

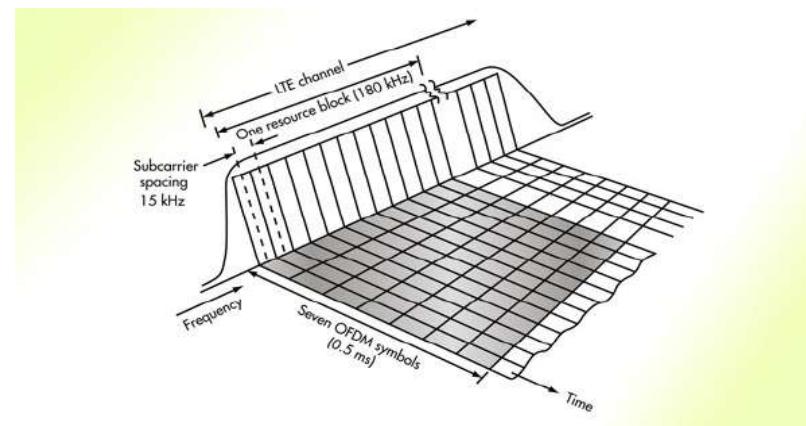
Nel 2010 ITU ha autorizzato l'utilizzo della denominazione 4G per le tecnologie LTE

- Utilizzo della **modulazione OFDM** (QPSK, 16QAM, 64QAM) per il downlink e Single-Carrier FDMA per l'uplink (al posto del W-CDMA dell'UMTS);

- utilizzo di un minimo di 1,25 MHz ed un massimo di 20 MHz per ciascun utente

Prestazioni:

LTE (4G) fino a 326,4 Mb/s in download
e fino a 86,4 Mb/s in upload.
RTT < 10 ms



VoLTE, acronimo di **Voice over LTE**, è una tecnologia che consente di effettuare chiamate vocali su rete [LTE \(4G\)](#) basandosi sul modello architettonale [IP Multimedia Subsystem \(IMS\)](#) [[wikipedia](#)].

Con VoLTE anche le chiamate voce utilizzano la rete dati TCP/IP.

5G

Il termine 5G (acronimo di 5th Generation) indica l'insieme di tecnologie di telefonia mobile e cellulare, i cui standard definiscono la quinta generazione della telefonia mobile con una significativa evoluzione rispetto alla tecnologia 4G/IMT-Advanced. La sua distribuzione globale si è avviata nel 2019. [[Wikipedia](#)]

3 bande di frequenza:

- **Banda alta (25-39 GHz)**
 - onde millimetriche (11 mm)
 - Alta velocità (> 1 Gb/s) e bassa latenza
 - Solo nei grandi centri urbani, Piccole celle, visibilità ottica
- **Banda intermedia (2,5-3,7 GHz)**
 - la più comune, velocità tra 100 e 400 Mb/s
- **Banda bassa (694 e 790 MHz)**
 - Bassa velocità (30 – 250 Mb/s), Elevata copertura

Adaptive beam switching: possibilità di saltare da una banda all'altra

<https://www.agendadigitale.eu/infrastrutture/5g-ecco-le-tecnologie-pilastro-tutto-cio-che-ce-da-sapere>



UNIVERSITÀ
DI PARMA

DIPARTIMENTO DI SCIENZE MATEMATICHE, FISICHE ED INFORMATICHE
Corso di Laurea in Informatica

Il Livello Data-Link

Parte I : Protocolli

RETI DI CALCOLATORI - a.a. 2023/2024

Roberto Alfieri

Il Livello Data-Link: sommario

PARTE I

- ▶ Scopi del livello Data-Link
- ▶ Framing, Rilevazione e correzione degli errori, controllo di flusso
- ▶ Protocolli per reti Punto-punto: PPP.
- ▶ Protocolli per reti MultiAccesso: Aloha, CSMA, CSMA/CD, CSMA/CA

PARTE II

- ▶ Gli standard IEEE802
- ▶ Ethernet: Sottoliv. MAC e LLC, tecnologie Ethernet, il Frame, Repeater, Switch, Bridge
- ▶ Spanning Tree Protocol.
- ▶ Lan Virtuali

PARTE III

- ▶ Lan Wireless

RIFERIMENTI

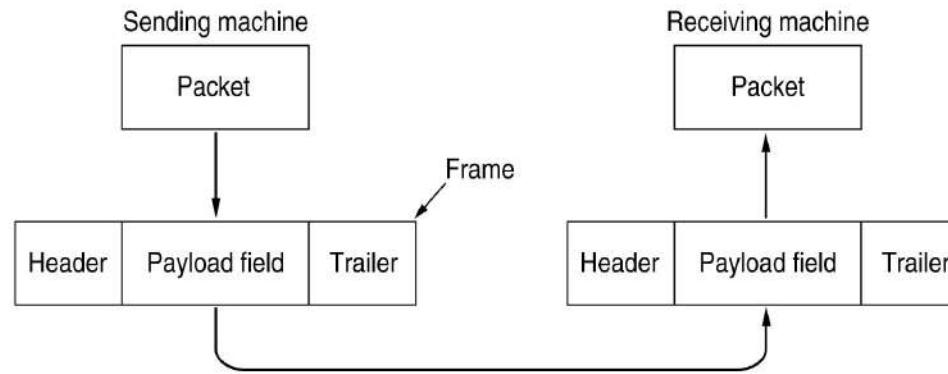
- ▶ *Reti di Calcolatori, A. Tanenbaum, ed. Pearson*
- ▶ *Reti di calcolatori e Internet, Forouzan , Ed. McGraw-Hill*

Scopi del Livello Data-Link

Il livello Data-Link ha la responsabilità di trasferire in modo sufficientemente affidabile i dati tra nodi adiacenti, ovvero su canali punto-punto o multi-accesso.

La comunicazione affidabile è realizzata mediante la **Suddivisione del flusso di dati in “frame”** con lunghezza massima fissata.

Il Frame contiene un payload che viene riempito con i dati da trasportare provenienti dal livello superiore e informazioni di servizio poste in testa e/o in coda al payload (header/trailer).



Le informazioni di servizio vengono utilizzate principalmente per:

- ▶ Delimitare inizio e fine del frame (framing)
- ▶ Gestire la **rilevazione ed eventualmente la correzione degli errori**.
- ▶ Gestire (eventualmente) il **controllo del flusso**
- ▶ Gestire l'**accesso al mezzo trasmissivo** (nei canali Multi-Accesso)

Servizi offerti al livello Network

I servizi forniti al livello network possono essere:

▶ **Senza connessione e senza conferma**

- Semplice e veloce, è adatto a mezzi trasmissivi affidabili (Es. Ethernet)

▶ **Senza connessione ma con conferma**

- Viene inviato un Frame di conferma per ogni Frame inviato.
- Utile se il mezzo è poco affidabile (es. LAN Wireless)

▶ **Con connessione e con conferma**

- Ogni frame inviato è parte di una connessione e quindi dotato di una numerazione
- Garantisce che ogni Frame viene ricevuto una sola volta e riordinato.
- 3 fasi distinte:
 - attivazione della connessione
 - invio dati numerati e conferme
 - chiusura connessione
- Overhead elevato. Raramente utilizzato a livello Link. Implementato nei livelli superiori (TCP)

Impacchettamento (Framing)

Il primo problema da risolvere è come delimitare inizio e termine di un frame.

I frame possono avere dimensione fissa o variabile. Se la dimensione è fissa non è necessario delimitare il frame (vedi la rete telefonica ATM). Se la dimensione è variabile occorre una strategia per distinguere i frame.

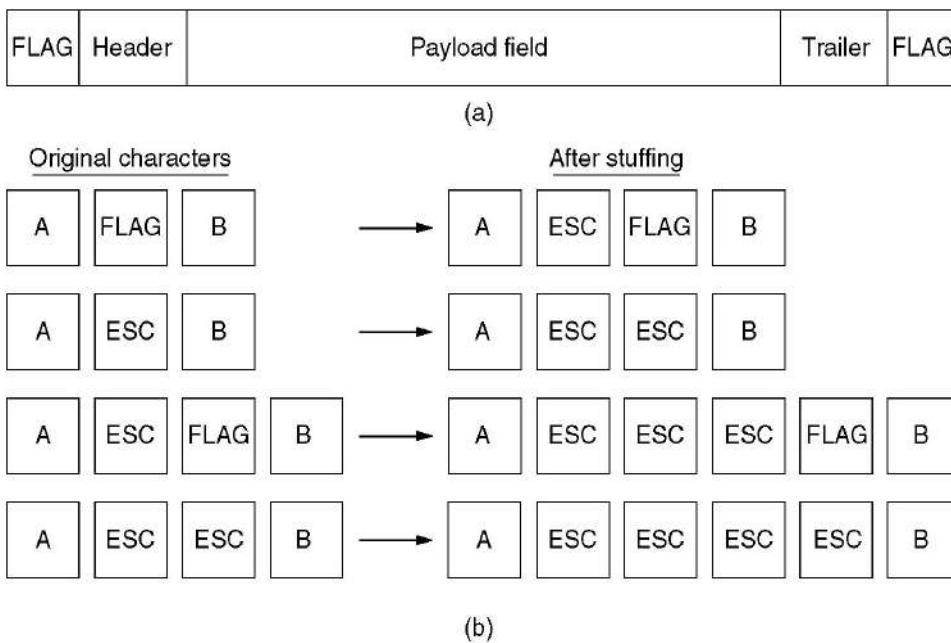
Possibili strategie:

- ▶ Un intervallo **temporale** tra un frame ed il successivo
- ▶ Far precedere ogni Frame con il **numero di byte del frame**.
- ▶ Delimitare il Frame con caratteri speciali (**Flag**)

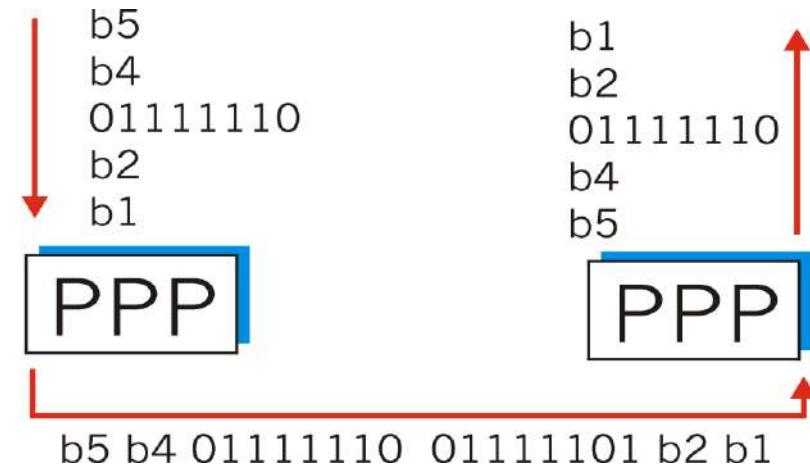
Gran parte dei protocolli di Data-Link usano l'abbinamento del Flag e dell'intervallo temporale per aumentare la ridondanza.

Framming con “ESC Stuffing”

La delimitazione del Frame è marcata da un Byte speciale denominato **FLAG**.
Un problema potrebbe nascere se all'interno del Frame è presente la sequenza di FLAG.
Per flussi **Byte-Oriented**, si può inserire un Byte di Escape (ESC) appena prima dell'occorrenza accidentale del Flag (o dell'ESC).
Il destinatario dovrà realizzare l'operazione di “destuffing”.



Esempio PPP (ESC = 01111101).



Bit stuffing

Se i flussi sono **bit-oriented** si può utilizzare il “Bit-stuffing”:

Ogni Frame inizia e termina con la sequenza 01111110 (Flag)

Ogni volta che nella trama si incontra la sequenza 11111 (5 uni) viene aggiunto un bit 0 (bit stuffing) per non confondere il destinatario.

(a) 011011111111111110010

(b) 01101111011111011111010010

Stuffed bits

(c) 011011111111111111110010

(a) Dati originali

(b) Dati elaborati dal mittente che aggiunge i bit di stuffing

(c) Dati elaborati dal destinatario che elimina i bit di stuffing.

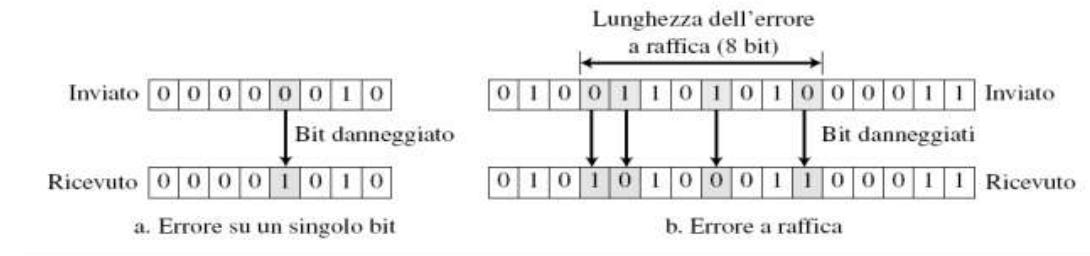
Rilevazione e Correzione degli errori

Durante la trasmissione di un Frame possono verificarsi disturbi o rumore termico che possono cambiare la forma del segnale e quindi alterare la ricezione dei bit.

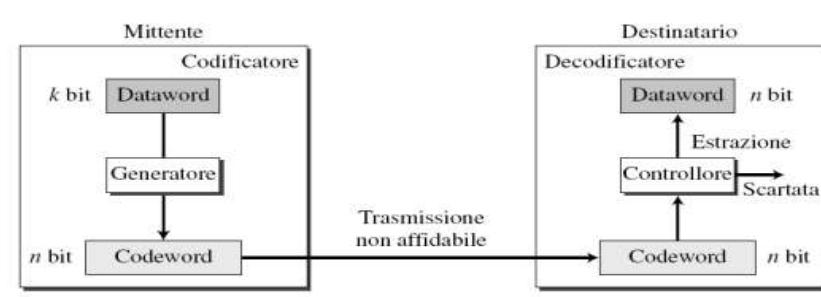
Gli errori sono rari su linee ottiche, mentre possono essere frequenti su canali come wireless o “ultimo miglio” sulla linea ADSL.

Tipi di errori:

- ▶ **a bit singolo**
- ▶ **a raffica (“burst”)**



Per individuare gli errori si utilizza la **Ridondanza**: il mittente, attraverso un opportuno algoritmo, determina una breve codice (Frame Control Sequence - FCS), che verrà inviata assieme al Frame. Se il destinatario riapplicando l'algoritmo otterrà una sequenza FCS diversa capirà che si è verificato un errore.



Rilevazione e Correzione degli errori: algoritmi

Esistono 2 strategie possibili:

Rilevazione degli errori (senza correzioni): richiede algoritmi più semplici ed un FCS più breve. Si utilizza su canali affidabili (es. Fibra Ottica), in cui gli errori sono rari e conviene eventualmente **ritrasmettere** il Frame.

Nota: La richiesta di ritrasmissione può essere

- esplicita: il ricevente manda un NACK in caso di errore
- automatica con protocollo ARQ (Automatic Repeat ReQuest): il mittente attiva un timer, il ricevente invia un ACK per i dati ricevuti correttamente e scarta i dati con errore; allo scadere del timer il mittente rispedisce il frame.

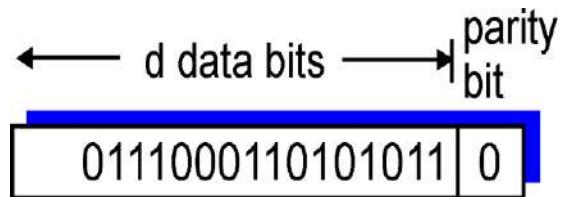
ARQ è in grado di gestire sia frame errati che frame perduti

Rilevazione e correzione degli errori: richiede algoritmi più complessi e maggiore ridondanza nel FCS. Si utilizza raramente, in reti poco affidabili o in trasmissioni Simplex, in cui non è possibile inviare al mittente la richiesta di ritrasmissione.

Codifica a blocchi: bit di parità

Semplice algoritmo per rilevazione dell'errore.

Il numero totale di 1 nella sequenza, compreso il bit di parità, deve essere dispari (o pari)



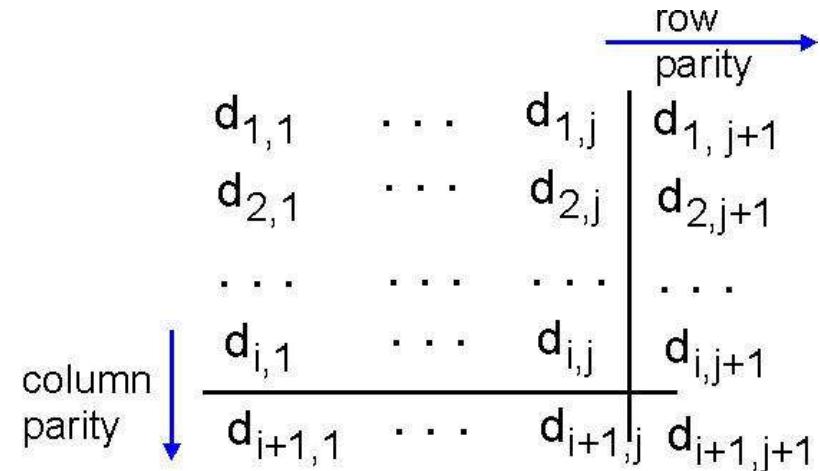
Applicato alla sequenza di un Frame determina l'esistenza di un singolo errore all'interno della sequenza.

Il bit di parità si usa in molti dispositivi hardware come ad esempio nei bus SCSI e USB e in molte cache di microprocessori

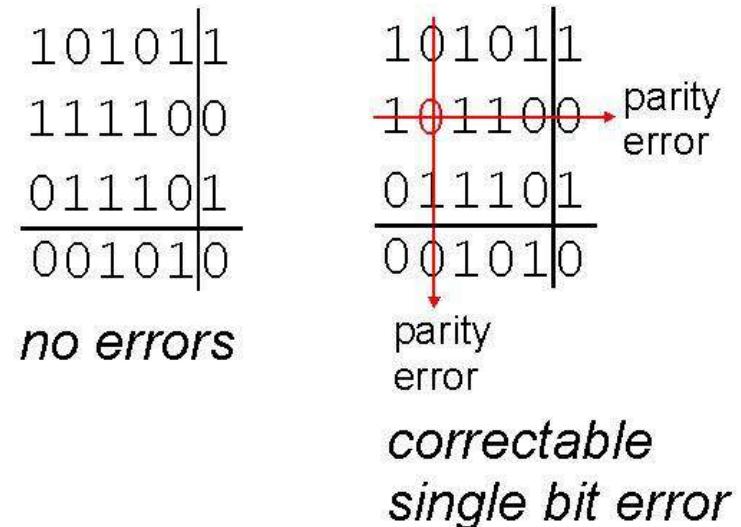
Codifica a blocchi: bit di parità 2D

Semplice algoritmo **esemplificativo** per la correzione dell'errore.

Suddividendo il Frame in più sotto-sequenze di uguale lunghezza possiamo calcolare la parità in 2 dimensioni e quindi individuare e correggere il singolo bit errato.



Questa tecnica è però poco efficiente e **non trova impieghi reali**.



Cyclic Redundancy Check (CRC)

Un Frame di d bit è visto come una lista di coefficienti di un polinomio D con d termini (di grado $d-1$). Per esempio 110001 rappresenta $x^5 + x^4 + x^0$

Trasmettitore e Ricevitore si mettono d'accordo su di un polinomio comune G di $r+1$ bit (grado r) detto “generatore”, che deve essere un numero primo.

Il **Trasmettitore** aggiunge r bit (il CRC) al termine della sequenza del Frame in modo che il nuovo Frame M (di grado $r+d-1$) sia divisibile per G.

Procedura:

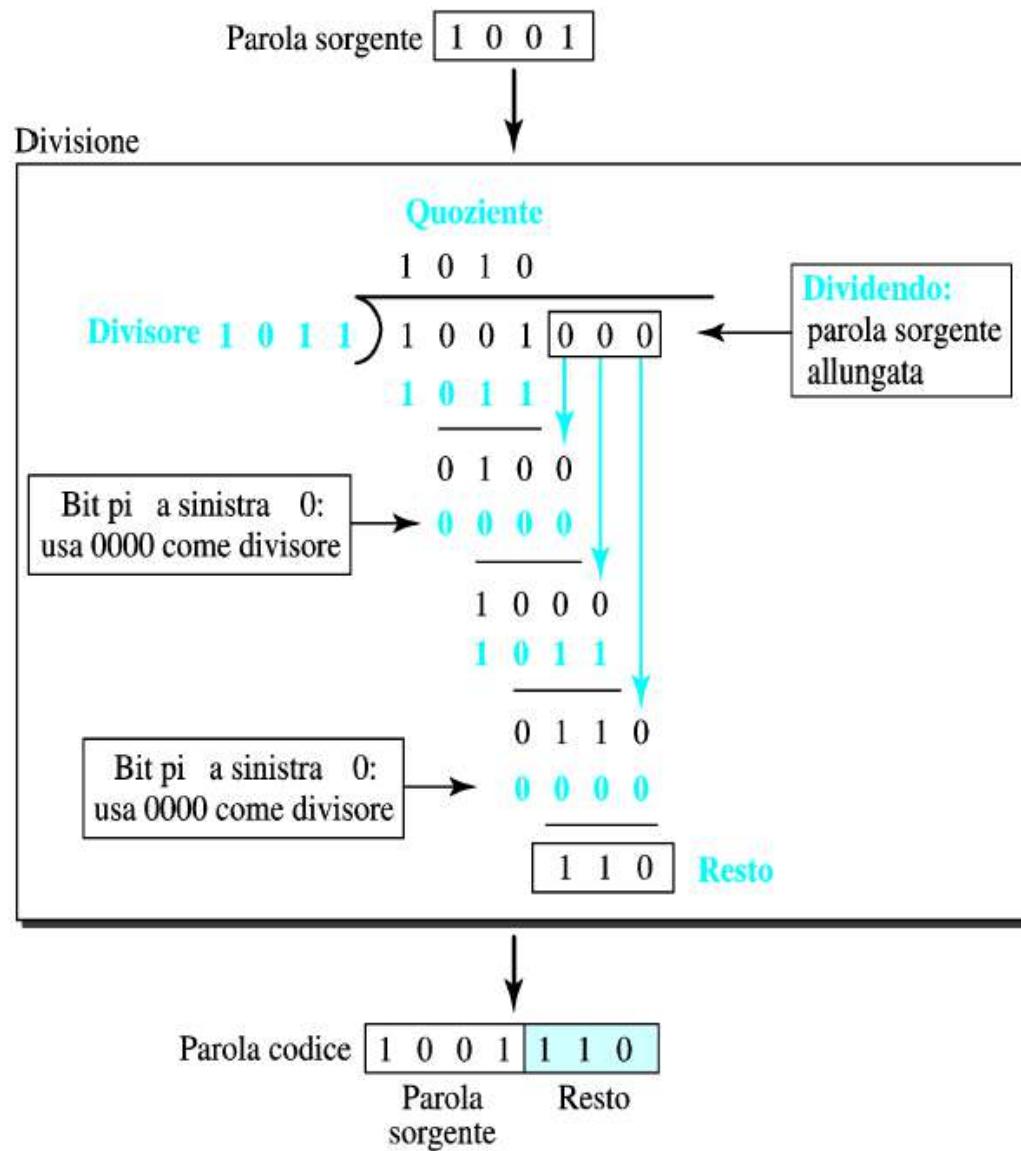
- 1) $N = D \cdot x^r$ (vengono aggiunti r zeri al termine di D)
- 2) $R = N/G$ (viene determinato il resto R della divisione)
- 3) $M = N - R = N \text{ xor } R$ ($N-R$ è divisibile per G. La sottrazione in mod2 si fa con XOR).

Il **Ricevitore** divide M/G. Se il resto è diverso da zero si è verificato un errore.

Procedura:

- 1) $R=M/G$ (viene determinato il resto R della divisione)
- 2) if ($R \neq 0$) then ERROR

CRC: esempio di calcolo



Il polinomio generatore (divisore) è di 4 bit (grado 3)

Si calcola l'XOR tra i primi 4 bit del dividendo e divisore.

Al risultato si aggiungono progressivamente i bit rimanenti del divisore e si ripete l'XOR con il divisore.

Se il bit più a sinistra del dividendo è 0, in quel passo occorre un divisore fatto di tutti 0.

CRC

Uno dei vantaggi del CRC è che i moduli di codifica e decodifica possono essere **facilmente implementati in hardware** usando componenti elettronici poco costosi.

La codifica polinomiale CRC con r bit di controllo è in grado di rilevare sequenze di errori di lunghezza fino a r.

Il CRC è il codice più utilizzato nei protocolli data-link:

Ethernet: Usa il CRC-32 ($x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x^1+1$) riesce a rilevare fino a 32 errori e tutti quelli che toccano un numero dispari di bit.

PPP: Usa CRC-16

ATM: Usa CRC-8

Checksum (somme di controllo)

Il checksum è un'altra tecnica, basata sulla rappresentazione dei numeri in complemento a 1, per individuare differenze tra i bit inviati da mittente e quelli ricevuti dal destinatario.

Il mittente divide il segmento in blocchi da 16 bit e li somma, quindi ne fa il complemento e inserisce il risultato nel campo checksum.

Il ricevente ricalcola il checksum (includendo il checksum ricevuto) senza complemento finale.

Il risultato, senza errori di trasmissione è una sequenza di 1, altrimenti c'è un errore.

E' adatto per implementazioni software e per questo non è usato a livello Link ma nei livelli superiori (IP, ICMP, TCP, UDP)

Il termine **Checksum** è spesso utilizzato per intendere in generale le tecniche per verificare l'integrità di un dato o di un messaggio.

Checksum: operazioni

Esempio a 8 bit:

Mittente:	Destinatario:
10101000	10101000
01010010	01010010
10001110	10001110
-----	-----
10001000 (somma)	01110111 (checksum)
01110111 (checksum)	-----
	11111111 (checksum OK)

La funzione mittente a 16 bit:

```
chsum(u_short *buf, int count)
{
    register u_long sum=0;
    while (count --)
    {
        sum += buf++;
        if (sum & 0xffff0000) {sum &= 0xffff; sum++;}
    }
    return ~(sum & 0xffff);
}
```

Protocolli per il controllo del flusso

Protocolli condivisi tra mittente e destinatario per garantire il corretto invio del flusso dei dati. Possono essere implementati a livelli Link o ai livelli superiori.

Protocolli in modalità non connessa

Per canali senza rumore

- ▶ Semplice
- ▶ Stop-and-wait (con conferma)

Protocolli in modalità connessa

Per canali con rumore

Si basano sulla numerazione dei frame

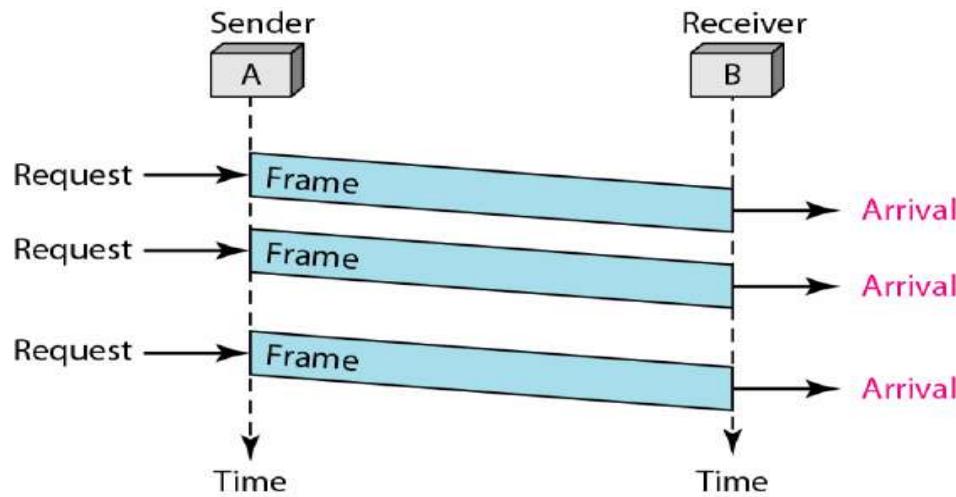
- ▶ Stop-and-wait ARQ
- ▶ Protocolli a finestra scorrevole (Sliding Window)
 - Go-back-N ARQ
 - Ripetizione selettiva ARQ

Protocollo semplice (simplex) senza restrizioni

Scenario (ideale):

- ▶ Il destinatario è sempre pronto a ricevere e a gestire i frame ricevuti.
- ▶ I dati arrivano senza errori

Protocollo Semplice:



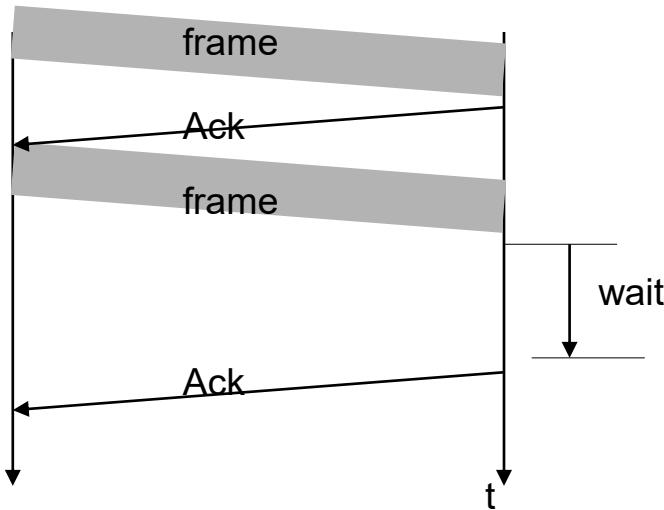
Protocollo stop-and-wait

Scenario:

- ▶ Il ricevente ha bisogno di tempo per elaborare i dati ricevuti

Protocollo Stop-and-Wait:

- ▶ Prima di inviare il prossimo dato il mittente deve ricevere una **conferma (Ack)**, per cui il destinatario, se sovraccarico, può moderare il tasso di invio dei dati ritardando l'invio di Ack (controllo di flusso con conferma).



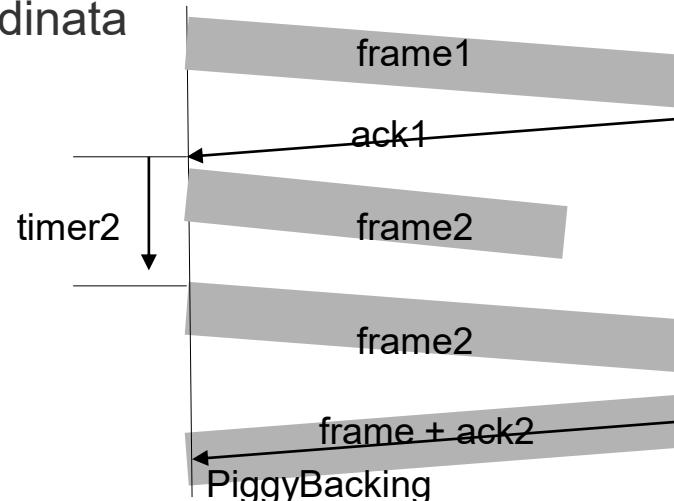
Protocollo stop-and-wait ARQ

Scenario:

- ▶ Il ricevente ha bisogno di tempo per elaborare i dati ricevuti
- ▶ I frame possono essere danneggiati o perduti (canale rumoroso o disturbato)

Protocollo Stop-and-Wait ARQ (Automatic RePeat reQuest):

- ▶ Poiché i frame possono andare perduti viene gestito l'invio di un frame di servizio, denominato ACK, a conferma della corretta ricezione.
- ▶ I frame danneggiati vengono scartati, oppure viene inviato un NACK (Not ACK).
- ▶ Per gestire la perdita di frame, il mittente attiva un timer per ogni frame inviato. Se il mittente non riceve un ACK in un certo tempo il frame viene **rispedito**.
- ▶ **Servizio di Connessione** : consegna garantita e ordinata grazie a **numerazione** dei frame e dei relativi ACK
- ▶ Il mittente deve mantenere copia dei frame fino all'ACK
- ▶ Se il traffico è bidirezionale l'ACK può viaggiare in un frame dati inviato in senso opposto (**PiggyBacking**).
- ▶ Nei protocolli Stop-and-wait la capacità del canale non è sfruttata al meglio poiché occorre attendere l'arrivo dell'ACK.



Protocolli “Sliding Window”

I protocolli “**Sliding Window**” migliorano l’efficienza del canale consentendo al trasmettitore di poter inviare fino SWS (Sender Window Size) frame senza attendere il riscontro ACK.

La finestra si sposta in avanti man mano che i riscontri arrivano. I Frame appartenenti alla finestra vengono memorizzati dal mittente per eventuali ritrasmissioni.

Il mittente assegna ad ogni Frame un **numero di Sequenza**.

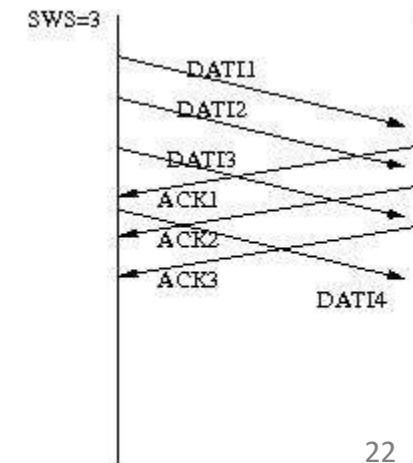
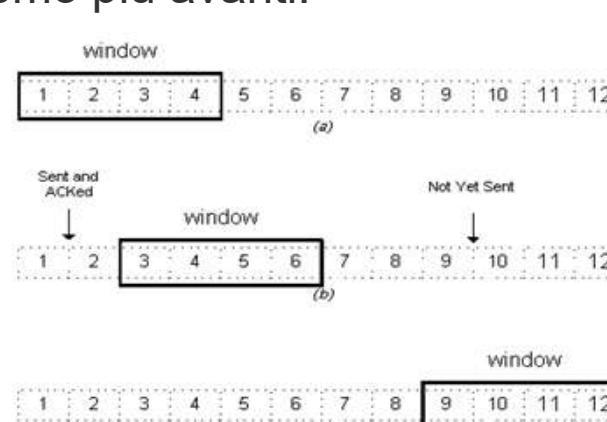
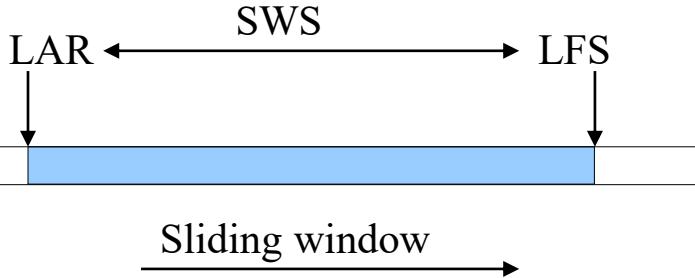
L’indice LFS (Last Frame Sent) contiene il numero dell’ultimo Frame inviato,

L’indice LAR (Last Ack Received) contiene l’indice dell’ultimo ACK ricevuto

Deve valere la regola $LFS - LAR \leq SWS$.

Il destinatario può comunicare al mittente la **finestra del destinatario**, che specifica il numero di dati che il destinatario può ricevere in quel momento. In genere corrisponde allo **spazio libero nel buffer del ricevente**.

La **finestra utilizzata** del mittente non può superare la finestra del destinatario ma può essere ridotta per altri motivi che vedremo più avanti.

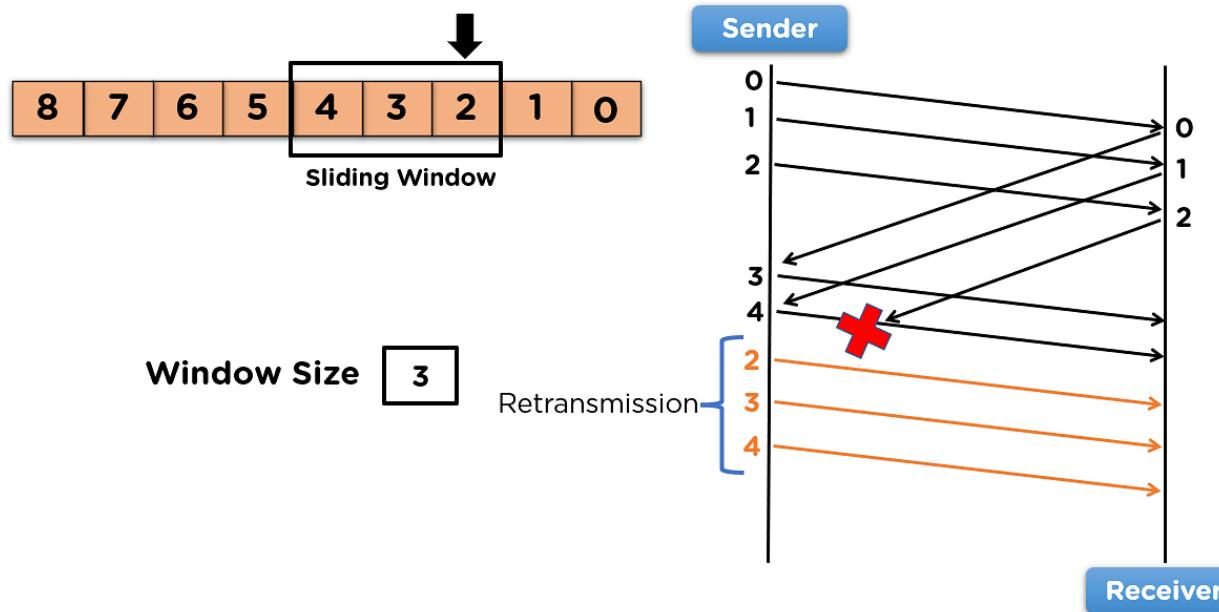


Protocollo per il recupero di errori Go-Back-N ARQ

Con il protocollo Sliding window può succedere che scada il timer di frame (errato o perduto) quando diversi altri Frame successivi sono stati consegnati correttamente. Se al destinatario arrivano Frame fuori ordine (successivi a Frame non ancora riscontrati) li scarta.

Solo i Frame arrivati correttamente (senza errori e in ordine) vengono mantenuti nel buffer del destinatario fino a quando l'applicazione non li gestisce.

Quando scade il timer del Frame errato/perduto il mittente torna indietro e ricomincia a spedire tutti i Frame a partire da quello errato (**Go-Back-N**)

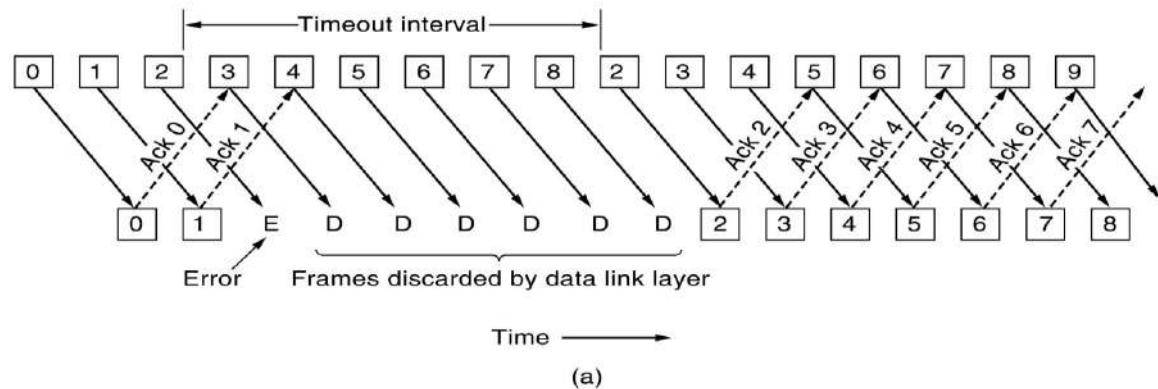


Protocollo per il recupero degli errori Ripetizione Selettiva

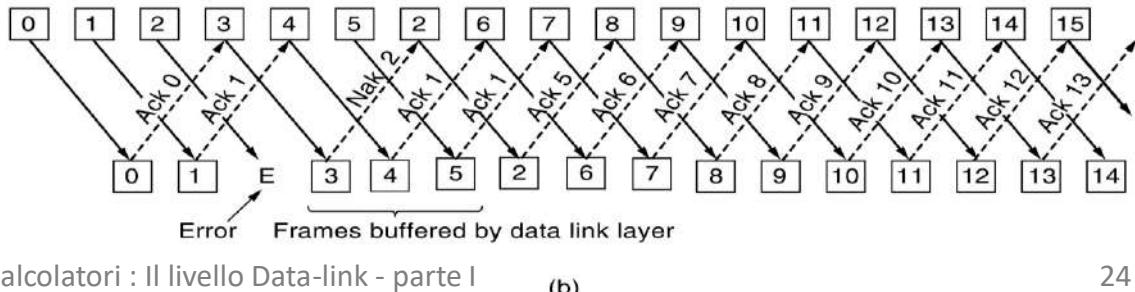
Con il protocollo a **Ripetizione Selettiva** i frame ricevuti correttamente, successivi a quello errato o perduto, vengono bufferizzati dal ricevente il quale sollecita il mittente al reinvio dei frame mancanti, tramite l'invio di un NACK.

- ▶ Il Mittente rispedisce il Frame richiesto
- ▶ Il Ricevente riscontra il Frame errato e tutti i frame successivi memorizzati nel buffer.
*Nota: Per il funzionamento di questo protocollo il destinatario deve gestire un **buffer** in cui vengono memorizzati i frame ricevuti.*

Esempio GO-BACK-N



Esempio Ripetizione Selettiva



Esempio di protocollo data-link: PPP

Evoluzione di HDLC per Internet. Utilizzato in ADSL.

NOTA: HDLC è un protocollo data link bit-oriented nato per comunicazioni punto-punto o multi-punto, con supporto sia alla modalità non connessa (unNumbered) che connessa.

PPP è protocollo Byte Oriented (Byte Stuffing) ed è definito in RFC 1661

(<http://www.ietf.org/rfc.html>)

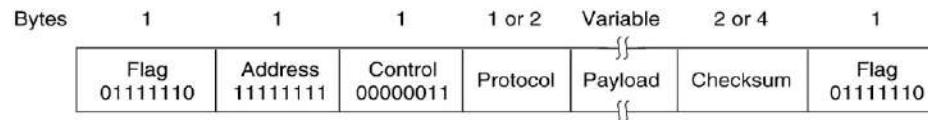
Supporto solo la modalità non connessa (solo UnNumbered)

Supporta vari protocolli dello strato rete (IP, AppleTalk, ...)

Gestisce protocolli ausiliari (LCP e NCP) per l'autenticazione, la configurazione degli indirizzi di rete (IP via DHCP), la concatenazione di diversi link.

I campi dell'header PPP

Il frame PPP aggiunge una intestazione di 6 (o 8) byte al payload, in cui vengono definiti alcuni campi originariamente ideati per HDLC.



Il framing è gestito con il Flag 01111110.

I campi **Address** e **Control** derivano da HDLC e in PPP hanno un valore fisso.

Il campo **Protocol** è stato aggiunto per supportare diversi protocolli a livello rete (IP, IPX, AppleTalk, ...) o protocolli ausiliari (LCP e NCP).

Nota: Il protocollo ausiliario **LCP** è utilizzato per configurare e verificare la connessione a livello data-link e consente di concatenare diversi link PPP.

Il protocollo ausiliario **NCP** serve per configurare i diversi protocolli di livello Network come DHCP per gli indirizzi di rete, per la compressione.

Payload: contiene un numero variabile di byte

MTU (Maximum Transfer Unit) è il payload massimo del protocollo. In PPP l'MTU standard è di 296 byte, ma può essere adattato (vedi PPPoA).

FCS (Gestione Errori): CRC-16, negoziabile fino a CRC-32. Il caso di errori il pacchetto viene scartato senza notifica. In caso di errori eccessivi viene abbattuta la connessione.

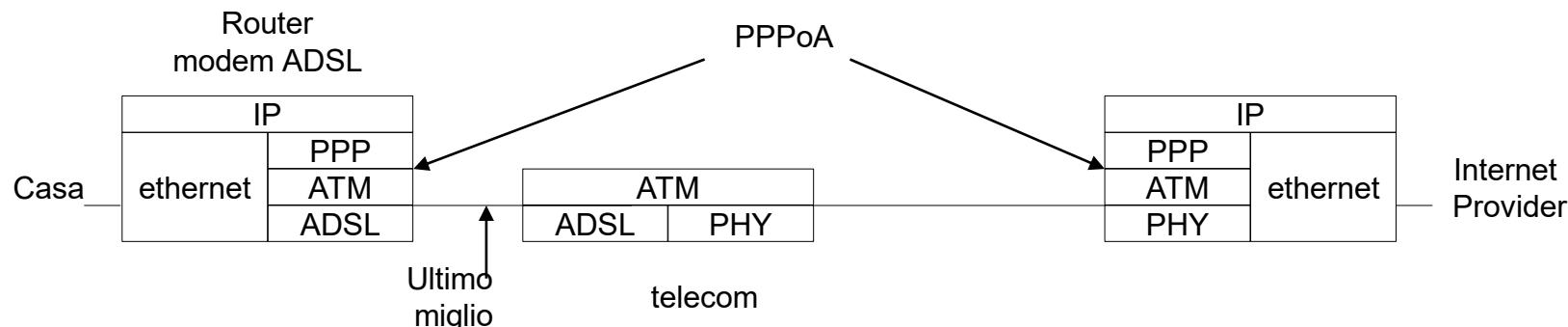
PPP over ATM (PPPoA)

Le telecom forniscono connessioni geografiche utilizzando le proprie reti commutate basate su tecnologia ATM.

ATM è una rete a commutazione di pacchetti, dette celle, di lunghezza fissa di 53 byte, di cui 48 di payload (vedi slide successiva).

Lo standard PPPoA (PPPooverATM) definisce le modalità per trasportare pacchetti PPP all'interno di celle ATM. PPP riceve frame Ethernet, per questo utilizza lo stesso MTU di 1500 byte.

Il frame PPP viene suddiviso in celle da 48 byte e riassemblato all'uscita della rete ATM.



Reti a circuito Virtuale: ATM

ATM (Asynchronous Transfer Mode) è una tecnologia, sviluppata da ITU-T (organismo internazionale che si occupa delle trasmissioni telefoniche) a partire dai primi anni 90, che realizza una infrastruttura di rete per trasmissioni a commutazione di pacchetto dedicata al sistema telefonico, ma con l'ambizione di essere utilizzato anche per le comunicazioni Internet.

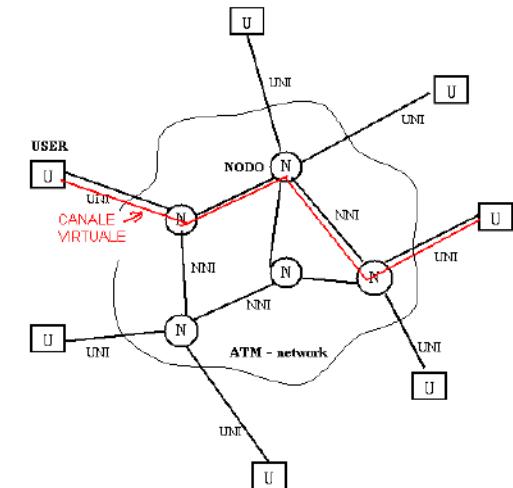
Nasce dal mondo della fonia, quindi i principi base dell'architettura sono adattati a questo tipo di esigenza:

- **commutazione di pacchetto a circuito virtuale** (simile alla commutazione di circuito)
- **Qualità del Servizio** (le trasmissioni telefoniche vengono integrate nelle trasmissioni dati, ma hanno diversi requisiti di qualità)
- **pacchetti (celle) di lunghezza fissa di 53 byte**
di cui 5 di intestazione e 48 di payload

Nota: ogni comunicazione telefonica trasporta la stessa quantità di dati per unità di tempo:

PCM genera 1Byte a 8KHz = 8 KB/s = 64 Kb/s.

Una cella trasporta $48/8K = 6$ ms di conversazione.



ATM non ha avuto successo al di fuori delle reti telefoniche,
se non per la realizzazione di reti WAN.

Viceversa la fonia sta diventando sempre più una applicazione di Internet (VoIP).

Reti Data-Link: Local Area Network

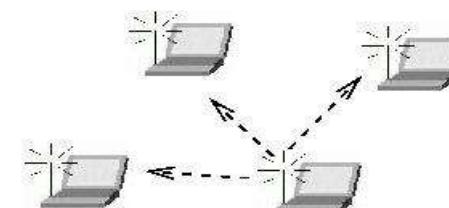
Un canale Multi-Accesso (o canale broadcast) è un canale condiviso per l'accesso diretto tra più terminali ed è il modo più semplice per realizzare una rete di calcolatori a livello Data-Link (LAN) che fanno parte dello stesso **dominio di broadcast** in cui i terminali possono scambiare tra loro messaggio unicast o broadcast.

Le problematiche delle reti LAN richiedono la definizione di un protocollo specifico per:

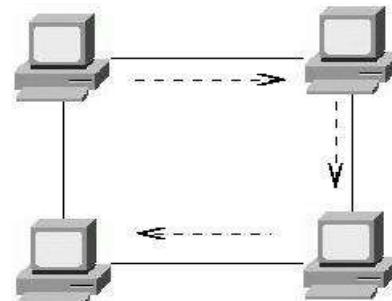
- Disciplinare l'accesso al canale (se fisicamente broadcast)
- Gestire gli indirizzamenti unicast, broadcast e multicast



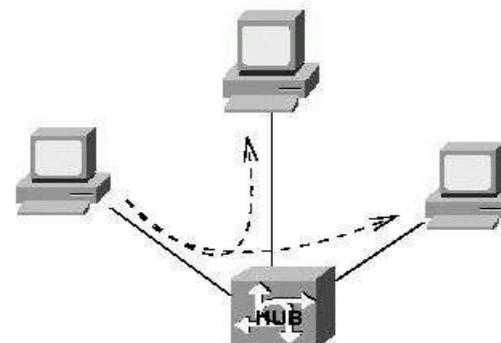
CANALE FISICAMENTE BROADCAST



CANALE FISICAMENTE BROADCAST



CANALE LOGICAMENTE BROADCAST



CANALE LOGICAMENTE BROADCAST

Canali Broadcast: Accesso al canale

Il canale può essere assegnato agli utenti in **modo statico o dinamico**.

Allocazione **statica** del canale

Si può realizzare con tecniche **FDM e TDM** suddividendo del capacità trasmittiva del canale in sotto-canali di numero e dimensione prestabilita

Se il numero di utenti è inferiore al numero di canali ho uno spreco di banda

Se il numero di utenti è superiore alcuni utenti non possono parlare, anche se altri stanno sottoutilizzando il proprio slot.

Questa tecnica è poco efficiente per le Reti Locali in cui gli utenti e le loro esigenze mutano rapidamente.

L'assegnazione del canale nelle principali tecnologie LAN è **dinamica**.

Assegnazione dinamica del canale : Accesso Multiplo

Un singolo canale viene condiviso da N stazioni, ma viene utilizzato solo da chi deve effettivamente inviare dati (assegnazione dinamica).

Nessuna stazione gestisce il canale, ma tutte le stazioni lo devono contendere.

Due possibili modalità di **tempo di trasmissione**:

- ▶ Tempo continuo: la trasmissione può iniziare in qualunque istante.
- ▶ Slotted: Il tempo è diviso in intervalli detti Slot. La trasmissione deve coincidere con l'inizio di un intervallo.

Collisioni: L'accesso a contesa implica che un Frame potrebbe entrare in collisione con un altro. In questo caso entrambi i Frame dovranno essere inviati nuovamente.

Un protocollo che gestisce i tempi di trasmissione e le eventuali collisioni è detto ad **Accesso Multiplo (Multiple Access – MA)**.

Verifica dell'occupazione del canale: in alcuni protocolli MA le stazioni verificano lo stato del canale (**Carrier Sense – CS**) prima di decidere se iniziare la trasmissione.

Alcuni protocolli verificano lo stato del canale anche durante la trasmissione per individuare rapidamente eventuali collisioni (**Collision Detection – CD**)

Protocolli ad Accesso Multiplo

I principali protocolli ad Accesso Multiplo (MA) sono:

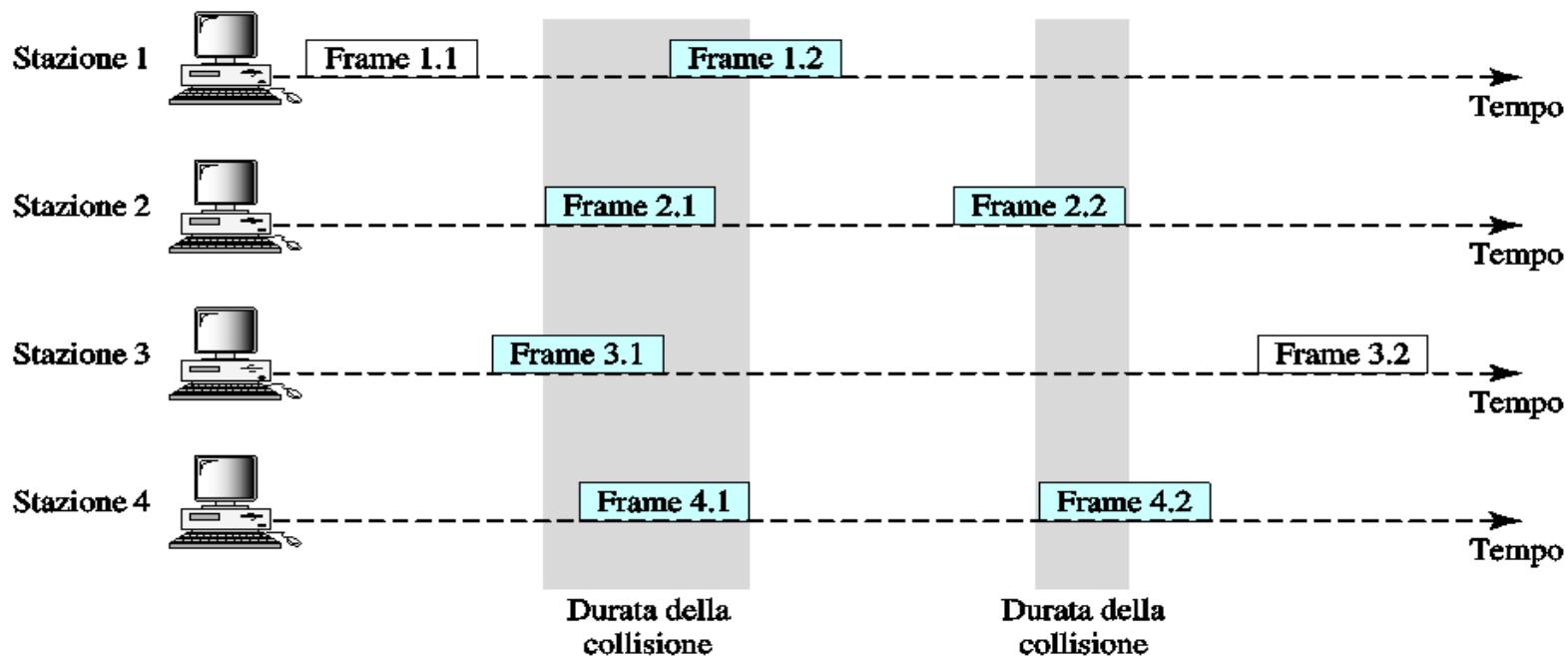
- ▶ MA puro (ALOHA) e slotted (Slotted ALOHA)
- ▶ CSMA persistente e non persistente
- ▶ CSMA/CD (802.3 - Ethernet su Rame o Fibra)
- ▶ CSMA/CA (802.11 - Ethernet Wireless)

ALOHA

ALOHA è il nome del primo protocollo Multiple Access (MA) ideato nei primi anni 70 dall'Università delle Hawaii per connettere via radio Honolulu con le altre isole dell'arcipelago.

ALOHA PURO (Norman Abramson 1970)

Ogni terminale invia i Frame senza accordo con gli altri (MA); l'assenza di conferma viene considerata una collisione con altri trasmettitori, per cui il Frame viene ritrasmesso dopo un **intervallo casuale** di tempo (tempo di backoff).



Algoritmo di Backoff

Il caso di collisione parte un algoritmo (detto algoritmo di Backoff) che determina un tempo di attesa prima di riprovare.

L'algoritmo di Backoff più utilizzato (Ethernet) è l'**esponenziale binario**:

- ▶ Dopo n collisioni consecutive si attende un numero di slot random tra 0 e $2^n - 1$.

Ethernet ammette un valore massimo di n=10

Dopo la prima collisione l'invio può avvenire dopo 0 slot (subito) con prob. 50% oppure dopo una attesa di 1 slot con prob. 50%.

Dopo la seconda collisione lo trasmissione avviene con probabilità al 25% per i 4 casi [0,1,2,3], e così via.

Nota: Se un host spedisce un Frame in un determinato slot, la probabilità di avere un collisione è data dalla somma delle probabilità di trasmissione degli altri host meno la probabilità del loro verificarsi in contemporanea (per non contarli doppi)

ALOHA: tempo di vulnerabilità

ALOHA PURO (Norman Abramson 1970)

Frame-Time T è il tempo necessario per trasmettere un frame

I Frame hanno lunghezza costante di L bits. $T=L/\text{bitrate}$

Anche con una sovrapposizione di un singolo bit entrambi i frame sono danneggiati.

Il **tempo di vulnerabilità** (intervallo di tempo in cui si può avere una collisione) è $2T$

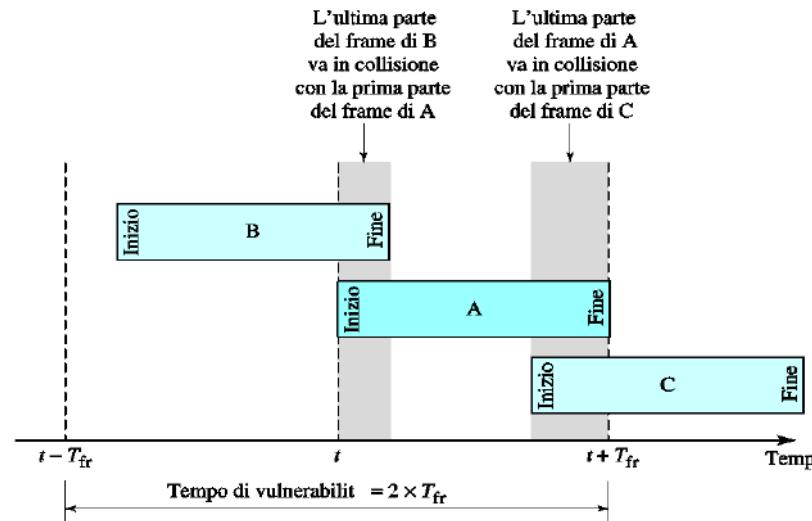
Con N frame generati mediamente nel tempo T :

- ▶ se $0 < N < 1$ ci aspettiamo un throughput ragionevole
- ▶ se $N > 1$ si va rapidamente alla paralisi

Per ogni collisione è necessario rispedire il Frame

G è il carico generato mediamente nel tempo T

$G = N + \text{frame rispetti}$



ALOHA: tempo di vulnerabilità

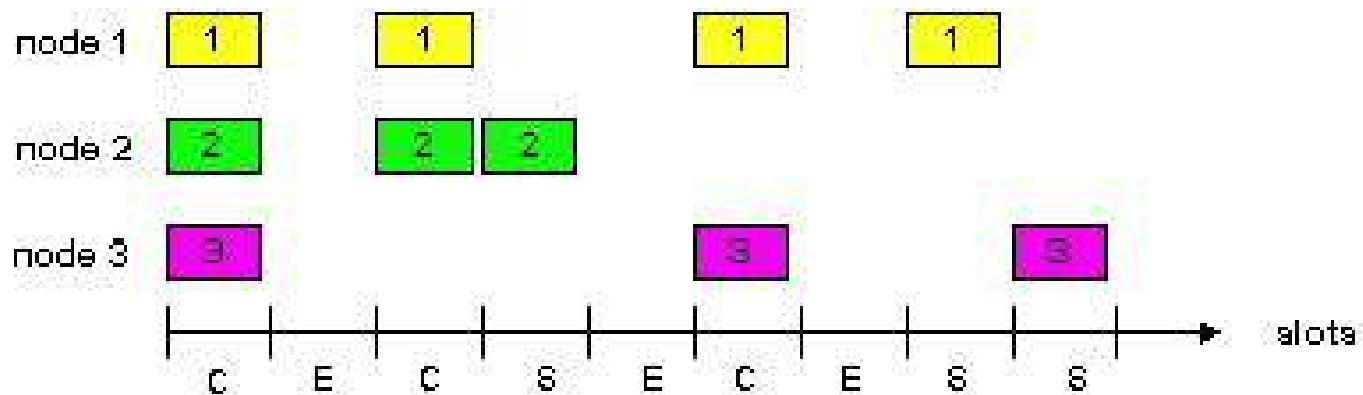
SLOTTED ALOHA (Roberts 1972)

Il tempo viene diviso in intervalli discreti

le trasmissioni possono iniziare solo all'inizio di un intervallo

Una speciale stazione emette un segnale all'inizio
di ogni intervallo per sincronizzare trasmettitori

Il tempo di vulnerabilità è T (dimezzato rispetto a Pure Aloha)



ALOHA Throughput

Qualunque sia il carico G che si presenta (pacchetti trasmessi nel tempo T), la capacità di trasporto S (Throughput) è G volte la probabilità P_0 (trasmissione con successo nel **tempo di vulnerabilità**): $S = G P_0$

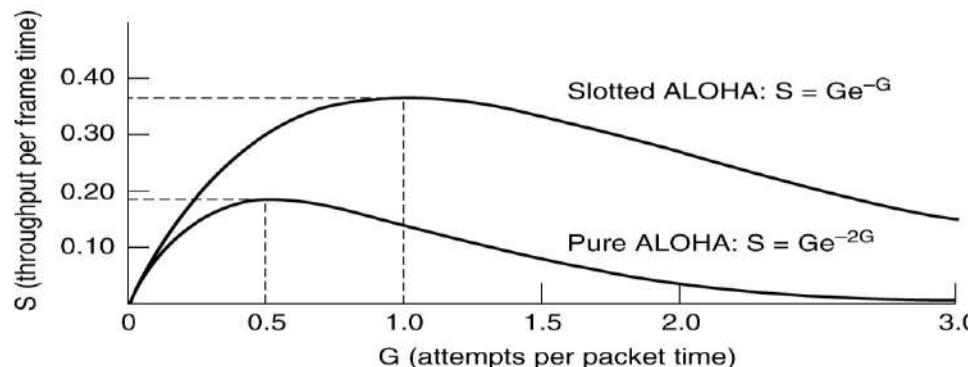
La probabilità che k frame siano generati durante il tempo T è dato dalla **distribuzione di Poisson**: $P[k] = G^k e^{-G} / k!$

ALOHA PURO: Per un periodo di vulnerabilità pari a $2T$ la probabilità che nessun altro frame venga generato durante il periodo di vulnerabilità:

$$P[0] = G^0 e^{-2G} / 0! = e^{-2G} \quad \text{Throughput } S = G e^{-2G} \quad \text{Max } S = 0.18 \text{ per } G = 0.5$$

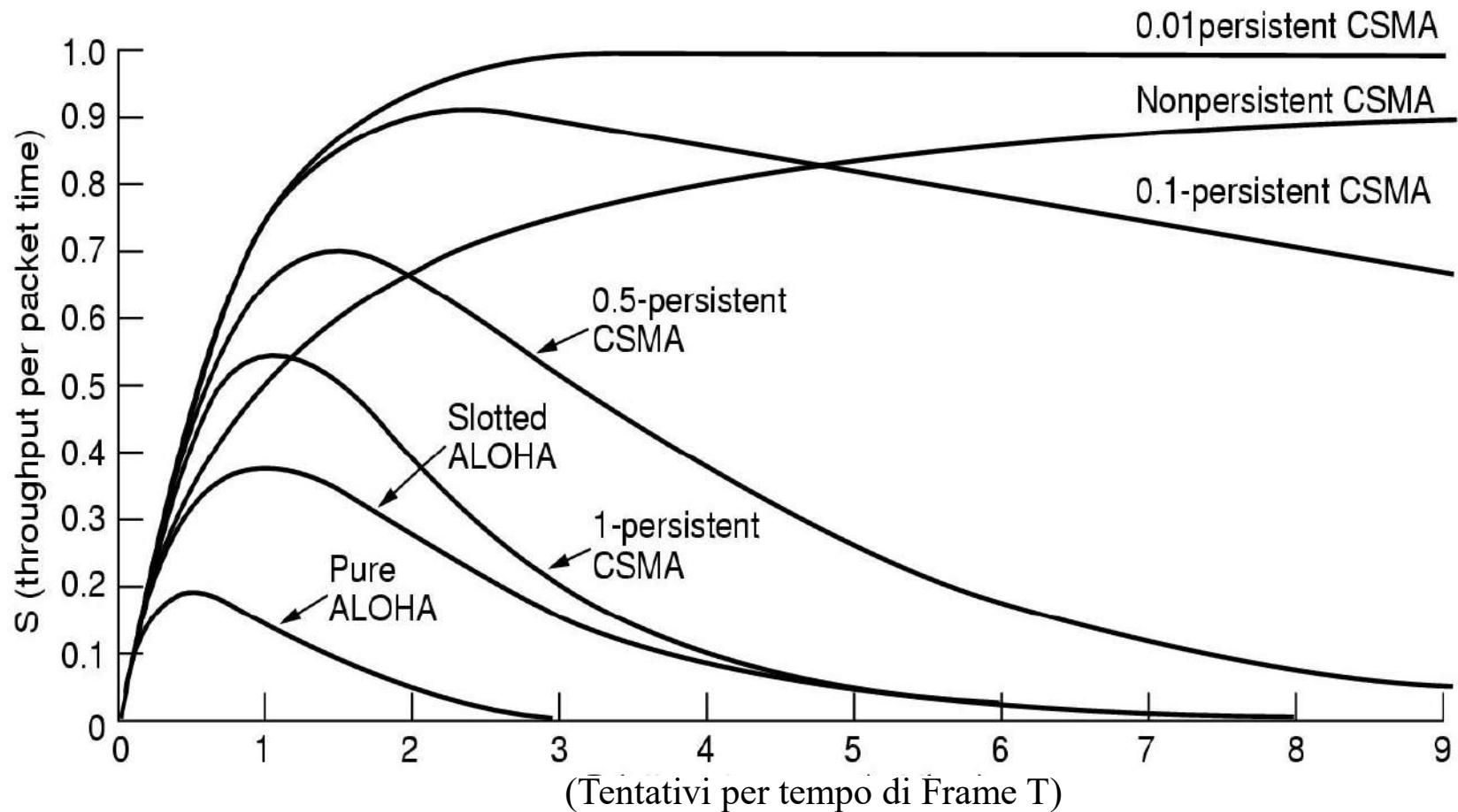
ALOHA SLOTTED: Il tempo di vulnerabilità è T (dimezzato rispetto a Pure Aloha)

$$\text{Throughput } S = G e^{-G} \quad \text{Max } S = 0.36 \text{ per } G = 1$$



Throughput

La figura mostra l'andamento del throughput mettendo a confronto diverse tecniche:



CSMA

CSMA (Carrier Sense Multiple Access)

migliora le prestazioni aggiungendo l'ascolto del canale: se il canale è occupato pospone la trasmissione

Il numero di collisione è molto ridotto (ma non azzerato) $G \approx N$

L'algoritmo che determina quando ritentare è fondamentale.

CSMA con persistenza

CSMA non persistente

se il canale è libero inizia la trasmissione altrimenti attende un tempo casuale prima di ritentare (anche in assenza di collisioni). Diminuisce la probabilità di collisione poiché è improbabile che 2 stazioni aspettino lo stesso tempo, ma aumenta il ritardo di trasmissione (anche in una rete con poco traffico).

CSMA 1-Persistente (utilizzato da Ethernet)

se il canale è libero inizia la trasmissione

altrimenti attende che si liberi prima di ritentare.

E' detto 1-persistent perché trasmette con probabilità 1 quando il canale è libero.

Problema: in caso di alto traffico è probabile che 2 nodi in attesa entrino in collisione.

CSMA p-persistente: Si applica ai canali divisi in intervalli temporali.

Se il canale è libero la trasmissione avviene con **probabilità p** e viene rimandata all'intervallo successivo con probabilità $1-p$.

Se anche questo è libero la trasmissione avviene con **probabilità p** e così via.

Se il canale è occupato si comporta come se ci fosse stata una collisione:

parte un algoritmo di Backoff (generalmente l'attesa è proporzionale al numero di collisioni consecutive)

Al crescere di p diminuisce il ritardo, ma aumenta la probabilità di collisione

CSMA/CD

CSMA/CD (Carrier Sense Multiple Access - Collision Detect)

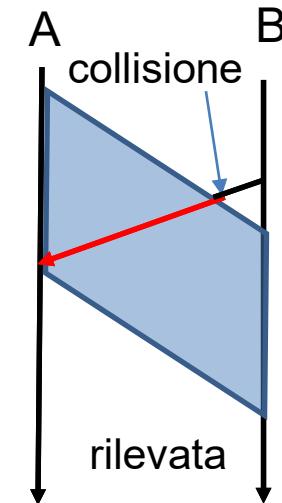
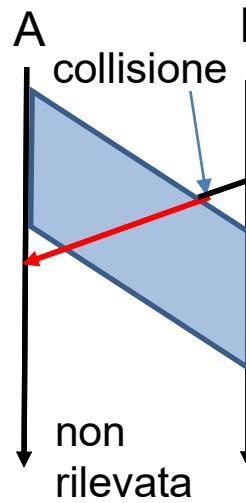
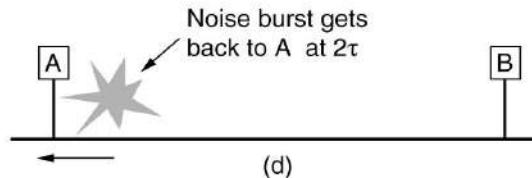
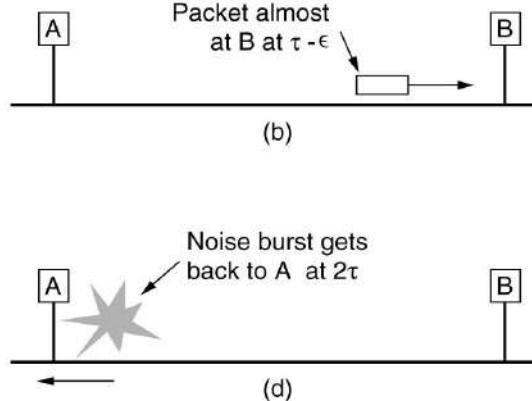
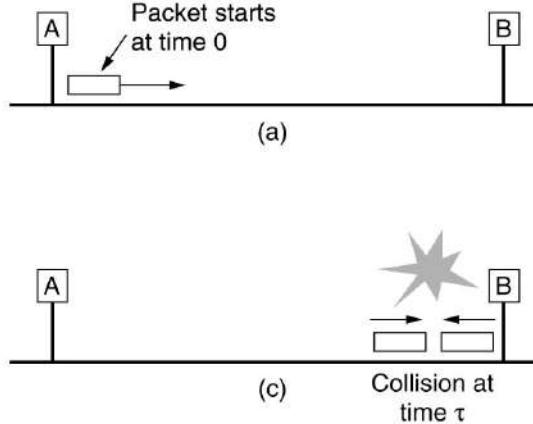
Chi spedisce rimane in ascolto del canale anche durante la trasmissione.

Vantaggi:

- in caso di collisione si interrompe la trasmissione → si riduce il tempo di vulnerabilità
- il mittente capisce se il Frame è stato inviato correttamente (senza collisioni)

Se T_{pr} è il tempo di propagazione del cavo, il massimo ritardo nell'individuare una collisione è $2T_{pr}$ (supponendo che il secondo nodo all'altro estremo inizi la trasmissione un attimo prima di ricevere il pacchetto)

Per individuare con certezza una collisione è quindi necessario che il frame abbia un tempo di trasmissione $T_{tr} \geq 2T_{pr}$



Dominio di Collisioni e dominio di Broadcast

L'insieme dei nodi che concorrono per accedere allo stesso mezzo trasmissivo costituisce un **Dominio di Collisione** (Collision Domain).

Il **dominio di Broadcast** è l'insieme dei nodi che possono comunicare direttamente, senza dover risalire al livello rete.

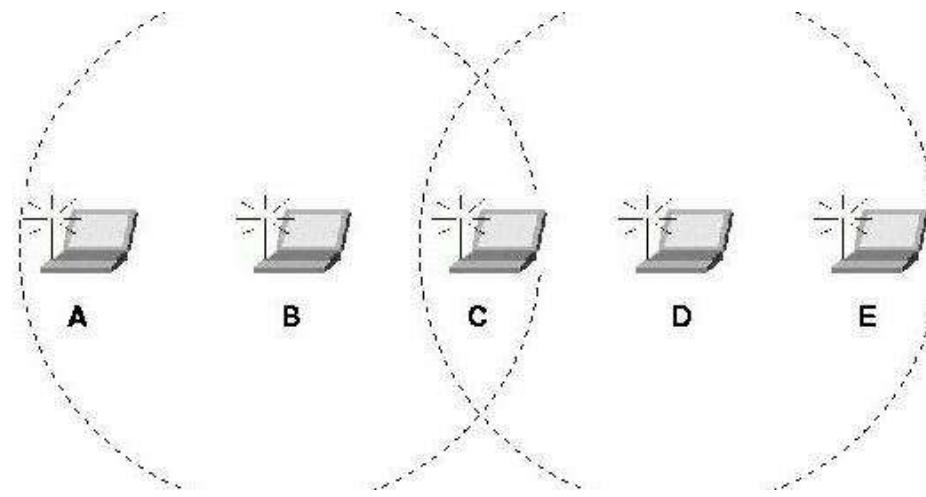
I due domini possono non coincidere per effetto di apparati di rete (Bridge) che separano i domini di collisione ma non i domini di broadcast.

Protocolli LAN Wireless

Nelle reti Wireless il dominio di collisione non è nettamente definito come nelle reti wired:

Problema del nodo nascosto: B trasmette a C. D non sente il segnale di B e trasmette contemporaneamente a B creando collisione non rilevata da B.

Problema del nodo esposto: B trasmette ad A. C vorrebbe trasmettere a D ma non lo fa perché crede erroneamente di creare una collisione.



La soluzione consiste nell'evitare le collisioni (Collision Avoidance) attraverso un opportuno protocollo (CSMA/CA) .

Protocolli LAN Wireless: CSMA/CA

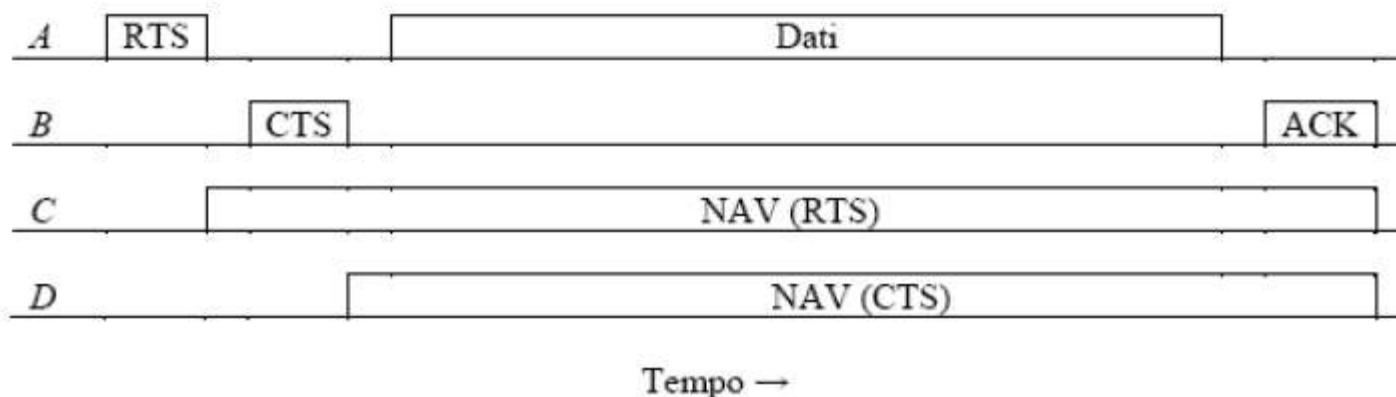
Nel protocollo **CSMA/CA** (CSMA with Collision Avoidance) il trasmettitore incita il ricevitore (con un Frame RTS) a trasmettere un piccolo Frame (CTS) il modo che le stazioni che si trovano alla sua portata evitino di inviare dati. Sia RTS che CTS contengono il tempo necessario per la trasmissione.

Le stazioni che ricevono RTS e CTS attivano un Carrier Sense Virtuale detto NAV (Network Allocation Vector) che è un contatore che viene decrementato e rappresenta il tempo in cui devono considerare indisponibile in canale.

Il ricevitore invia un pacchetto ACK dopo ogni Frame ricevuto con successo.

Eventuali collisioni di pacchetti RTS sono comunque possibili e sono gestite con il protocollo CSMA.

Questo protocollo è utilizzato nelle reti WiFi (IEEE 802.11) e WiMax (802.16)



Protocolli LAN Wireless: CSMA/CA

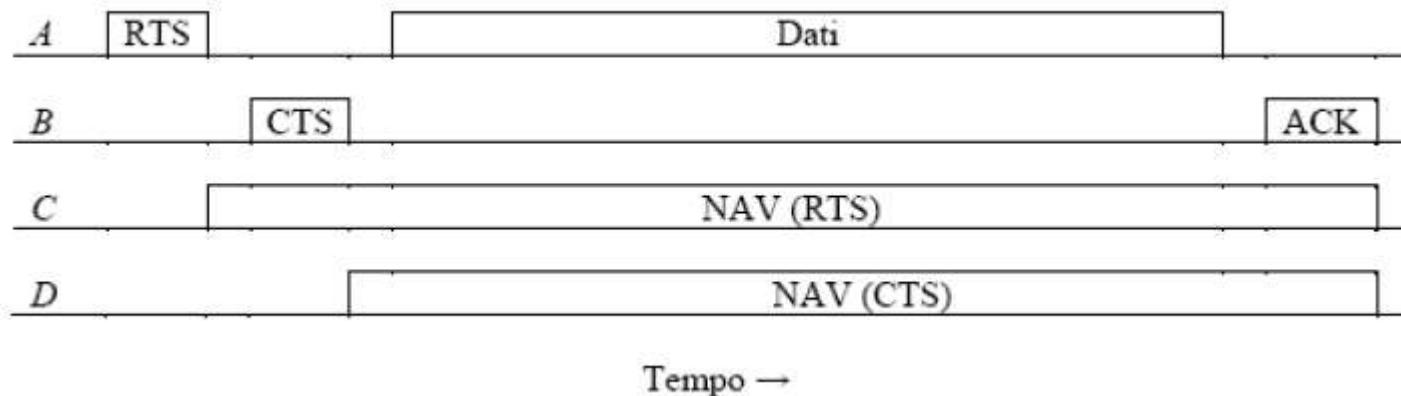
La stazione che deve trasmettere (A) valuta se il mezzo è libero consultando sia NAV che il mezzo reale.

Se il canale è considerato libero emette un RTS altrimenti avvia la procedura di Backoff Esponenziale Binario.

Quando il ricevente (B) riceve l'RTS risponde con un CTS.

Se A non riceve il CTS di B avvia la procedura di Backoff con tempo raddoppiato, altrimenti inizia la trasmissione.

Se dopo l'invio del dato A non riceve un ACK entro un tempo stabilito deve ripetere la procedura.





UNIVERSITÀ
DI PARMA

DIPARTIMENTO DI SCIENZE MATEMATICHE, FISICHE ED INFORMATICHE
Corso di Laurea in Informatica

Il Livello Data-Link

Parte II : Ethernet

RETI DI CALCOLATORI - a.a. 2023/2024

Roberto Alfieri

Il Livello Data-Link: sommario

PARTE I

- ▶ Scopi del livello Data-Link
- ▶ Framing, Rilevazione e correzione degli errori, controllo di flusso
- ▶ Protocolli per reti Punto-punto: PPP.
- ▶ Protocolli per reti MultiAccesso: Aloha, CSMA, CSMA/CD, CSMA/CA

PARTE II

- ▶ Gli standard IEEE802
- ▶ Ethernet: Sottoliv. MAC e LLC, tecnologie Ethernet, il Frame, Repeater, Switch, Bridge
- ▶ Spanning Tree Protocol.
- ▶ Lan Virtuali

PARTE III

- ▶ Lan Wireless

RIFERIMENTI

- ▶ *Reti di Calcolatori, A. Tanenbaum, ed. Pearson*
- ▶ *Reti di calcolatori e Internet, Forouzan , Ed. McGraw-Hill*

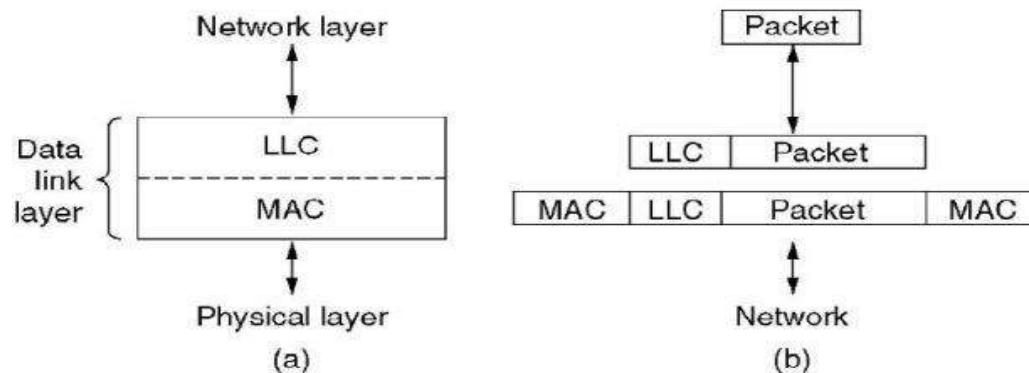
Standard IEEE 802

Molte delle architetture per reti locali, personali e metropolitane sono state standardizzate sotto il nome di IEEE 802.

IEEE (Institute of Electrical and Electronics Engineers) è una grande organizzazione professionale con scopo di ricerche e applicazioni in vari campi ingegneristici tra cui l'elettronica e l'informatica. Una delle attività è la definizione di Standard in questi campi.

Il numero "802" è il primo numero libero negli standard IEEE al momento della formazione del comitato, creato nel 1980 per la standardizzazione delle reti locali.

IEEE 802 separa le funzionalità del livello Data-Link in 2 sottolivelli distinti: LLC e MAC.

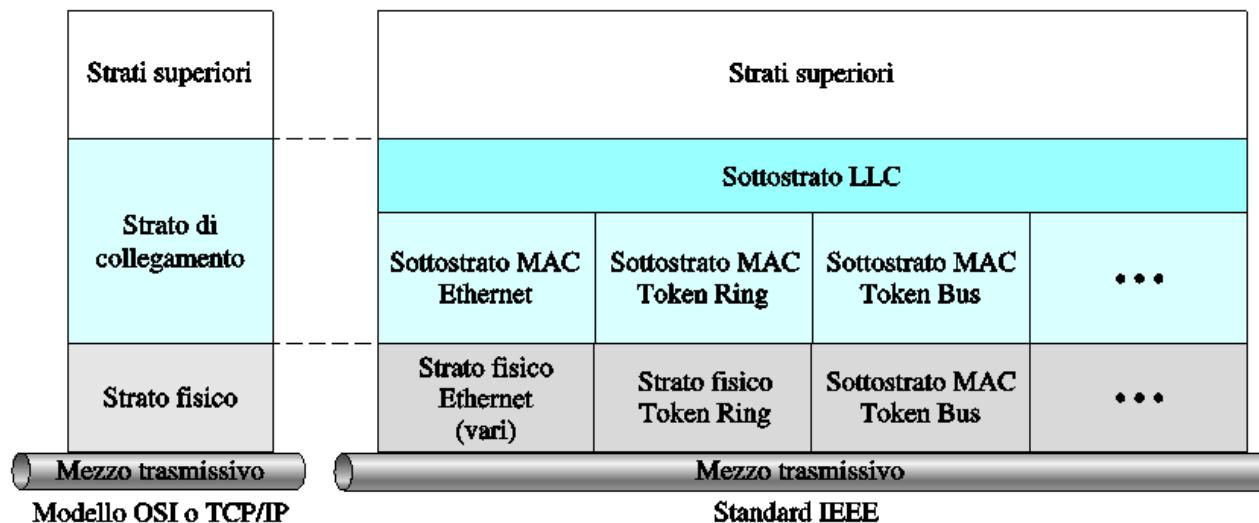


Sottolivelli MAC e LLC

Il sottolivello **MAC** (Medium Access Control) gestisce l'accesso al mezzo mediante diversi possibili protocolli di Accesso Multiplo quali ad esempio:
802.3 (Ethernet), 802.5 (Token Ring), 802.11 (LAN Wireless) 802.15 (Bluetooth),
802.16 (WiMax), ecc

Il sottolivello **LLC** (Logical Link Control, 802.2) è **opzionale** e aggiunge una intestazione contenente la codifica del protocollo che ha generato il frame e a cui è destinato, il numero di sequenza e Acknowledge (Ack), consentendo di operare in 3 modalità:

- datagramma inaffidabile,
- datagramma con Ack ,
- servizio affidabile orientato alle connessioni.

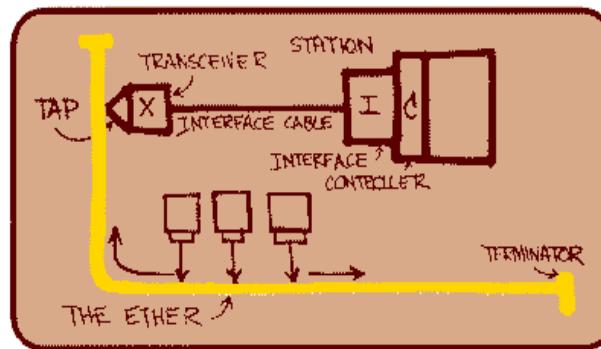


Media Access Control: Ethernet

Tecnologia dominante per le LAN.

Ideata a metà degli anni 70 da Bob Metcalfe (dottorando di Harvard) partendo dal lavoro di Abramson (AlohaNet). Quando Metcalfe viene assunto da Xerox progetta e costruisce la prima rete locale, chiamata Ethernet. Nel 1978 Xerox, assieme a DEC e Intel, mise a punto uno standard per la versione a 10Mbps.

Un segmento Ethernet si realizza con un cavo coassiale di lunghezza fino a 500 mt, un cavo simile a quello usato per la TV ma con una impedenza di 50 Ohm anziché 75. I **terminatori** connessi alla fine del segmento assorbono il segnale e impediscono che rimbalzi e interferisca con altri segnali.



Ethernet/IEEE802.3

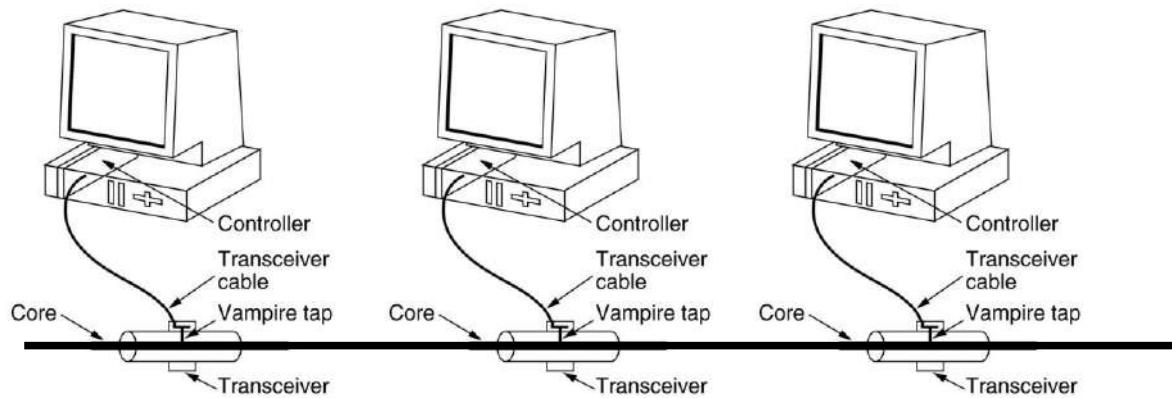
Nella versione Ethernet 2.0 (1980) la velocità era di 10Mb/s su cavo coassiale.

Nel 1983, con piccole modifiche, Ethernet venne recepita da IEEE con il nome di 802.3.

Servizio di invio di **Frame senza connessione e senza riscontro**.

Il servizio è **inaffidabile (Best effort)** : non dà alcuna garanzia dell'effettiva consegna dei dati né tantomeno livelli di qualità o priorità garantiti. Se il ricevente rivela un errore di trasmissione scarta il pacchetto.

Eventuali servizi di connessione e ritrasmissione sono demandati ai livelli superiori. Questo rende Ethernet più snello e veloce.



Ethernet/IEEE802.3: il protocollo

Protocollo di accesso al mezzo:

CSMA/CD 1-persistente con algoritmo di Backoff esponenziale binario

1) se il canale è libero trasmette il Frame

1.1) Mentre trasmette ascolta il canale; se durante la trasmissione non rivela segnali diversi considera il Frame spedito.

1.2) Se durante la trasmissione rivela segnali diversi arresta immediatamente la trasmissione e invia un breve segnale di disturbo (jamming sequence) di 32 bit, quindi entra nella fase di attesa esponenziale prima di riprovare (random tra 2 -1 slots dopo i collisioni)

Vengono inviati almeno 96 bit: 64 di preambolo + 32 di disturbo. Il tempo di 96 bit rappresenta il tempo minimo di attesa tra un due Frame (InterFrame Gap – IFG).

1.3) Dopo 10 collisioni l'intervallo rimane congelato ($2^{10}-1 = 1023$ slots)

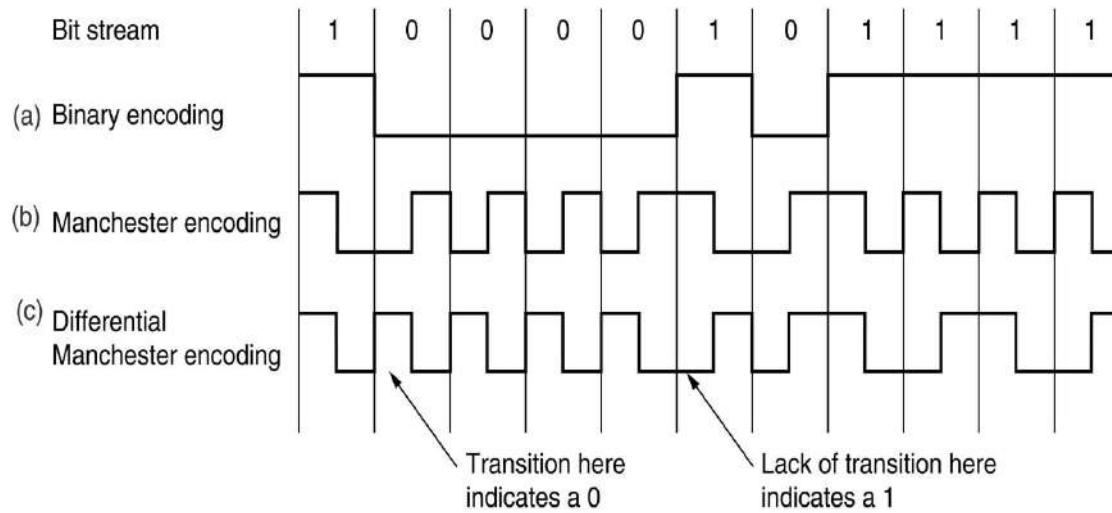
1.4) Dopo 16 collisioni il controller rinuncia e segnala un errore.

2) se il canale è occupato aspetta fino a quando la linea diventa inattiva poi trasmette immediatamente (1-persistente)

Ethernet: codifica Manchester

Codifica **Manchester** (usata da 802.3 - Ethernet) opera in banda Base del livello fisico. Occupa una banda doppia rispetto alla codifica binaria (per avere 10Mb/s occorre una clock a 20Mb/s)

Transizione alto-basso=1, transizione basso alto =0. Alto = 0.85V, Basso=-0.85V



La codifica **Manchester Differenziale** (usata da 802.5 – Token Ring) è più complessa da realizzare ma offre una maggiore immunità al rumore.

Ethernet: Formato del Frame

Il campo **Preamble** contiene 7 (o 8) sequenze 10101010 ovvero un'onda quadra di circa 5 ms che consente al ricevitore di sincronizzarsi.

SoF (Start of Frame) indica l'inizio di un Frame. Contiene la 10101011; gli ultimi 2 bit a 1 indicano che il prossimo bit sarà il primo del Frame.

Il campo **Type/Length** contiene il codice del protocollo di livello superiore che ha generato il Frame. Nello standard IEEE802.3 il campo Type è sostituito dal campo Length (numero di byte del campo DATA). I 2 Standard possono convivere.

Ad esempio: 0000-05DC=DataLength 0800=IP 0806=ARP 809B=AppleTalk

Per il formato degli indirizzi (**Dest. Add. e Source add.**) vedi slide successiva.

Il campo **DATA** contiene le PDU dei livelli superiori.

Il campo **FCS** contiene il valore di CRC-32 (Cyclic Redundance Code).

Il marcitore di fine Frame manca e, tale ruolo, è assunto dall'Inter Frame Gap (IFG) il cui valore minimo è di 96 bit (10Mbps->9.6us 100Mbps->960ns 1Gbps->96ns).

	Bytes	8	6	6	2	0-1500	0-46	4	
Ethernet	(a)	Preamble	Destination address	Source address	Type	Data »	Pad	Check-sum	
IEEE802.3	(b)	Preamble	S o F	Destination address	Source address	Length	Data »	Pad	Check-sum

Indirizzi Ethernet

Gli indirizzi Ethernet sono di 6 byte ($2^{48} = 281.474.976.710.656$ possibili indirizzi)

- ▶ Esempio: 08 00 20 00 70 DF

L'indirizzo viene scritto nel firmware del controller di rete dal produttore ed è unico
I primi 3 byte identificano il Vendor.

- ▶ Ad esempio: 00000C=Cisco 00AA00=Intel 08005A=IBM 080020=SUN ecc
- ▶ Per una lista completa: <http://standards-oui.ieee.org/oui/oui.txt>

Negli indirizzi **Unicast** il primo byte è pari (LSB=0)

Negli indirizzi **Multicast** il primo byte è dispari (LSB=1)

- ▶ Ad esempio: 01:80:C2:00:00:00 è l'indirizzo multicast per l'algoritmo STP

L'indirizzo di **Broadcast** è ff-ff-ff-ff-ff-ff

Un adattatore Ethernet accetta i pacchetti nei seguenti casi:

- ▶ Tutti i Frame **Broadcast**
- ▶ I frame **Unicast** indirizzati all'interfaccia
- ▶ I frame di un particolare indirizzo **Multicast** se l'interfaccia è stata configurata opportunamente
- ▶ Tutti i Frame se l'interfaccia è configurata in modo **promiscuo**

Ethernet: Dimensione del Frame e slot time

La **massima dimensione** di dati trasportabili da un servizio Data-Link è detta **MTU** (Maximum Transfer Unit). L'MTU di Ethernet è 1500 Byte.

La **minima dimensione** di un frame è di 64 byte (46 payload + 18 header) = 512 bit

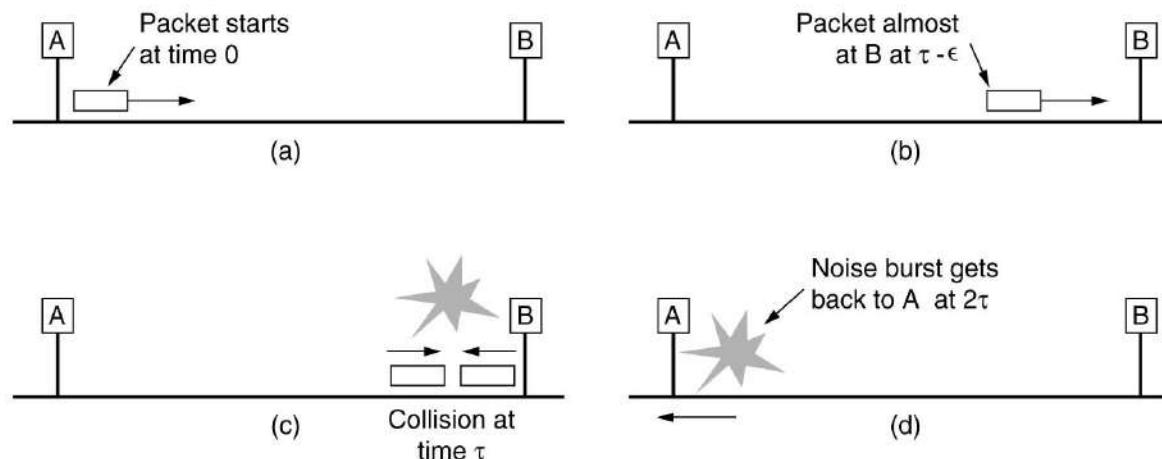
A 10 Mbps: 1 bit dura $1/10\text{Mbps} = 100\text{ns}$ e misura $100\text{ns} \times 2 \cdot 10^8 \text{ m/s} = 20 \text{ metri}$

Per 512 bit (frame minimo) abbiamo $L=10\text{Km}$

La **durata minima** di un frame è quindi di $t=51.2 \mu\text{s}$, che rappresenta lo **Slot Time**

Questo è anche il tempo utile per rilevare una collisione, per cui in questo tempo il Frame deve avere percorso in andata e ritorno l'intera rete.

La lunghezza massima teorica della rete deve essere al più la metà del Frame più piccolo, ovvero circa 5Km. Nella realtà occorre tener conto dell'attraversamento di eventuali apparati di rete (repeater, switch, ...) che rallentano il viaggio del Frame e quindi riducono la dimensione massima della rete, che viene stabilita in 2500 metri.



Evoluzione dello standard: Ethernet

Tecnologia in evoluzione retro-compatibile:

1980: Ethernet

Digital Intel e Xerox rilasciano le specifiche di Ethernet 2.0 che rappresenta lo standard “De Facto” per le reti locali. Topologia a Bus su cavo coassiale a 10Mb/s

Non ha bisogno di apparati di rete, ma è impossibile la gestione centralizzata.

1983: IEEE802.3

Ethernet diventa un standard “Ufficiale” con il nome di IEEE802.3

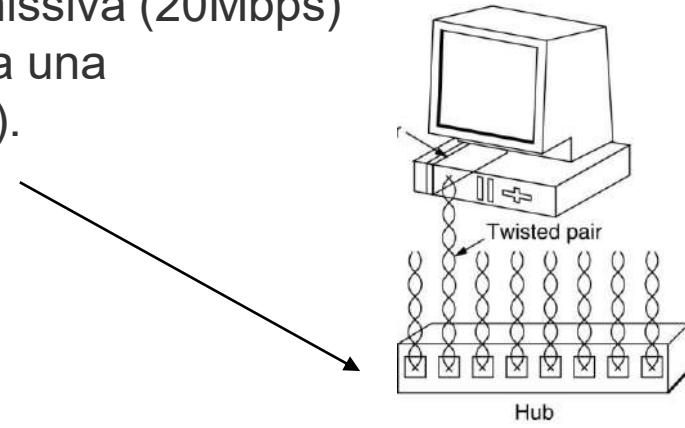
10Base-T utilizza la codifica Manchester con un clock a 20MHz su doppino (cat 3 o 5)

Si usano 2 doppini per avere una connessione **full duplex**.

La connessione full-duplex raddoppia la capacità trasmissiva (20Mbps)

Il doppino è un canale punto-punto, quindi è necessaria una topologia a stella con un apparato concentratore (HUB).

Name	Cable	Max. seg.	Nodes/seg.	Advantages
10Base5	Thick coax	500 m	100	Original cable; now obsolete
10Base2	Thin coax	185 m	30	No hub needed
10Base-T	Twisted pair	100 m	1024	Cheapest system
10Base-F	Fiber optics	2000 m	1024	Best between buildings



Evoluzione dello standard: Fast Ethernet

1995: IEEE802.3u (100 Mb/s - Fast Ethernet)

Stesso formato di pacchetto, interfacce e regole di 10baseT, ma 10 volte più veloce:
Un bit a 100Mb/s : $t=1/100M = 10^{-8} \text{ s} = 10\mu\text{s}$ $I = 2 \times 10^8 \times 10^{-8} = 2 \text{ m}$

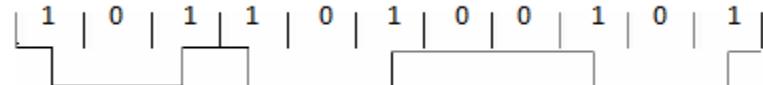
Varianti:

100BaseT4 usa un clock di 25MHz (rispetto ai 20MHz di Ethernet) con codifica 8B/6T (8bit con sequenze di 6 simboli a 3 stati) utilizzando 4 coppie di doppino di cat. 3.

100BaseTX: utilizza la codifica NRZ-I (ogni 1 comporta una transizione) di tipo [4B/5B](#), garantendo almeno una transizione per gruppo. Per avere 100Mb/s occorre un clock di 125MHz, che è supportato dal doppino di cat.5.

100baseFX: 2 fibre multimodo 62.5/125 - Max 2Km

Anche in questo caso si utilizza NRZ-I con la codifica 4B/5B



Name	Cable	Max. segment	Advantages
100Base-T4	Twisted pair	100 m	Uses category 3 UTP
100Base-TX	Twisted pair	100 m	Full duplex at 100 Mbps (Cat 5 UTP)
100Base-FX	Fiber optics	2000 m	Full duplex at 100 Mbps; long runs

Evoluzione dello standard: Gigabit Ethernet

1999: IEEE802.3z - 1 Gbps

Name	Cable	Max. segment	Advantages
1000Base-SX	Fiber optics	550 m	Multimode fiber (50, 62.5 microns)
1000Base-LX	Fiber optics	5000 m	Single (10 μ) or multimode (50, 62.5 μ)
1000Base-CX	2 Pairs of STP	25 m	Shielded twisted pair
1000Base-T	4 Pairs of UTP	100 m	Standard category 5 UTP

Standard:

1000baseCX Poiché il frame minimo (di 64 byte) può essere trasmesso 100 volte più velocemente rispetto a Ethernet standard la massima distanza con CSMA/CD diventerebbe 25metri (praticamente inutilizzato)

Esistono 7 standard per trasmissione su fibra, tra cui:

- ▶ **1000baseSX** fibra ottica multi-modale (fino a 550 metri)
- ▶ **1000baseLX** fibra ottica mono-modale (fino a 5Km)
- ▶ **1000baseBX10** singola fibra ottica mono-modale con 2 differenti portanti nelle 2 direzioni (fino a 10Km)

1000baseT usa doppino almeno di cat 5 (max 100mt)

Si usano tutte le 4 coppie. Ogni coppia trasporta 250Mbps. Il baud-rate è ancora 125Mbaud (come Fast Ethernet), ma la codifica PAM5 (modulazione di una ampiezza di una portante, usa 5 simboli con 2 bit per baud).

Nella modalità full duplex abbiamo 500Mbps per direzione, mentre nella modalità half duplex si può arrivare a 1 Gbps.

Gigabit Ethernet full duplex e half duplex

Modalità FullDuplex (modalità normale)

tutte le configurazioni Gigabit Ethernet sono Punto-Punto abbandonando il sistema MultiAccesso di Ethernet originale. Utilizza solo Ethernet Switch, che bufferizzano i frame in modo da eliminare le collisioni e il conseguente limite sulla durata minima del pacchetto.

La lunghezza massima del cavo è determinata dell'attenuazione del segnale.

Modalità HalfDuplex

Usato solo per mantenere la compatibilità con il protocollo CSMA/CD delle precedenti versioni.

Per risolvere il problema del rilevamento della collisione vengono introdotte 2 estensioni:

Carrier extension: Consente di aumentare il Padding, portando il Frame minimo a 512 byte (4096 bit, 4.1 μ sec)

Frame Bursting: Consente di aggregare più Frame in spedizione, in modo da raggiungere 512 byte (max 64Kbit)

Nota: Per ridurre l'overhead alcuni vendor supportano i “**Jumbo frame**” consentendo un MTU di 9 KB, ma questo non è parte dello standard IEEE.

Evoluzione dello standard: 10GbE

Standard rilasciato nel 2005 per fibra ottica (IEEE 802.3-2005); successivamente esteso al doppino (emendamento IEEE-802.3an).

E' solo switched full duplex quindi **CDMA/CD non è utilizzato**.

Codifica: si applica 64B/66B (per evitare lunghe sequenze di zeri o uni)

Cablaggio:

Fibra multi-modo 10GbaseSR (300m)

Fibra mono-modo 10GbaseLR (10Km) - 10GbaseER (40Km)

Doppino cat. 6A 10GbaseT (100m)

Le comunicazioni Ethernet a 10Gb/s sono state impiegate fino al 2009 praticamente solo per comunicazioni fra apparati di rete di backbone in quanto per poter alimentare un'interfaccia a 10Gb/s è necessario un bus di sistema sufficientemente veloce.

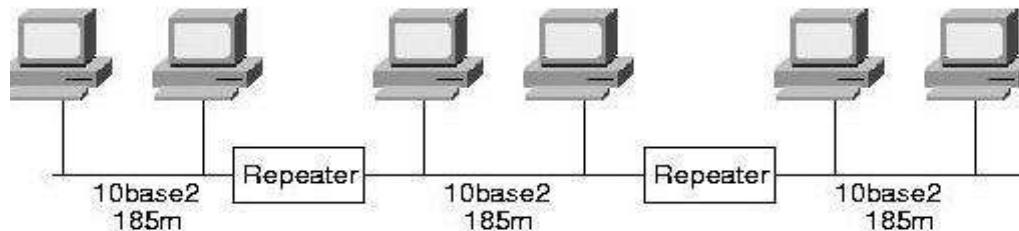
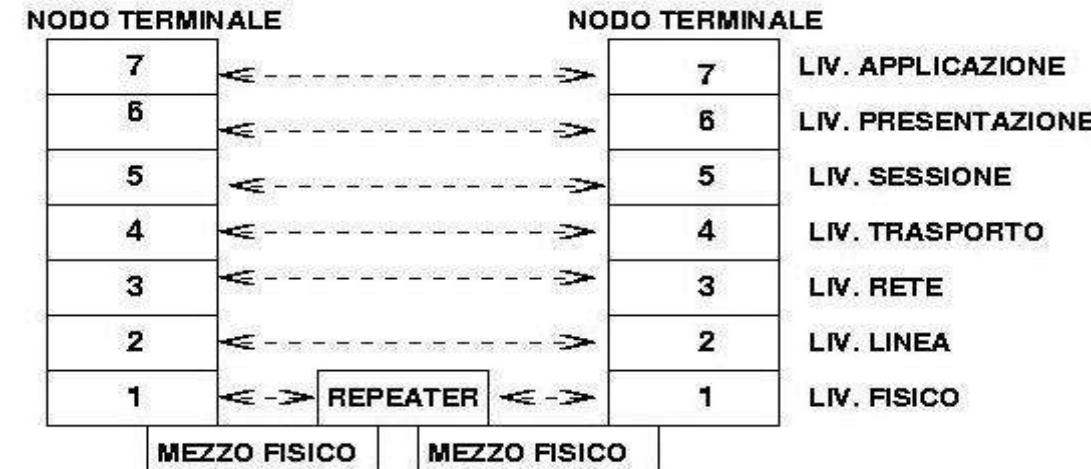
Riferimenti: http://it.wikipedia.org/wiki/10_gigabit_Ethernet

Ethernet Repeater

Il Repeater è un nodo di Transito che agisce a livello Fisico connettendo 2 o più segmenti Ethernet.

I bit ricevuti da una interfaccia vengono rigenerati e replicati sulle altre.

Ethernet stabilisce un massimo di 4 Repeater in una rete e una distanza max di 2.5 Km.

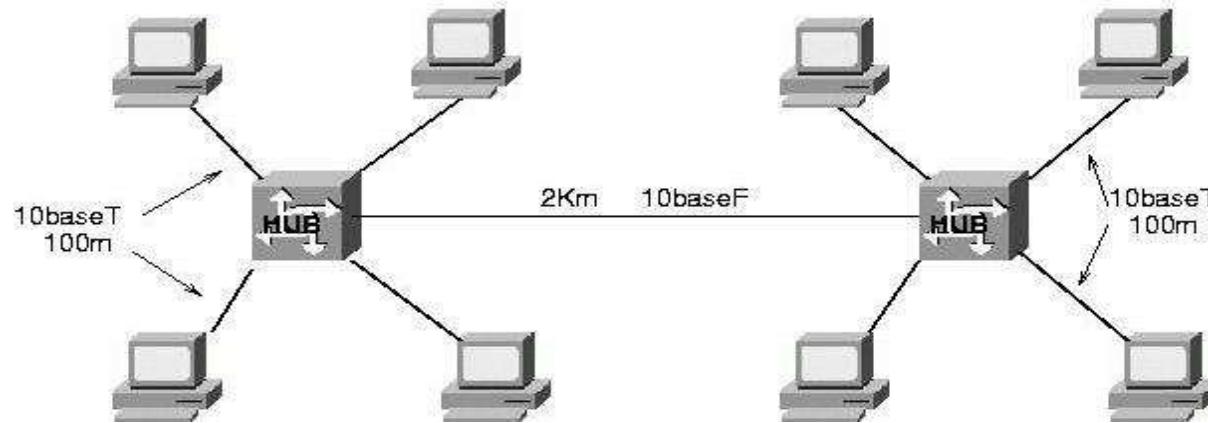


Ethernet Hub

Un Hub è un repeater con più di 2 interfacce Ethernet.

L'Hub consente di realizzare una rete Broadcast utilizzando canali punto-punto.

L'Hub estende sia il **Dominio di Collisione che il dominio di broadcast**.

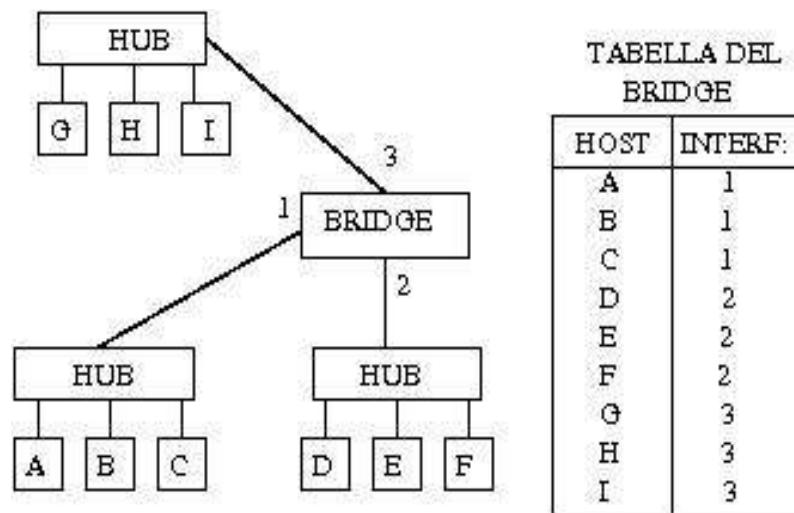


Commutazione nel livello data-link: il Bridge

Filtering: A differenza dell'HUB ritrasmette solamente i pacchetti che devono effettivamente transitare da una LAN ad un'altra e i broadcast.

In questo modo vengono separati i domini di collisione ma non il dominio di broadcast.

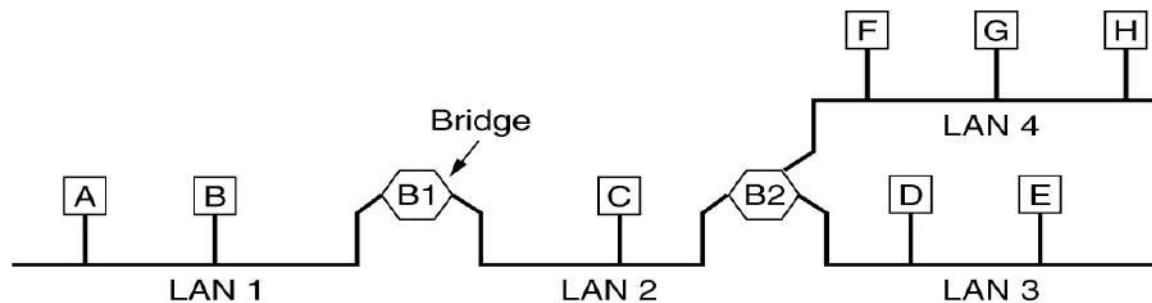
L'instradamento (forwarding) avviene in base a una **Tabella del Bridge** in cui sono indicati gli indirizzi MAC dietro ad ogni Interfaccia.



Bridge: autoapprendimento

La tabella del Bridge è costruita automaticamente in modo autonomo:

- 1) All'accensione del Bridge la Tabella è vuota.
- 2) Quando arriva un frame, l'indirizzo di mittente viene scritto nella tabella, associato all'interfaccia di provenienza e al tempo attuale. Se l'indirizzo di destinazione non è presente in tabella il frame viene inoltrato su tutte le interfacce tranne quella di provenienza (**flooding**), altrimenti viene inoltrata sulla sola interfaccia indicata in tabella.
- 3) Il Bridge cancella un indirizzo dalla Tabella se per un certo periodo di tempo (**tempo di invecchiamento**) non riceve alcun Frame con quell'indirizzo di provenienza.

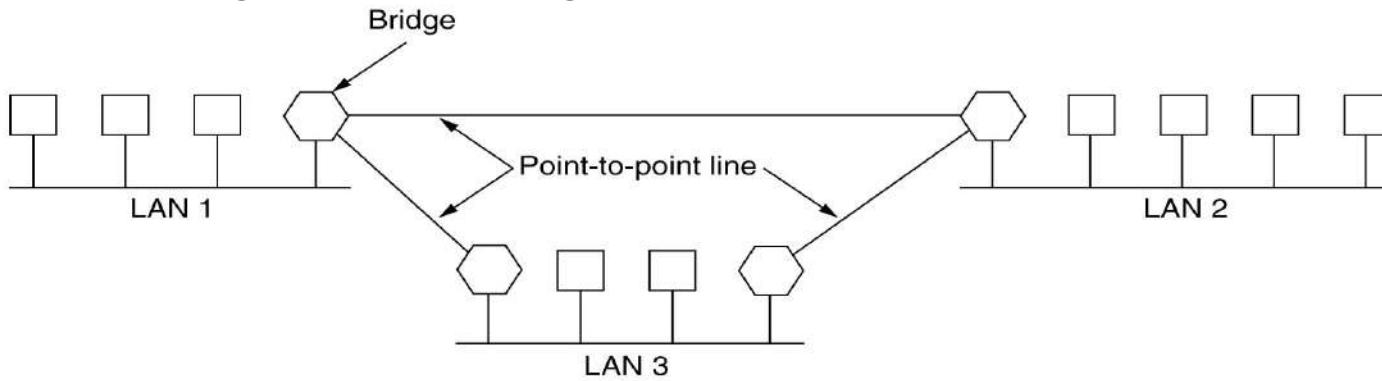


Bridge remoti

I bridge sono comunemente utilizzati per collegare 2 o più LAN distinte.

Se le LAN sono remote (in diverse edifici o diverse città) i Bridge vengono collegati mediante linee punto-punto su cui possiamo utilizzare protocolli punto-punto (come PPP) e incapsulare al suo interno il frame MAC, oppure possiamo usare la linea punto-punto direttamente con protocollo MAC.

In ogni caso il Bridge non cambia gli indirizzi fisici del frame.



Ogni LAN ha un proprio formato dei frame (es. Ethernet e Wifi nei router ADSL)

- Se l'MTU della LAN di destinazione è troppo piccolo i bridge devono eliminare il frame (la frammentazione è prevista solo a livello rete).
- Sicurezza: Se la LAN di provenienza è cifrata (es WiFi) il bridge deve essere in grado di decifrare prima di inoltrare.

Ethernet commutata: gli Switch

E' un commutatore di Frame a livello 2: i Frame che riceve vengono smistati solo sull'interfaccia dove è attestato il destinatario.

Sono sostanzialmente dei Bridge ad alte prestazioni con interfacce multiple.

Vantaggi:

- ▶ **Aumento del Throughput** sotto carico. Uno Switch a N porte può essere attraversato contemporaneamente da $N/2$ connessioni.
 - ▶ Ogni connessione può essere **Full-Duplex** (raddoppio del Throughput)
 - ▶ **Security:** Sull'interfaccia di un terminale non transitano Frame di altre connessioni.
 - ▶ I segmenti diventano domini di collisione separati o **privi di collisioni** (Switch-PC).



Tecnologie Switch

Gli Switch hanno una tabella interna con le associazioni MAC-Interfaccia.

Costruisce la tabella leggendo gli indirizzi dai Frame che lo attraversano.

Se il mittente di un Frame non è in tabella viene inserito.

Se il destinatario non è in tabella o è broadcast

(ff-ff-ff-ff-ff-ff) il Frame viene inviato a tutti (flooding).

La commutazione è realizzata da dispositivi Hardware

(Crossbar Switch)

Esistono 2 tipi di Switch:

- ▶ **Store-and-forward**

Il Frame viene ricevuto interamente in un buffer, viene verificato il CRC, poi ritrasmesso se la linea di uscita è libera (vedi immagine).

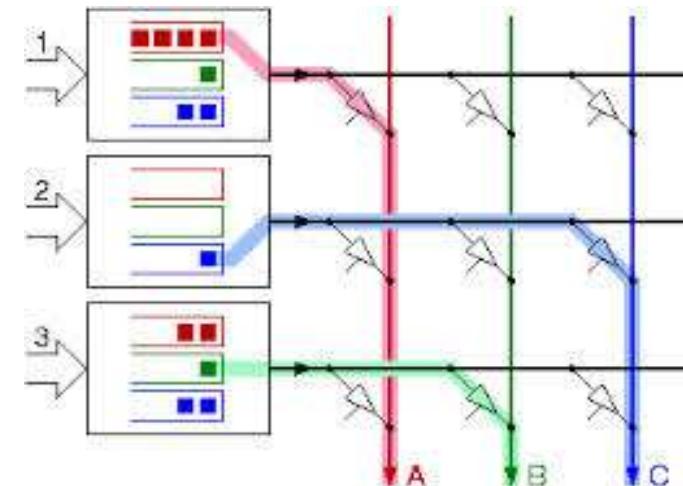
Assenza di collisioni. Latenze dovute a processamento del frame e dall'accodamento

Consente l'utilizzo di velocità eterogenee (autosensing 10baseT/100baseTX/1000baseT)

- ▶ **Cut-Through**

L'indirizzo di destinazione viene analizzato mentre il Frame sta entrando nello Switch.

Il Frame viene instradato sull'interfaccia di uscita senza bufferizzazione, dopo che sono stati letti i primi 14 byte (circa 25 us per Ethernet e 7 us per Fast Ethernet). Tempi di latenza bassi, ma vengono inoltrati anche Frame corrotti e si estende il dominio di collisione.

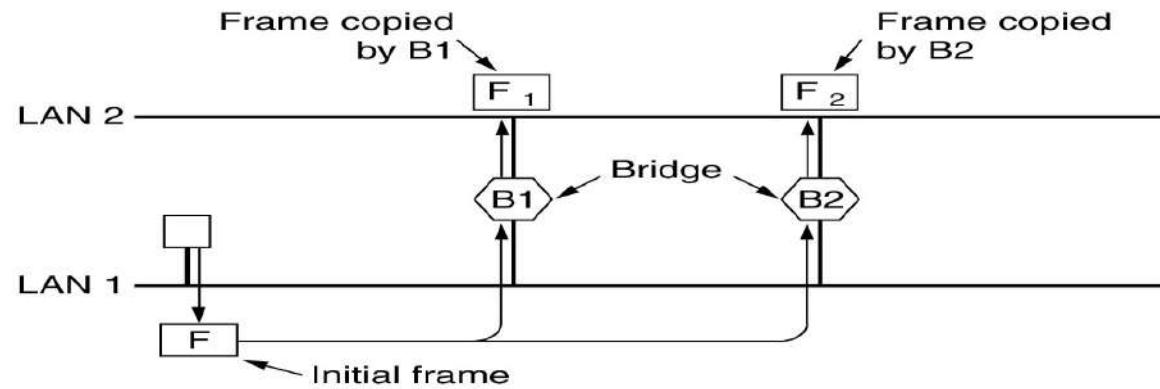


Spanning Tree

In una LAN composta da Bridge/Switch possiamo avere una topologia magliata per 2 possibili motivi:

- ▶ Progetto di ridondanza, per garantire percorsi alternativi in caso di guasti
- ▶ Errore di configurazione.

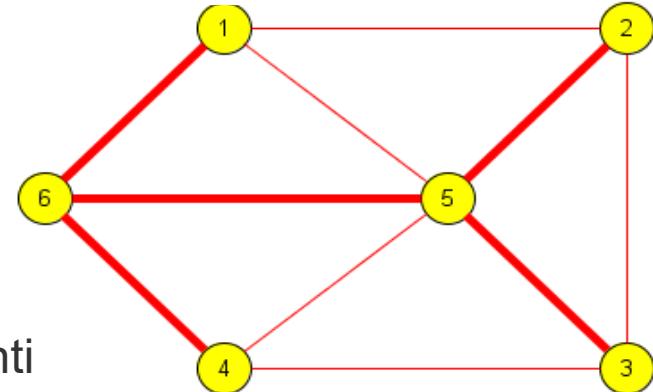
La presenza di anello nella rete potrebbe però portare a situazioni in cui i frame circolano all'infinito:



Si tratta quindi di rendere aciclico un grafo connesso con archi pesati non orientati, isolando in modo opportuno alcuni archi.

Il metodo utilizzato nelle LAN consiste nel determinare un albero ricoprente (**Spanning Tree**) sopra il grafo.

L'algoritmo che determina l'albero e isola gli archi eccedenti nelle LAN è detto **Spanning Tree Protocol (STP)**.



Bridge con Spanning Tree Protocol

Funzionalità:

- ▶ Deve intervenire nel più breve tempo possibile
- ▶ Deve introdurre un overhead limitato
- ▶ Deve essere flessibile, cioè deve poter ammettere che un bridge venga aggiunto successivamente a rete configurata senza che sia indispensabile riconfigurare tutta la rete.

L'algoritmo distribuito STP è stato proposto da Perlman nel 2000 ed è stato standardizzato con il nome **IEEE802.1D**

Il protocollo si basa sullo scambio di frame (detti BPDU- Bridge Protocol Data Unit) tra i bridge del dominio di broadcast utilizzando l'indirizzo multicast **01:80:C2:00:00:00**.

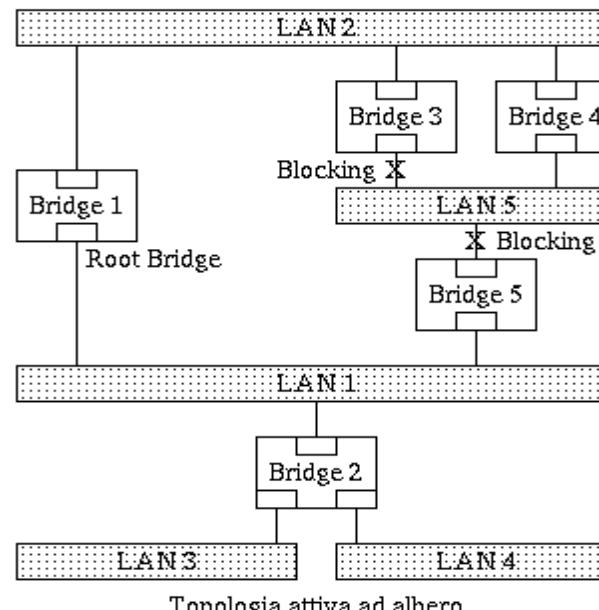
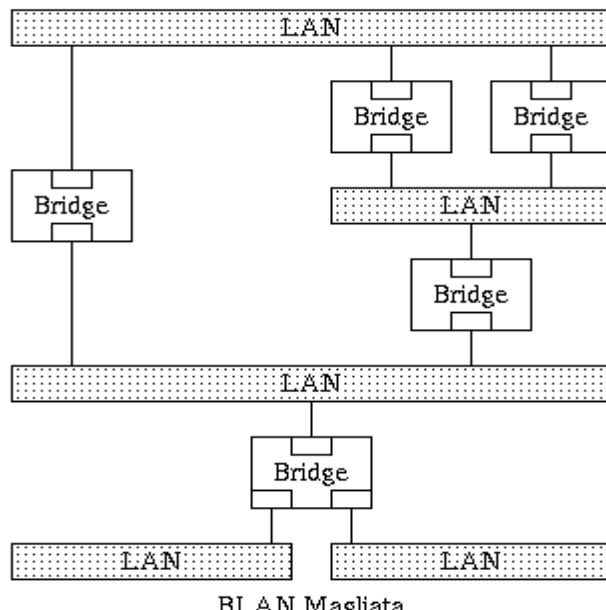
Ci sono 3 tipi di BPDU:

- **Configuration** BPDU, usati per il calcolo la configurazione del protocollo
- **Topology Change Notification** e **Topology Change Notification Ack** usati per annunciare un cambiamento della topologia

Spanning Tree Protocol: l'algoritmo

L'algoritmo opera nei seguenti passi:

- 1) elezione del **Root-Bridge**: ogni bridge ha un proprio identificativo (ID) che spedisce in multicast in modo che tutti i bridge sappiano chi è il bridge con l'ID più piccolo. Questo bridge diventa la radice dell'albero.
- 2) selezione della **Root-Port**: per ogni bridge si seleziona la porta più conveniente per interconnetterlo al root
- 3) selezione del **Designated-Bridge**: per ogni LAN si sceglie quale bridge è designato ad interconnetterla con il root; la porta utilizzata è la **Designated-Port**. Al termine di queste azioni, lo spanning procede alla messa in **Stato di Blocking** per tutte le porte che non sono Root-Port o Designated-Port.



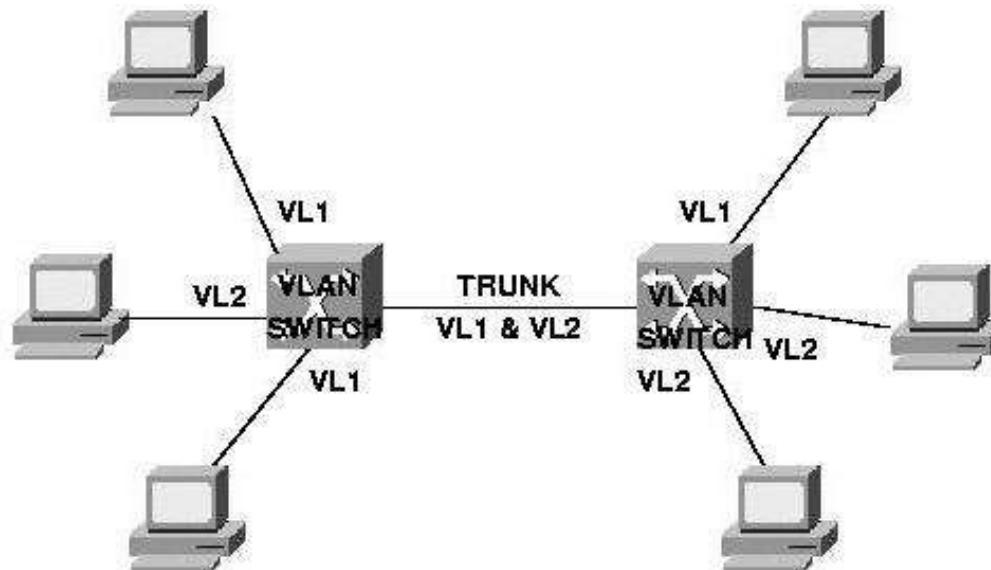
LAN VIRTUALI (VLAN)

Sfruttano la capacità di inoltro intelligente da parte degli Switch.

Permettono di costruire su un'unica infrastruttura fisica più LAN logicamente separate.

Vantaggi:

- ▶ **Limitano il traffico di Broadcast** all'interno della singola VLAN.
- ▶ Permettono la **progettazione logica** della rete indipendentemente dalla dislocazione fisica delle Stazioni.
- ▶ Aumentano il livello di **sicurezza** della rete confinando il traffico interno di ogni VLAN alle sole stazioni appartenenti ad essa.



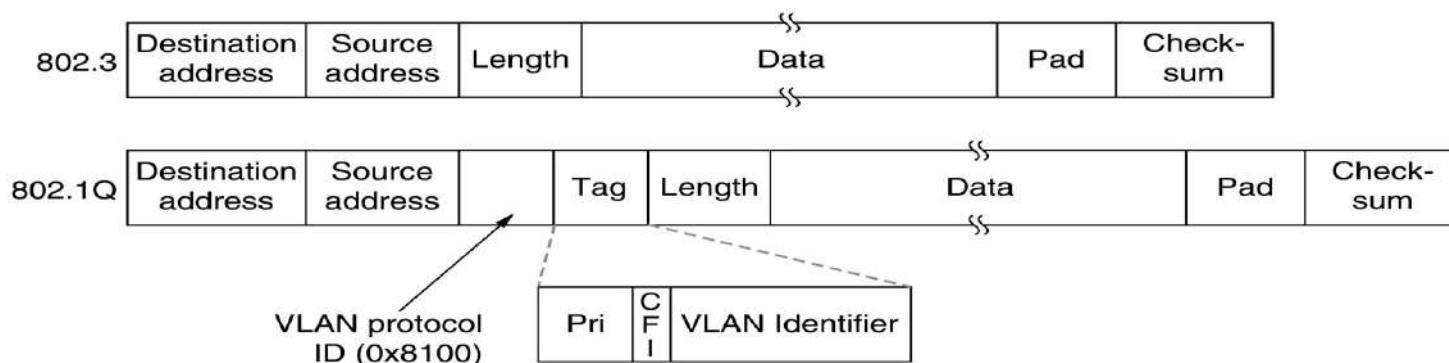
Criteri di Appartenenza alle VLAN

I VLAN-Switch sono Switch che supportano la gestione VLAN mediante opportune tabelle di configurazione in cui sono elencate le VLAN disponibili e le porte associate. Ad una porta possono essere associate più VLAN (se dietro la porta c'è un HUB o uno Switch)

Le trame che attraversano diversi Switch devono essere etichettate per consentirne una corretta gestione. Occorre quindi una modifica dell'header del Frame per far posto all'etichetta VLAN. Il primo Switch che tocca il Frame aggiunge l'etichetta e l'ultimo sul percorso la rimuove.

Una porta di uno switch su cui viaggiano frame con VLAN TAG è detta **tagged port**. Una porta su cui viaggiano frame senza modifica (verso un host) è detta **untagged**.

802.1Q è lo standard proposto nel 1998 da IEEE per la modifica del Frame Ethernet.



Criteri di Appartenenza alle VLAN

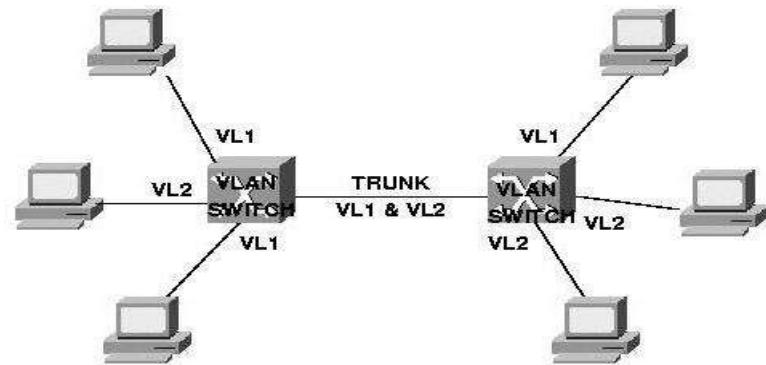
L'appartenenza ad un host ad una VLAN può essere definita secondo vari criteri:

- ▶ **Porte:** ciascuna porta di uno Switch è configurata per appartenere ad una data VLAN.

Esempio di configurazione di 2 VLAN in uno Switch:

VLAN1 : port 1-4 untagged, port 25 tagged

VLAN2: port 5-24 untagged, port 25 tagged



- ▶ **Autenticazione:** diversi apparati possono essere assegnati automaticamente ad una VLAN sulla base di criteri di autenticazione (esempio Access Point WiFi).
- ▶ **Protocollo:** Ad esempio i pacchetti IP possono appartenere ad una VLAN , diversa da quella usata dai pacchetti IPX.



UNIVERSITÀ
DI PARMA

DIPARTIMENTO DI SCIENZE MATEMATICHE, FISICHE ED INFORMATICHE
Corso di Laurea in Informatica

Il Livello Data-Link

Parte III : Reti Wireless

RETI DI CALCOLATORI - a.a. 2023/2024

Roberto Alfieri

Il Livello Data-Link: sommario

PARTE I

- ▶ Scopi del livello Data-Link
- ▶ Framing, Rilevazione e correzione degli errori, controllo di flusso
- ▶ Protocolli per reti Punto-punto: PPP.
- ▶ Protocolli per reti MultiAccesso: Aloha, CSMA, CSMA/CD, CSMA/CA

PARTE II

- ▶ Gli standard IEEE802
- ▶ Ethernet: Sottoliv. MAC e LLC, tecnologie Ethernet, il Frame, Repeater, Switch, Bridge
- ▶ Spanning Tree Protocol.
- ▶ Lan Virtuali

PARTE III

- ▶ Lan Wireless

RIFERIMENTI

- ▶ *Reti di Calcolatori, A. Tanenbaum, ed. Pearson*
- ▶ *Reti di calcolatori e Internet, Forouzan , Ed. McGraw-Hill*

Reti Wireless

Famiglie di protocolli Wireless IEEE802:

- ▶ 802.11: Wireless LAN (Local Area Network) casa, scuola, ufficio, ... (e.g. WiFi)
- ▶ 802.16: Wireless MAN (Metropolitan Area Network) entro i limiti urbani. (e.g. WiMax)
- ▶ 802.15: Wireless PAN (Personal Arean Network) entro i 10 metri (e.g. Bluetooth)

Tecnologie Wireless per l'IoT:

- ▶ Low Power Short Range : RFID
- ▶ Low Power LAN

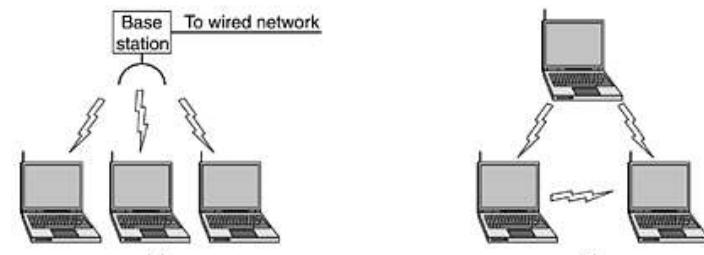
IEEE 802.11: architettura e stack dei protocolli

IEEE802.11 è il principale standard per le LAN wireless.

Wi-Fi Alliance è una organizzazione, composta da industrie leader nel settore, che collabora con IEEE802.11 per guidare l'adozione standardizzata delle diverse tecnologie WiFi.

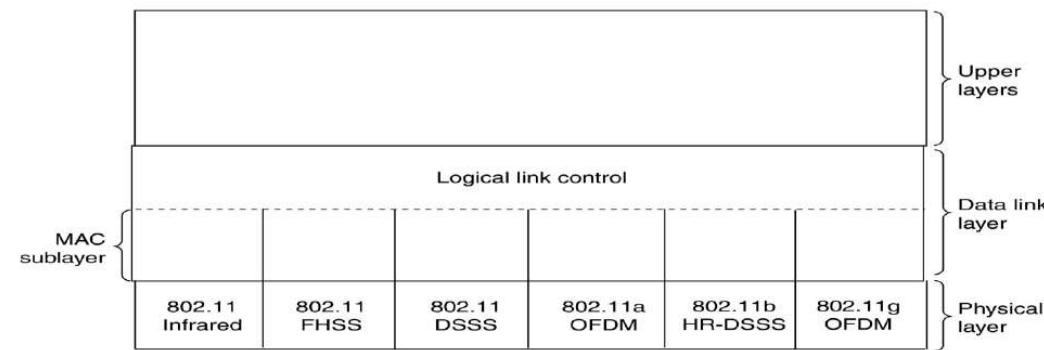
Due modalità di utilizzo:

- con **infrastruttura**: ogni client è associato ad una Stazione Base (Bridge)
- **reti ad hoc**: rete di computer associati tra loro (poco utilizzata)



Lo standard 802.11 si compone di 2 sottostrati:

- **sottostrato MAC 802.11**: si occupa delle problematiche di condivisione del mezzo trasmittivo e fa parte del Data-link layer.
- **sottostrato Fisico 802.11**: si occupa della codifica dei dati. Esistono diverse varianti (es. 802.11a, 802.11b, ecc)



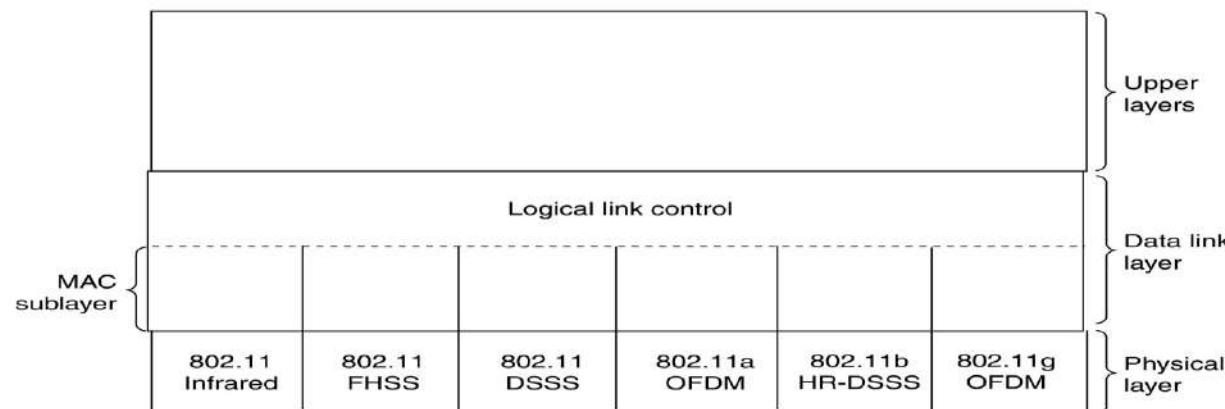
802.11 : Le varianti dello strato fisico

Il sotto-strato fisico può utilizzare 6 diverse tecniche di trasmissione (Infrared, HDSS, DSSS, OFDM, HR-DSSS e OFDM)

Dal 1997 ad oggi sono stati realizzati diversi protocolli che utilizzano queste tecniche, tra cui: 802.11, 802.11a 802.11b, 802.11g, 802.11n, 802.11ac

Tutte le implementazioni (eccetto gli infrarossi in 802.11) operano nelle frequenze ISM: 902-928MHz, 2400-2483MHz e 5727-5850MHz

Il canale è multi-accesso per cui il throughput è condiviso.



Il sottostrato fisico 802.11

802.11 Pubblicato nel 1997 è il protocollo originale.

Aveva una velocità limitata a 1 o 2 Mbps con 3 varianti tecniche:

- ▶ **Infrarosso** ($0.85\mu\text{m}$ o $0.95\mu\text{m}$) Non penetra i muri ed è sensibile alla luce solare.
- ▶ **Frequency Hopping Spread Spectrum (FHSS)**
- ▶ **Direct Sequence Spread Spectrum (DSSS)**

Il sottostato fisico 802.11b

802.11b Pubblicato nel 1999.

Opera a 2.4GHz con la tecnica di modulazione **HR-DSSS** (HighRate DSSS) con uno schema di modulazione denominato Complementary Code Keying (CCK).

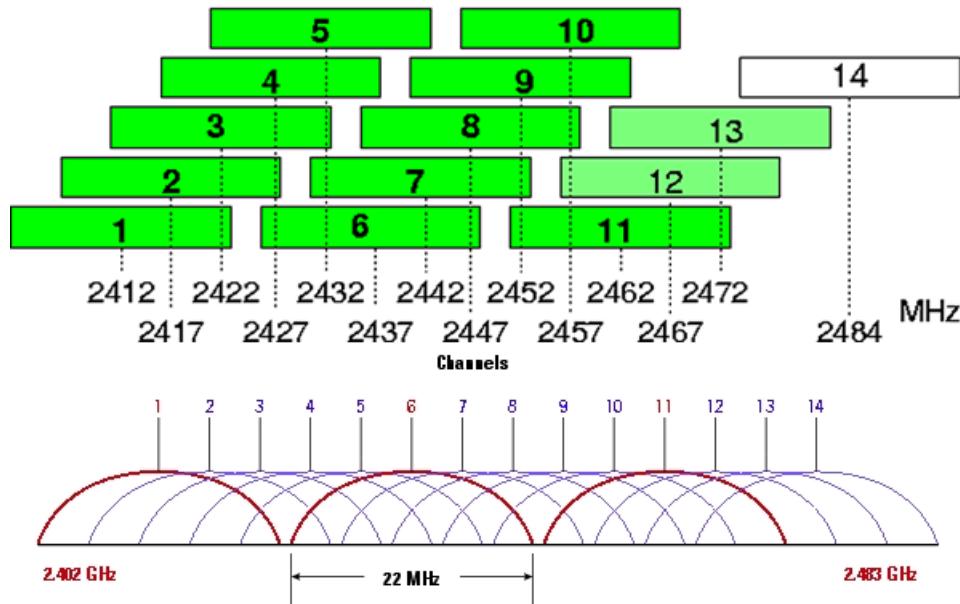
Ha una velocità dichiarata di 11Mb/s, ma il protocollo MAC CSMA/CA introduce un overhead che ne **dimezza** la velocità reale (5.9Mbps TCP).

Se il segnale è debole la velocità massima può essere ridotta a 5.5, 2 o 1 Mbps

Le velocità di 1 e 2 Mbps funzionano in DSSS per compatibilità con 802.11

Lo spettro (83 MHz da 2400 a 2483) è suddiviso in 14 canali da 22 MHz parzialmente sovrapposti.

Per evitare interferenze vengono generalmente utilizzati 2 gruppi di canali (1,6,11 e 2,7,12) quando ci sono diverse reti WiFi.

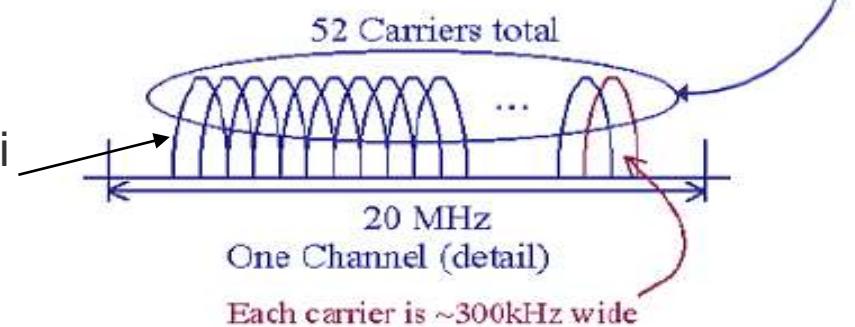


Il sottostato fisico 802.11a

Pubblicato nel 2001 utilizza 8 canali da 20MHz attorno a 5.2GHz raggiungendo una velocità di 54Mb/s (25/31 reali) ridotta a 48, 36, 34, 18, 9 o 6 Mb/s.



La tecnica utilizzata per la codifica è la **OFDM** (Orthogonal Frequency-Division Multiplexing): Ogni canale da 20MHz è suddiviso in 52 portanti da 300KHz (48 dati + 4 per la sincronizzazione) con modulazione di fase (QPSK) e di ampiezza e fase (QAM), come ADSL.

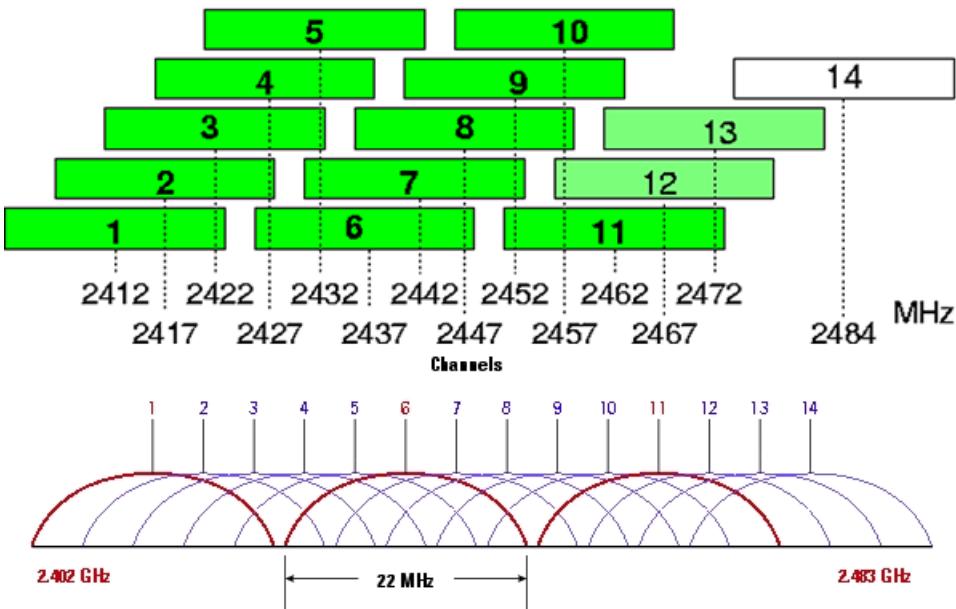


La trasmissione ortogonale consente di ridurre l'interferenza tra canali adiacenti parzialmente sovrapposti.

Questo standard non ha riscosso i favori del pubblico dato che l'802.11b si era già molto diffuso e in molti paesi (Europa) l'uso delle frequenze a 5 GHz era riservato.

Il sottostato Fisico 802.11g

Pubblicato nel 2003 utilizza la frequenza di 802.11b, con cui è backward-compatible, e la modulazione OFDM su 52 portanti come 802.11a, ma utilizza i 14 canali da 22MHz attorno a 2.4GHz di 802.11b.



La velocità è di 54 Mbps (24 Mbps reali), ma può scendere a 48, 36, 18, 12, 9 e 6. Può anche lavorare a 11 o 5.5 Mbps in modalità retro-compatibile con 802.11b

I sottostati 802.11n e 802.11ac

IEEE802.11n (Denominato Wi-Fi 4 da WiFi Alliance)

La versione definitiva dello standard è stata ratificata nel 2009.

Opera a 2.4 e 5.4 Ghz con canali da 20MHz o 40MHz

Utilizza OFDM con modulazione QPSK, 16-QAM o 64-QAM

Altri miglioramenti rispetto a 802.11g grazie alla tecnologia MIMO (Multiple Input Multiple Output): si utilizzano fino a 4 antenne per trasmettere fino a 4 flussi radio paralleli oppure per migliorare il raggio di copertura.

Utilizzando canali da 40MHz e 2 antenne MIMO si raggiunge una velocità attorno ai 300Mb/s (200 reali).

IEEE802.11ac (Denominato Wi-Fi 5 da WiFi Alliance)

La versione definitiva dello standard è stata ratificata nel 2014.

Rispetto a 802.11n opera solo a 5 GHz con canali più larghi (80 o 160 MHz), più antenne Mimo (fino a 8) e una modulazione fino a 256-QAM.

La velocità massima per flusso (banda 160Mhz , 256 QAM, 1 antenna Mimo) è di circa 800 Mbps (throughput 600 Mbps).

http://www.cisco.com/c/en/us/products/collateral/wireless/aironet-3600-series/white_paper_c11-713103.html

I sottostrati 802.11ax e 802.11ay

IEEE802.11ax https://en.wikipedia.org/wiki/IEEE_802.11ax

Denominata Wi-Fi 6 da WiFi Alliance

La versione definitiva dello standard è stata ratificata nel 2019.

Velocità 4x rispetto a 802.11ac

Opera a 2.4 e 5.4 Ghz con canali raddoppiati (20 o 40 Mhz)

Utilizza OFDM con modulazione fino a 1024-QAM

Utilizzo previsto: ambienti ad alta densità di dispositivi collegati e di persone presenti

IEEE802.11ay https://en.wikipedia.org/wiki/IEEE_802.11ay

Attualmente in fase di definizione

Opera a 60GHz (onde millimetriche), con velocità 20-40 Gbps a 300-500 metri

L'utilizzo previsto: in sostituzione di ethernet delle case e negli uffici.

E' la risposta WiFi all'utilizzo delle onde millimetriche in 5G.

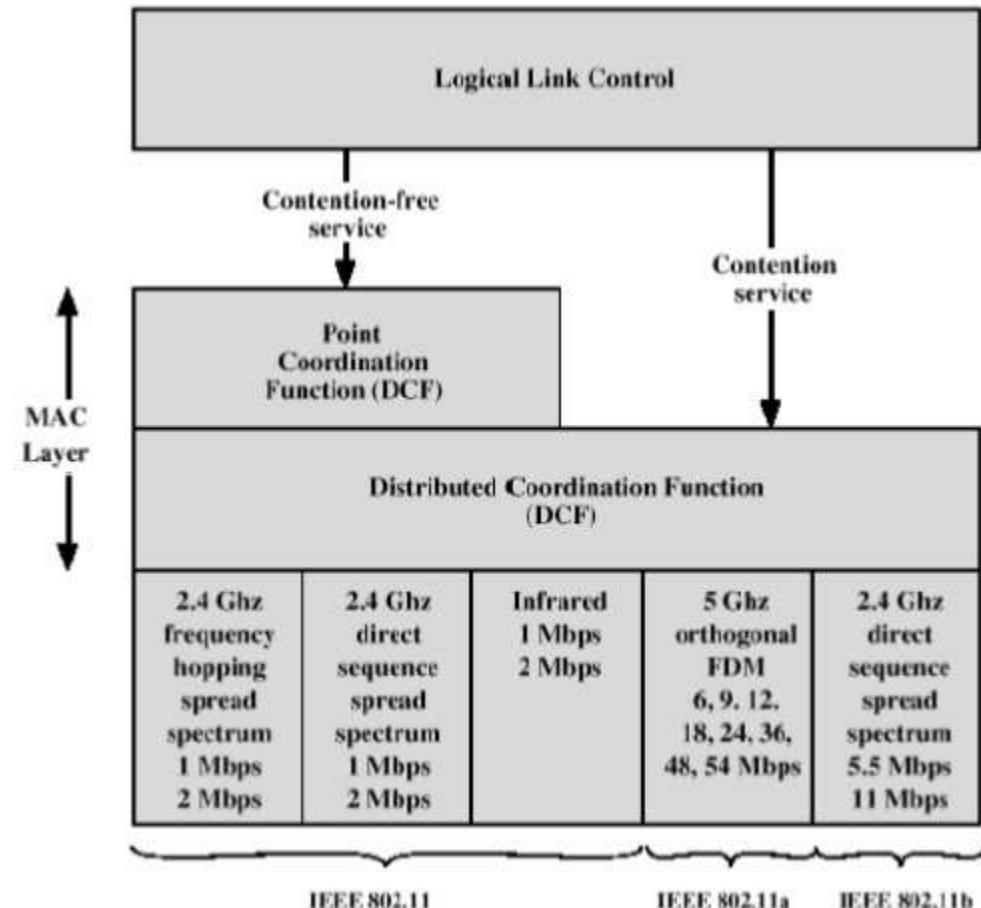
Il sottostrato MAC di 802.11

Copre 3 funzioni:

- ▶ Consegna affidabile dei Frame
- ▶ Controllo dell'accesso
- ▶ Sicurezza

Fornisce 2 modalità di funzionamento:

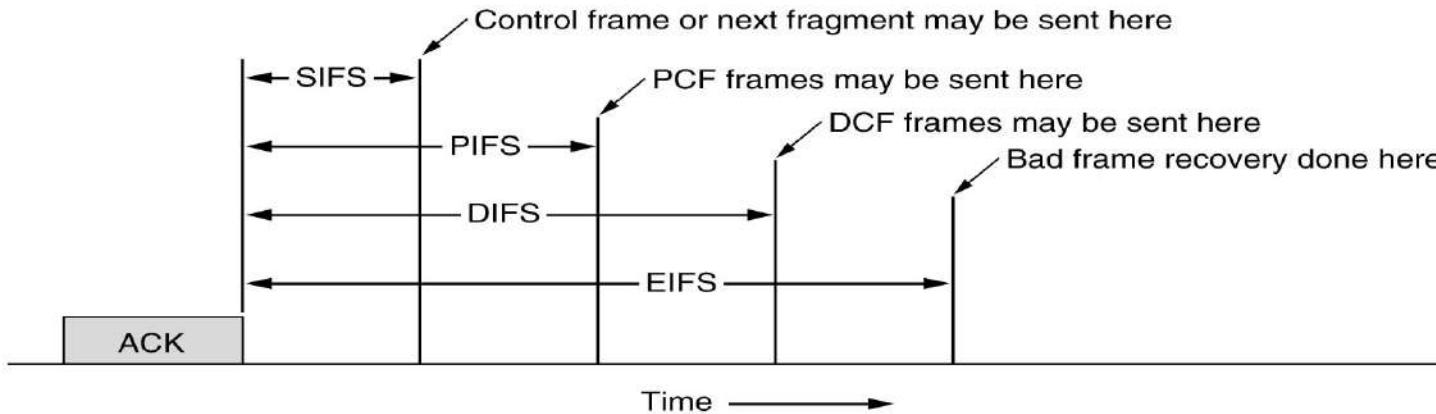
- ▶ A contesa: DCF
(Distributed Coordination Function)
- ▶ Senza contesa: PCF
(Point Coordination Function)



Protocolli MAC: Inter-Frame Space (IFS)

DCF e PCF possono convivere nella stessa cella, grazie ad una opportuna assegnazione dei tempi di attesa (priorità nella risposta):

- SIFS (Short Inter Frame Space). Per separare i Frame di una singola trasmissione.
- PIFS (PCF IFS). Un eventuale AP può inviare i suoi Frame di controllo.
- DIFS (DCF IFS). Se l'AP tace le stazioni possono tentare di acquisire il canale.



Protocollo MAC a contesa del canale (DCF)

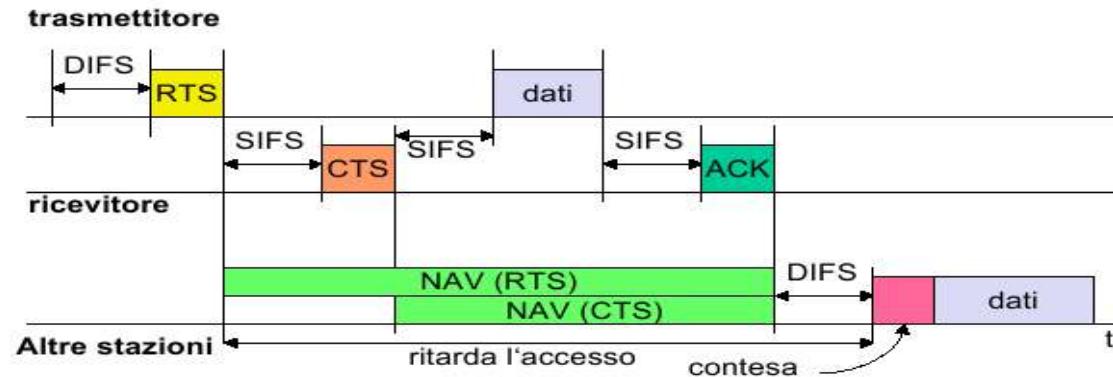
DCF (Distributed Coordination Function)

Gestione distribuita (come Ethernet), con **CSMA/CA**

Quando la stazione deve trasmettere manda un RTS, quindi attende il CTS dal destinatario ed invia i Frame. Per ogni frame inviato attiva un Timer e attende un ACK. Se non arriva l'ACK (CRC errato, ..) il protocollo si ripete.

Al contrario delle reti wired, le reti wireless sono soggette a rumore e quindi inaffidabili. Se i frame sono troppo lunghi hanno poche probabilità di arrivare intatti a destinazione. Per questo 802.11 ammette frammentazione in parti piccole.

Una volta acquisito il canale (con RTS e CTS) più frammenti possono essere inviati in sequenza (Fragment Burst); il NAV copre solo il primo frammento, mentre i frammenti successivi sono garantiti dal tempo di SIFS.



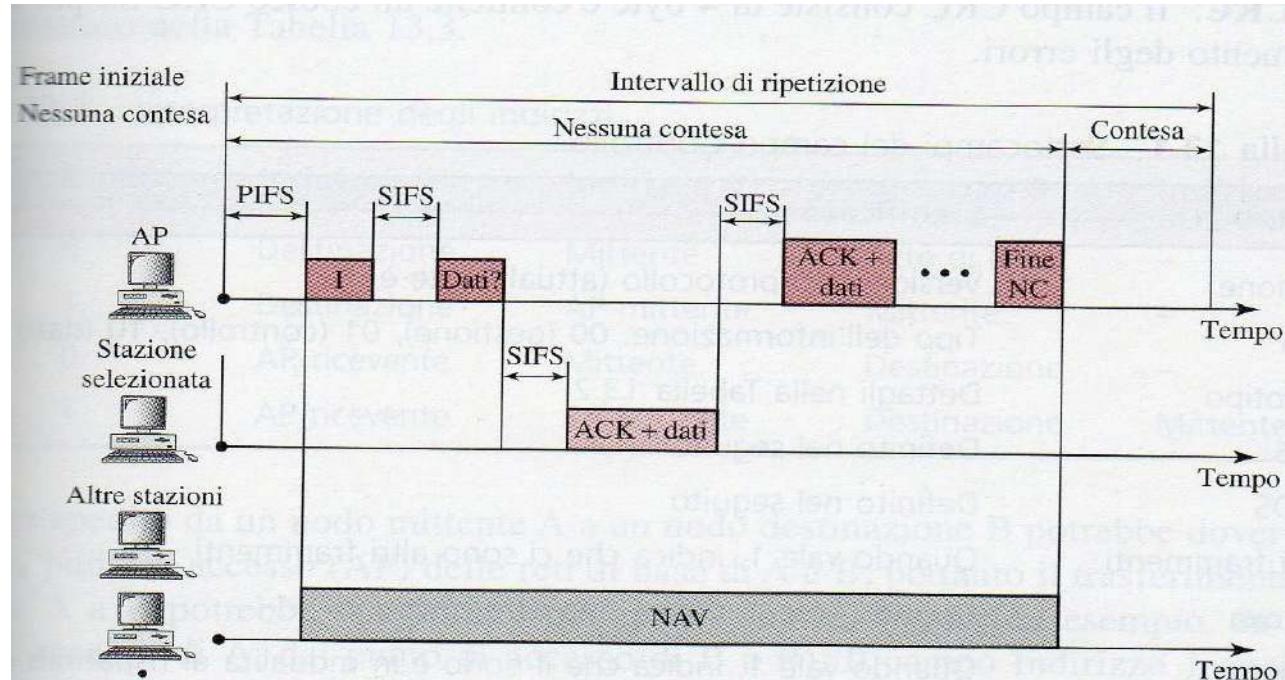
Protocollo MAC senza contesa (PCF)

L'AP gestisce tutte le stazioni della sua cella.

Un broadcast (Beacon Frame) viene inviato da 10 a 100 volte al secondo per sincronizzare le stazioni e rappresenta l'invito ad associarsi da parte di nuove stazioni, che rispondono con le caratteristiche della stazione.

Le stazioni associate vengono abilitate a comunicare in un periodo senza contesa con un meccanismo di polling . Al termine l'AP invia un frame di fine del periodo di Nessuna Contesa e possono riprendere le contese (vedi figura).

Tutte le implementazioni supportano DCF, mentre PCF è opzionale.

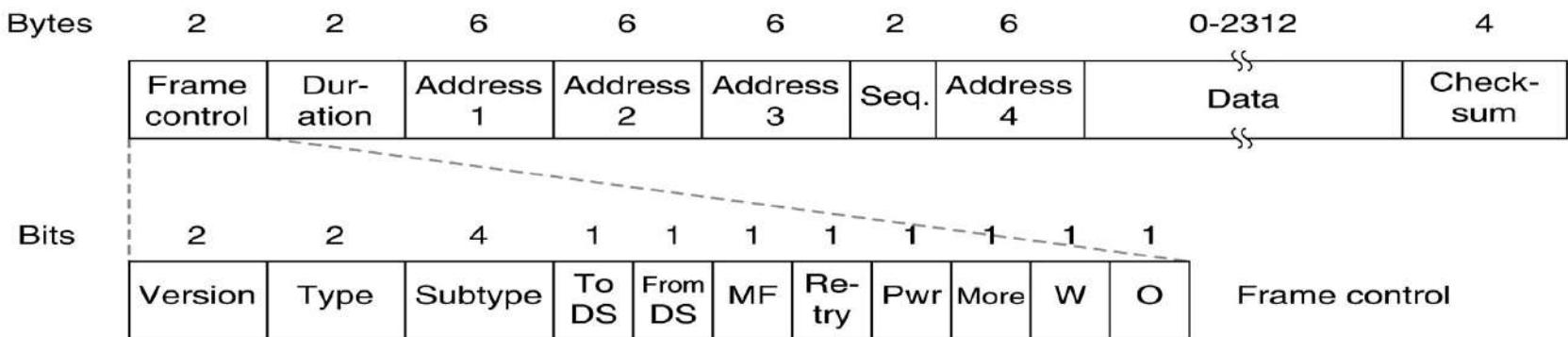


Il Frame di 802.11

Esistono 3 tipi di Frame: Dati, Controllo (RTS CTS ACK) , Management

Campi di un Frame:

- ▶ Frame Control: vedi Slide successiva
- ▶ Duration: consente di prevedere per quanto tempo il mezzo resterà occupato.
- ▶ Address: Formato IEEE 802 (48bit). Il Frame dati contiene 4 indirizzi perché oltre a MAC delle stazioni di origine e destinazione sono presenti anche quelli degli AP di entrata e di uscita.
- ▶ Sequence: consente di numerare i frammenti
- ▶ Data: è il Payload lungo fino a 2312 byte (header 34 byte, max totale=2346 byte)
- ▶ Non c'è minimo poiché non è possibile avere collisione sui dati.
- ▶ Checksum: è il solito CRC-32



Vedi: <https://witestlab.poly.edu/blog/802-11-wireless-lan-2/>

Frame 802.11: Campo Controllo

Il primo ottetto è suddiviso in 3 campi con il seguente significato:

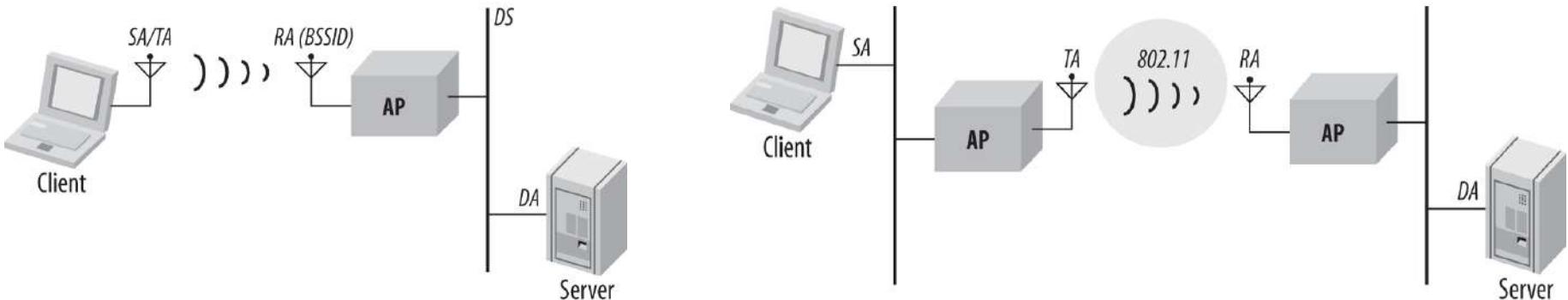
- ▶ **Versione** - Versione dello standard IEEE 802.11
- ▶ **Tipo (2 bit)** Specifica il tipo del frame: Management (00) , Controllo (01) o Dati (10)
- ▶ **Sottotipo (4 bit)** Alcuni Esempi:
 - Management: Assoc. Req (0000), Assoc. Resp (0001), Beacon (1000),
 - RTS (01-1011) , CTS (01-1100) , ACK (01-1101), Beacon (00-1000), R

Gli 8 flag che seguono, quando impostati ad 1, hanno il seguente significato:

- ▶ **Al DS** - il frame è diretto al Sistema di Distribuzione
- ▶ **Dal DS** - il frame proviene dal Sistema di Distribuzione
- ▶ **Altri Frammenti (More Fragment)** - seguono altri frammenti appartenenti allo stesso frame
- ▶ **Ripetizione (Retry)**- questo frammento è la ripetizione di un frammento precedente
- ▶ **Risparmio energia** – la stazione base mette una stazione in sleep (modalità basso consumo)
- ▶ **Altri Frame** - il trasmettitore ha altri frame per il ricevitore
- ▶ **WEP** - il campo *Dati* è stato crittografato con l'algoritmo WEP (Wired Privacy)
- ▶ **Ordinati** - frammento appartenente alla classe di servizio *StrictlyOrdered*

Frame 802.11: Indirizzi

Nell' 802.11 il trasmettitore (TA) e ricevitore (RA) potrebbero non coincidere con la sorgente (SA) o destinazione (DA) del frame. In alcuni casi la comunicazione a livello link passa per l'AP (BSSID). Vedi <https://networkengineering.stackexchange.com/questions/25100/four-layer-2-addresses-in-802-11-frame-head>



To DS From DS	Meaning
To DS = 0, From DS = 0	A data frame direct from one STA to another STA within the same IBSS, as well as all management and control type frames.
To DS = 0, From DS = 1	Data frame exiting the DS.
To DS = 1, From DS = 0	Data frame destined for the DS.
To DS = 1, From DS = 1	Wireless distribution system (WDS) frame being distributed from one AP to another AP.

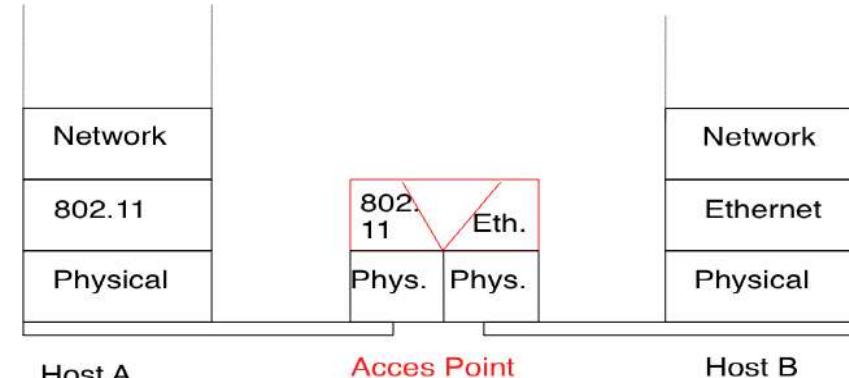
To DS	From DS	Address 1	Address 2	Address 3	Address 4
0	0	RA = DA	TA = SA	BSSID	N/A
0	1	RA = DA	TA = BSSID	SA	N/A
1	0	RA = BSSID	TA = SA	DA	N/A
1	1	RA	TA	DA	SA

MTU e frammentazione

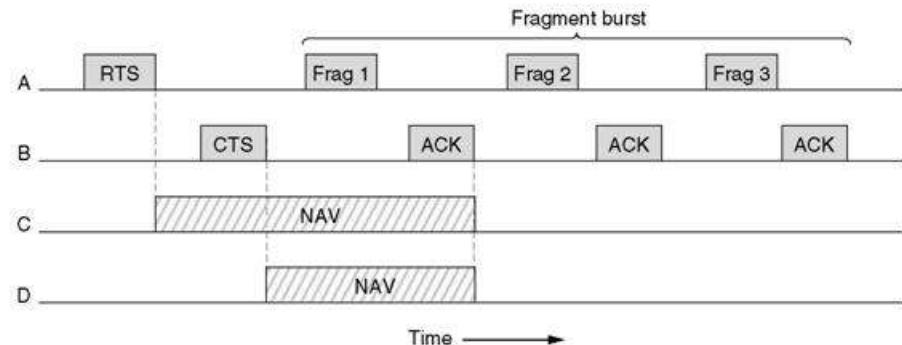
Il payload può arrivare fino a 2312 byte, che è quindi l'MTU di IEEE-802.11, ma è possibile definire una soglia di frammentazione tra 400 e 2312 byte

Questo valore è controllabile dall'utente

(solitamente è 1500 quando Wifi è associato a Ethernet, come negli access-point)



La frammentazione migliora le prestazioni poiché riduce la probabilità di errore (e quindi reinvio)



Servizi

Ogni LAN wireless conforme allo standard 802.11 deve fornire servizi classificabili in 2 categorie:

“**servizi di distribuzione**” : sono forniti dall’AP

“**servizi di stazione**”: devono essere assolti da tutte le stazioni

I servizi sono gestiti mediante lo scambio di appositi Frame di Management.
Un Frame di Management importante è il **Beacon** che viene inviato dall'Access Point ad intervalli regolari (tipicamente 10 al secondo) e contiene l'identificatore della cella Service Set Identifier (SSID). In questo modo le stazioni vengono a conoscenza degli Access Point attivi e possono inviare Frame di Management per la richiesta di servizi.

Servizi di Distribuzione

1. **Associazione** - Appena una stazione entra nel raggio d'azione di un AP, invoca questo servizio per informare la stazione base della sua presenza e delle sue necessità.
2. **Dissociazione (Separazione)** - Sia le stazioni che gli AP possono terminare una precedente associazione. Una stazione mobile dovrebbe utilizzare questo servizio prima di spegnersi o lasciare la cella.
3. **Riassociazione** - Una stazione in moto può trasferire il controllo da un AP all'altro.
4. **Distribuzione** - L'AP smista i frame che lo raggiungono verso le stazioni della propria cella (via radio) o verso gli altri AP, attraverso il sistema di distribuzione (rete cablata).
5. **Integrazione** - Questo servizio gestisce la traduzione dei frame 802.11 verso altri formati (es. 802.3)

Servizi di Stazione

Dopo aver completato l'Associazione vengono utilizzati i servizi di Stazione:

1. Autenticazione - Una stazione deve autenticarsi per evitare che i frame arrivino a stazioni non autorizzate. Dopo l'Associazione la base invia un frame Challange a cui la stazione deve rispondere con un Frame per l'autenticazione.

Ci sono due tipi di autenticazione:

- **A sistema Aperto**: nessuna sicurezza.

- **A chiave condivisa**: l'AP condivide con tutte le stazioni una chiave segreta. L'AP invia una stringa di prova (Challenge); la stazione codifica il Challenge con la chiave e la invia all'AP.

2. Deautenticazione - Una stazione che voglia abbandonare la rete deve “deautenticarsi” e “dissociarsi”.

3. Riservatezza - I dati trasmessi via radio possono essere ascoltati da chiunque si trovi all'interno dell'area di diffusione. Questo servizio gestisce la crittografia WEP (Wired Equivalent Privacy) dei frame attraverso l'algoritmo RC4 con chiavi a 64 o 128 bit (incluso il vettore di inizializzazione)

4. Trasmissione - Scambio di frame, fra due stazioni, a livello di MAC sublayer.

MAN Wireless

Consente di distribuire dati in area Metropolitana, su un agglomerato di case tramite una potente antenna, con una velocità di trasmissione fino a 70Mbps e un raggio fino a 50Km. Utilizza frequenze non ISM, maggiori rispetto a WiFi (comprese tra 2 e 66 Ghz).

La tecnologia e la sua evoluzione sono controllati da **WiMax Forum**, un consorzio di 420 aziende nato nel 2001.

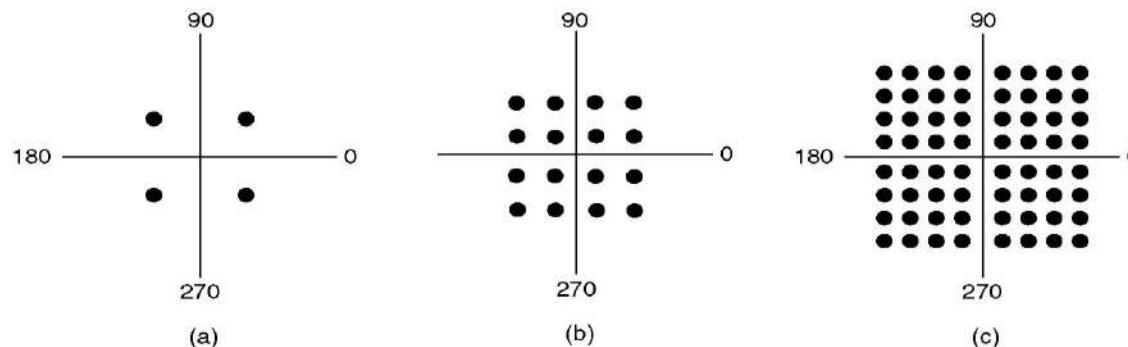
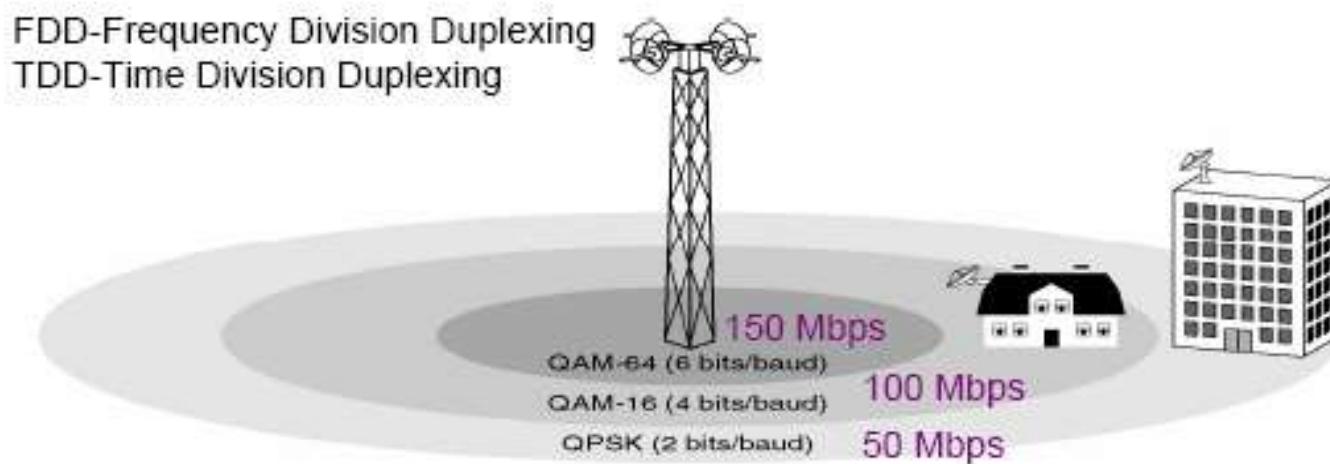
Il gruppo di lavoro IEEE che se ne occupa è **802.16** (WiMax).

Standard 802.16 attivi:

- ▶ IEEE 802.16d: detto anche 802.16-2004, per utenze fisse.
Alternativo al cablaggio dell' "ultimo miglio". Banda 2-11 Ghz, velocità 75 Mbps
- ▶ IEEE 802.16e : per utenti mobili (fino a 120 Km/h) Banda 2-6 GHz, velocità 15 Mbps
- ▶ IEEE 802.16f : Supporto multihop. Le stazioni possono fare da ponte per stazioni che non raggiungono la stazione base.

Lo strato Fisico di 802.16

Le onde viaggiano in linea retta e l'intensità diminuisce bruscamente con la distanza.
Per questo adotta 3 diversi schemi di modulazione: QPSK, QAM-16 e QAM-64 e
una tecnica tipica di trasmissione OFDM con 256 sotto-portanti



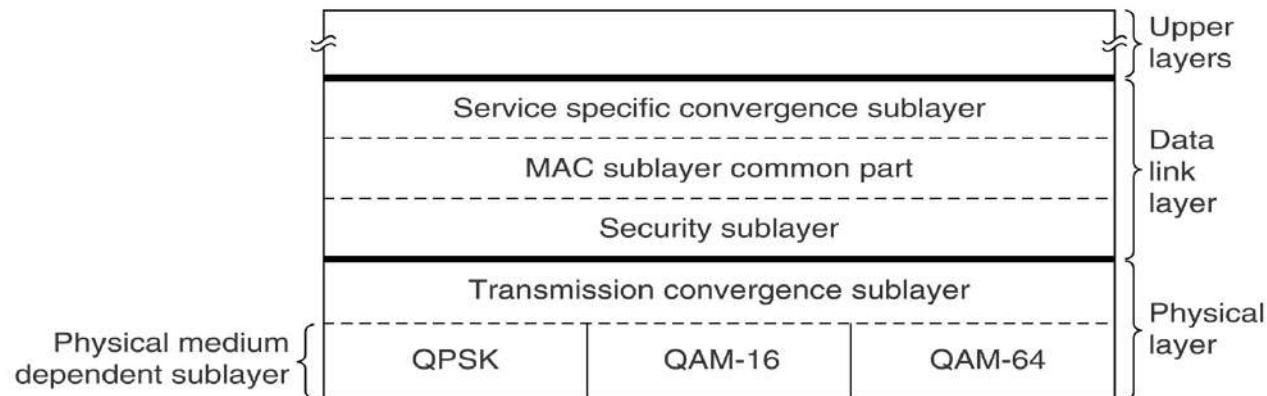
Il protocollo MAC di 802.16

Il livello data-link è suddiviso in 3 sotto-strati.

Lo stato più basso (Security) si occupa della cifratura del payload (non dell'intestazione).

Il MAC Common Part garantisce l'accesso al sistema, l'allocazione della banda, l'instaurazione e la manutenzione (establishment and maintenance) della connessione. Usa un algoritmo di scheduling. Una stazione deve competere una sola volta per entrare nella rete, poi gli viene allocato uno slot che può crescere o diminuire ma rimane sempre allocato alla stazione

Il MAC Service Specific comunica con il livelli superiori fornendo diversi livelli di servizio: A bit-rate costante, bit-rate variabile in tempo reale, bit-rate variabile in tempo non reale, e Best Effort.



Formato dei Frame 802.16

EC indica che il Payload è cifrato

Type è il tipo di Frame

CI indica se esiste un CRC finale

EK indica le chiavi di codifica utilizzate

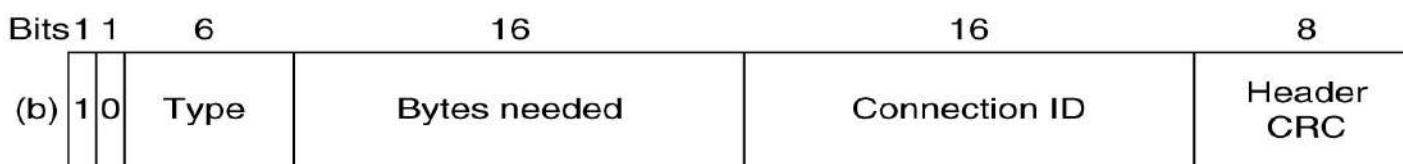
Length è la lunghezza complessiva del Frame

ConnID indica la connessione di appartenenza del frame

Header CRC è il CRC solo per l'header

Payload (può essere cifrato con chiave simmetrica CBC o 3DES scambiate con chiave pubblica RSA e X509)

Il **CRC finale** è opzionale (i frame errati non verrebbero comunque ritrasmessi)



(a) Frame generico

(b) frame di richiesta di banda

WiMAX in Italia e in Europa

Bando per l'asta delle licenze [WiMAX](#) a Ottobre 2007 con concessioni decennali a livello provinciale o regionale.

E' la versione fissa 802.11d (max 74Mbps, max 54Km)

Le frequenze assegnate (3,4 e 3,6GHz) erano utilizzate al Ministero della Difesa (sistemi Radar) e verranno assegnate al Ministero delle Comunicazioni che venderà ai provider mediante asta pubblica.

L'asta si è conclusa il 27 febbraio 2008 con l'assegnazione di tutte le licenze, con un incasso per l'erario di poco superiore ai 136 milioni di euro

Principali providers WiMAX in Italia:

- ▶ [Linkem](#)
- ▶ [WiMore](#)

Bluetooth e 802.15

Nasce nel 1999 dall'associazione tra Sony-Ericsson, IBM, Toshiba e Nokia con l'obiettivo di realizzare un sistema di comunicazione senza fili tra dispositivi digitali (cellulari, portatili, macchine fotografiche ecc..) e le loro periferiche.

Sistema a bassa potenza, raggio 10 m, banda ISM a 2.4 Ghz (no licenza)

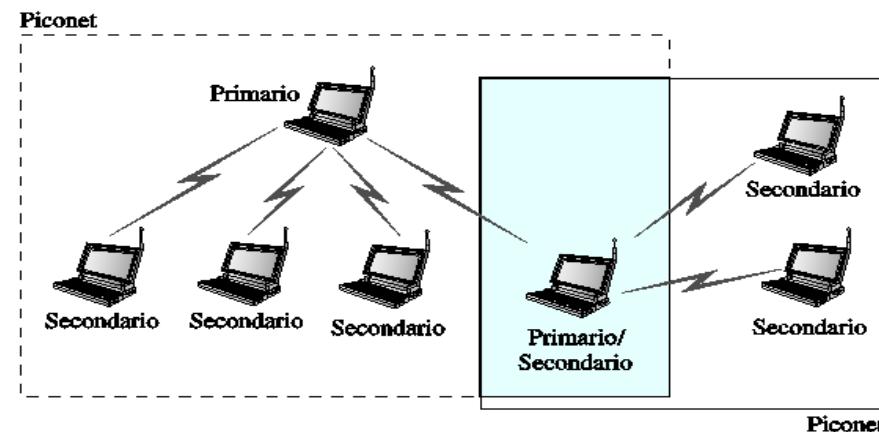
La banda è suddivisa in 79 canali di 1 MHz (1 bit per Hz -> max 1 Mbps) e utilizza Frequency Hopping Spread Spectrum (FHSS, come 802.11)

Un dispositivo Bluetooth Master può gestire simultaneamente la comunicazione fino a 7 nodi slave (fino a 255 parked nodes) entro un raggio di 10 metri, realizzando una "Piconet". Due nodi Slave non possono comunicare tra loro.

Due Piconet possono essere collegate tra loro mediante un nodo condiviso, formando una rete allargata denominata ScatterNet.

802.15 è una emanazione di IEEE in cui è definita la **WPAN (Wireless Personal Area Network)** basata su Bluetooth v1.1.

Attualmente sono definiti solo lo strato fisico e data-link.
L'utilizzo di IPv4 su WPAN è scarso.



Il progetto [6toWPAN](#) (IPv6 over Low-Power WPAN) mira ad un utilizzo più diffuso di dispositivi WPAN in IoT (Internet of Things).

RFID

RFID (Radio Frequency IDentification) è una tecnologia per l'identificazione wireless di diverse tipologie di oggetti quali smartcard, passaporti, libri e per la tracciabilità di animali.

RFID non è integrata nei protocolli utilizzati dalle reti di calcolatori e ha molte varianti, tra cui l'EPC (electronic product code).

L'RFID EPC ha le stesse funzionalità del codice a barre, migliorandone la praticità di utilizzo poiché è leggibile elettronicamente fino a 10 metri, anche quando non è visibile.

RFID EPC ha due componenti: TAG e Lettore:

I TAG sono dispositivi piccoli ed economici con un identificatore univoco a 96 bit e una piccola quantità di memoria che può essere letta o scritta da un lettore RFID.

La memoria può essere utilizzata ad esempio per tracciare gli spostamenti attraverso una catena di approvvigionamento.

Un TAG contiene inoltre un circuito integrato con una piccola antenna e può essere senza batteria (prende energia dalle trasmissioni radio del lettore) o con batteria.

Il compito principale dei Lettori è di fare un inventario dei TAG presenti nelle vicinanze

Low Power LAN

Sono reti Wireless progettate per comunicazioni a lunga distanza e basso costo / basso consumo di un basso tasso di bit (da 0.3 a 50 Kbps).
L'utilizzo tipico è per sensori nell'ambito dell'IoT.

Esistono diverse tecnologie LPLAN tra cui Sigfox e LoRa e NB-IoT

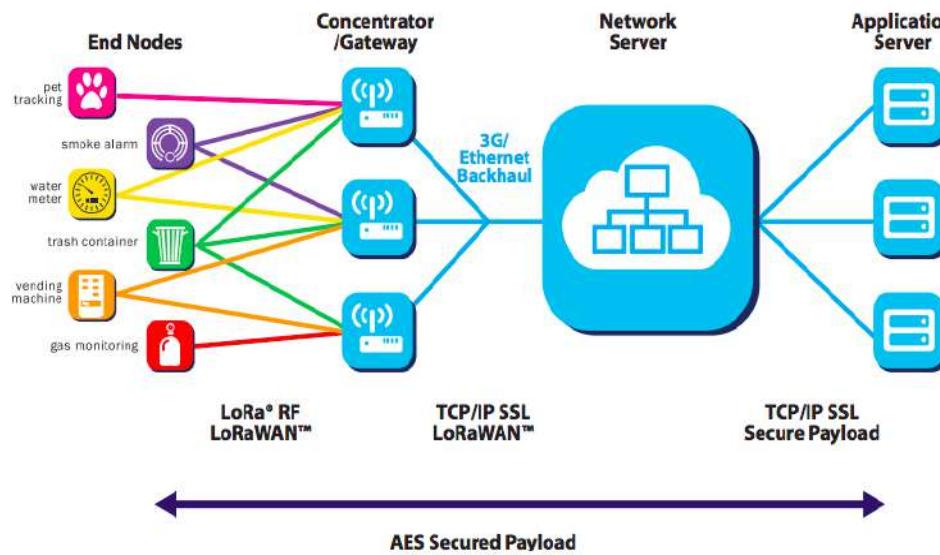
Caratteristica	Sigfox	LoRaWAN	NB-IoT
Modulazione	BPSK	CCS	QPSK
Frequenza	Banda libera 868 Mhz EU 915 Mhz USA 433 Mhz Asia	Banda libera 868 Mhz EU 915 Mhz USA 433 Mhz Asia	Banda licenziata LTE
Larghezza di Banda	100 Hz	250 kHz – 125 KHz	200 KHz
Data rate massimo	100 bps	50 kbps	200 kbps
Bidirezionale	Limitato /Half duplex	Si /HalfDuplex	Si /HalfDuplex
N° max di messaggi giornalieri	140	Illimitati	Illimitati
Range	10 km area urbana 40 km area rurale	5 km area urbana 20 km area rurale	1 km area urbana 10 km area urbana
Immunità a interferenze	Molto alta	Molto alta	Bassa
Autenticazione e cifratura	No	Sì	No
Adaptive data rate	No	Sì	No
Localizzazione	Sì	Sì	No
Reti private	No	Sì	No
Standardizzazione	Sigfox company	LoRa-Alliance	3GPP

LoRa

LoRa (Long Range) è una tecnologia wireless a lungo raggio (fino a 50Km) e bassa potenza, ideata per la connessione sicura di dispositivi IoT (geolocalizzazione, sensori RFID, ecc).

E' composta da due livelli.

Il primo livello: physical layer (PHY) ovvero lo strato fisico che utilizza una modulazione proprietaria derivata dal Chirp Spread Spectrum (CSS) su frequenze ISM.
secondo livello: protocollo per il livello MAC (Media Access Control) chiamato LoRaWAN.





UNIVERSITÀ
DI PARMA

DIPARTIMENTO DI SCIENZE MATEMATICHE, FISICHE ED INFORMATICHE
Corso di Laurea in Informatica

Il Livello Network

Parte I : IPv4

RETI DI CALCOLATORI - a.a. 2023/2024

Roberto Alfieri

Il livello Network: sommario

PARTE I

- ▶ Scopi del livello Network
- ▶ Commutazione di circuito e di pacchetto
- ▶ La famiglia dei protocolli TCP/IP
- ▶ Il protocollo IPv4: trama indirizzi, instradamento
- ▶ Protocolli di servizio per IPv4: ARP, ICMP, DHCP

PARTE II

- ▶ IPv6

PARTE III

- ▶ Algoritmi e protocolli di routing
- ▶ Distance Vector e Link State.

RIFERIMENTI

Reti di Calcolatori, A. Tanenbaum, ed. Pearson

Reti di calcolatori e Internet, Forouzan , Ed. McGraw-Hill

Scopi e servizi del livello Network

Estendere i servizi che il livello Data-Link offre a macchine connesse anche a macchine che non hanno una connessione diretta.

Compito fondamentale dello strato di rete è trasportare i pacchetti lungo tutto il percorso dal mittente al destinatario, attraversando tutti i nodi di transito dove sono possibili scelte alternative per le linee di uscita.

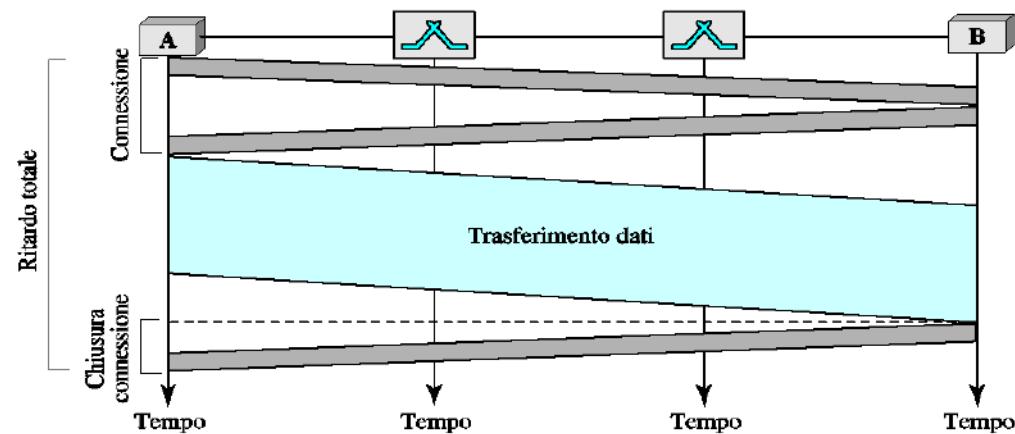
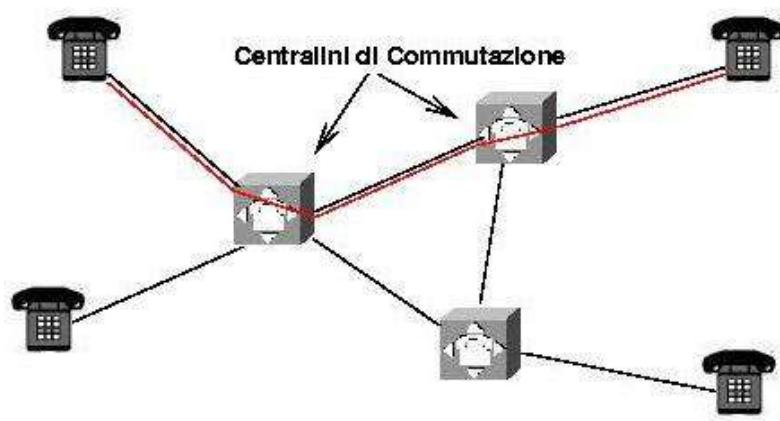
La funzione di collegamento di una linea di ingresso con una di uscita opportunamente scelta, che viene svolta nei nodi è detta **Commutazione** (Switching)

Le due tecniche di commutazione usate tradizionalmente nelle reti sono:

- **Commutazione di circuito** (utilizzata storicamente dai Provider di telefonia)
- **Commutazione di pacchetto** (utilizzata nelle reti di calcolatori)

Commutazione di Circuito

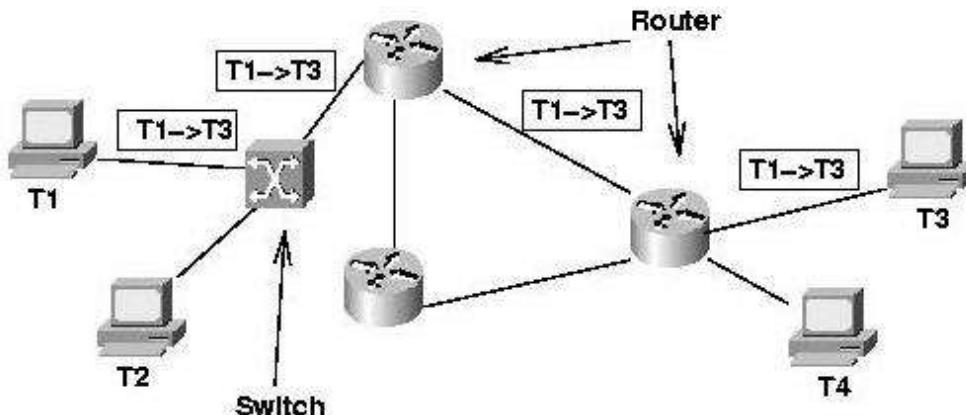
- ▶ I nodi di transito sono i Centralini di Commutazione (Manuali, Meccanici o Elettronici)
- ▶ L'algoritmo per la commutazione interviene all'apertura del canale fisico
- ▶ Nella fase di connessione vengono allocate le risorse necessarie
- ▶ Ritardo: è minimo nel trasferimento dati, ma è elevato in fase di apertura e chiusura della connessione
- ▶ Efficienza: le risorse allocate sono riservate anche se la connessione è inutilizzata. Questo non avviene per le telefonate, ma può avvenire per il trasferimento dati.



Commutazione di Pacchetto

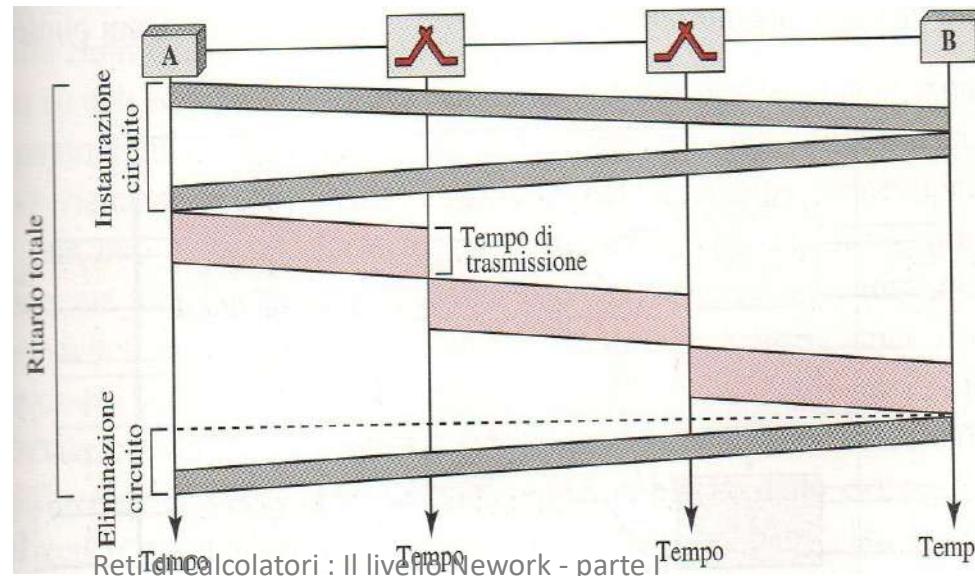
Commutazione di Pacchetto

- ▶ Comunicazione frazionata in “pacchetti”
- ▶ Algoritmo per la commutazione interviene sui pacchetti
- ▶ Esistono diversi tipi di nodi di transito a seconda della loro funzione:
 - Hub, Bridge, Switch, Router o Gateway.
- ▶ Esistono 2 tipologie di commutazione a pacchetto:
 - A circuito virtuale
 - A datagramma



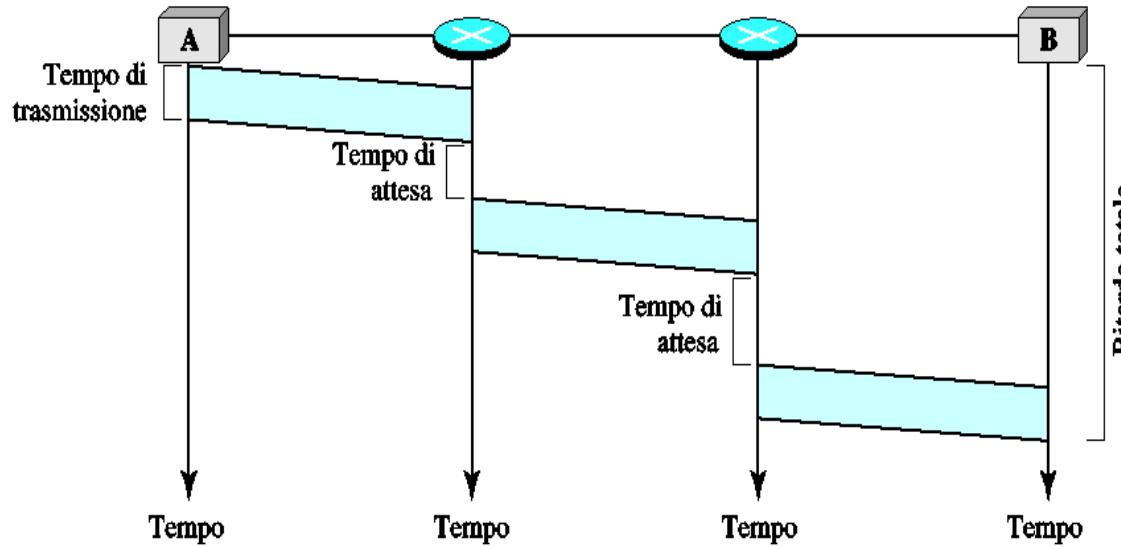
Commutazione di pacchetto a circuito virtuale

- ▶ Algoritmo per la commutazione interviene solo all'inizio per l'apertura del Canale Virtuale (VC).
- ▶ Ad ogni nuovo VC viene assegnata una etichetta; ogni router viene marcato con l'etichetta del VC e la relativa porta di uscita.
- ▶ I pacchetti seguono il percorso individuato
- ▶ Implementazioni principali:
 - ATM. E' la rete che utilizza la commutazione di pacchetto a circuito virtuale per la telefonia.
 - In internet è possibile creare isole a circuito virtuale con il protocollo MPLS
 - La versione 6 di IP supporta (anche) reti a circuito virtuale.



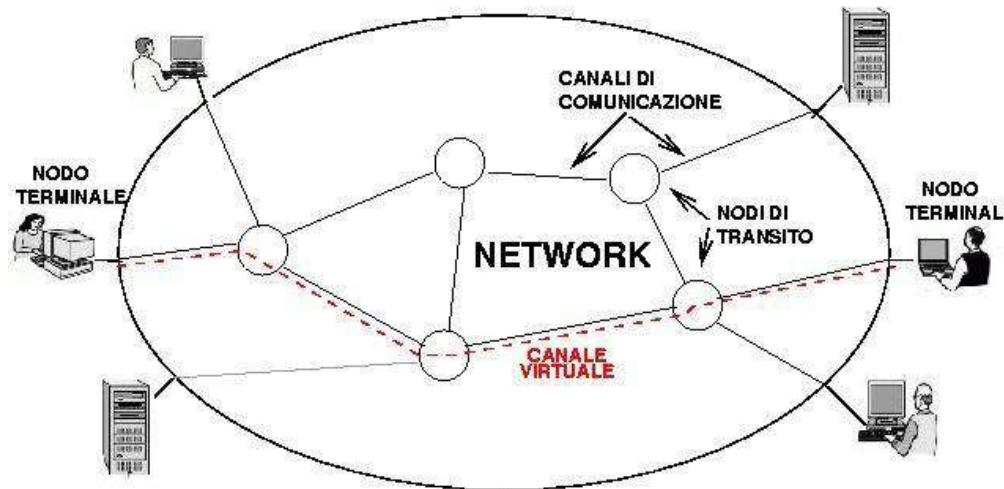
Commutazione di pacchetto a datagramma

- ▶ I pacchetti sono instradati in modo indipendente in base all'indirizzo di destinazione
- ▶ L'instradamento è determinato dai router attraversati in base “**tabelle di instradamento**” che ogni router costruisce dinamicamente mediante gli “**algoritmi di routing**”.
- ▶ Pacchetti della stessa connessione possono seguire strade diverse.
- ▶ Implementazioni principali: IPv4 e IPv6.



Routing

Il routing è quella parte del software dello strato Network che si preoccupa di dell'instradamento dei pacchetti in transito.



- ▶ Se la Rete è a **Datagramma** il routing viene determinato per ogni pacchetto, poiché il percorso migliore può cambiare nel tempo.
- ▶ Se la Rete è a **Circuito Virtuale** il routing viene determinato al momento dell'attivazione del circuito. Da quel momento in poi tutti i pacchetti seguono il percorso stabilito.

Confronto tra i metodi di commutazione a pacchetto

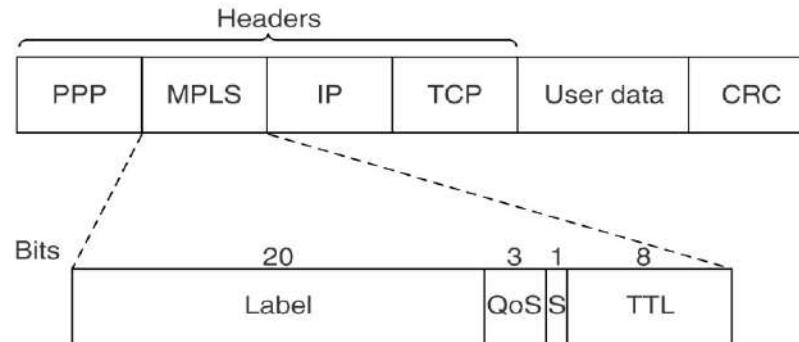
Issue	Datagram subnet	Virtual-circuit subnet
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

Reti a Circuito Virtuale: MPLS

MPLS (MultiProtocol Label Switching) consente di creare in Internet aree a commutazione di Label.

E' uno strato che si pone sotto il livello rete aggiungendo un proprio Header di 4 byte tra l'header di livello rete (IP) e quello di livello dati (ppp o Ethernet). Per questo può essere considerato un protocollo di livello 2.5.

I campi principali sono la Label (20bit), QoS, e TTL.

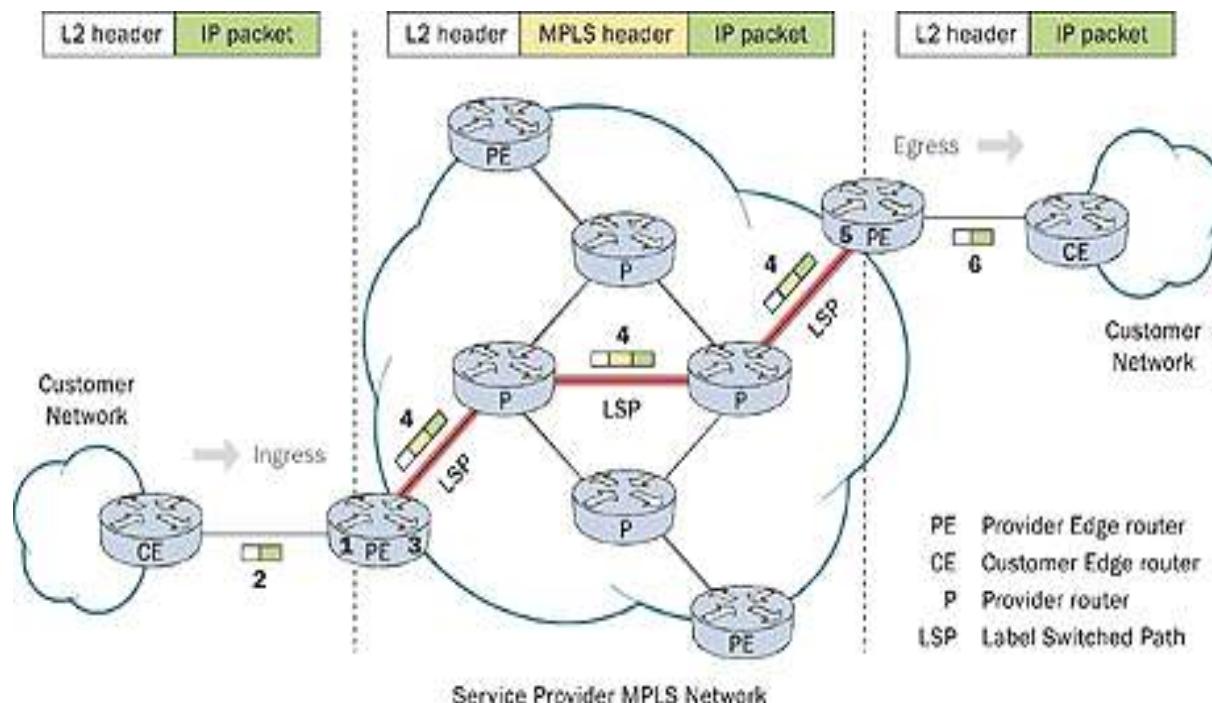


Vantaggi: QoS, Traffic Shaping (e-mail, web, ...), VPN.

Il routing MPLS

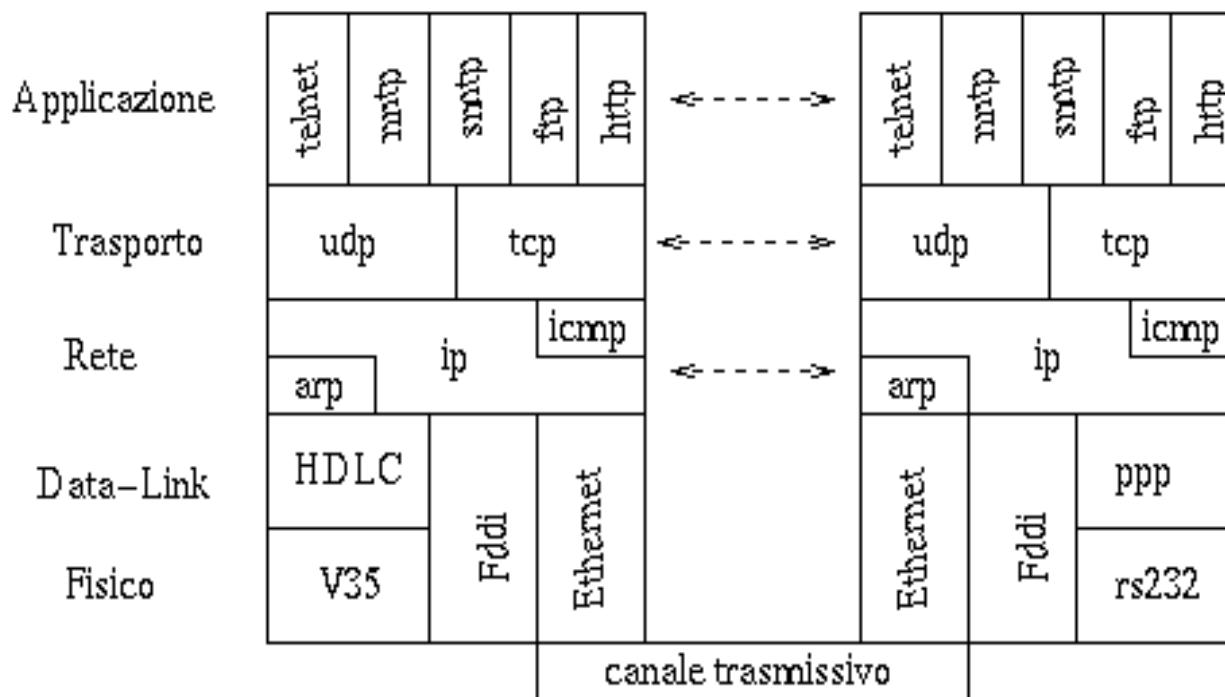
Richiede al proprio interno Router specifici che supportano il protocollo.
Il router di frontiera (Edge) determina il percorso e aggiunge l'header MPLS al pacchetto.

Attraverso le etichette il primo pacchetto definisce un “tunnel” nella rete MPLS.
I pacchetti successivi della stessa connessione seguono il percorso del primo.

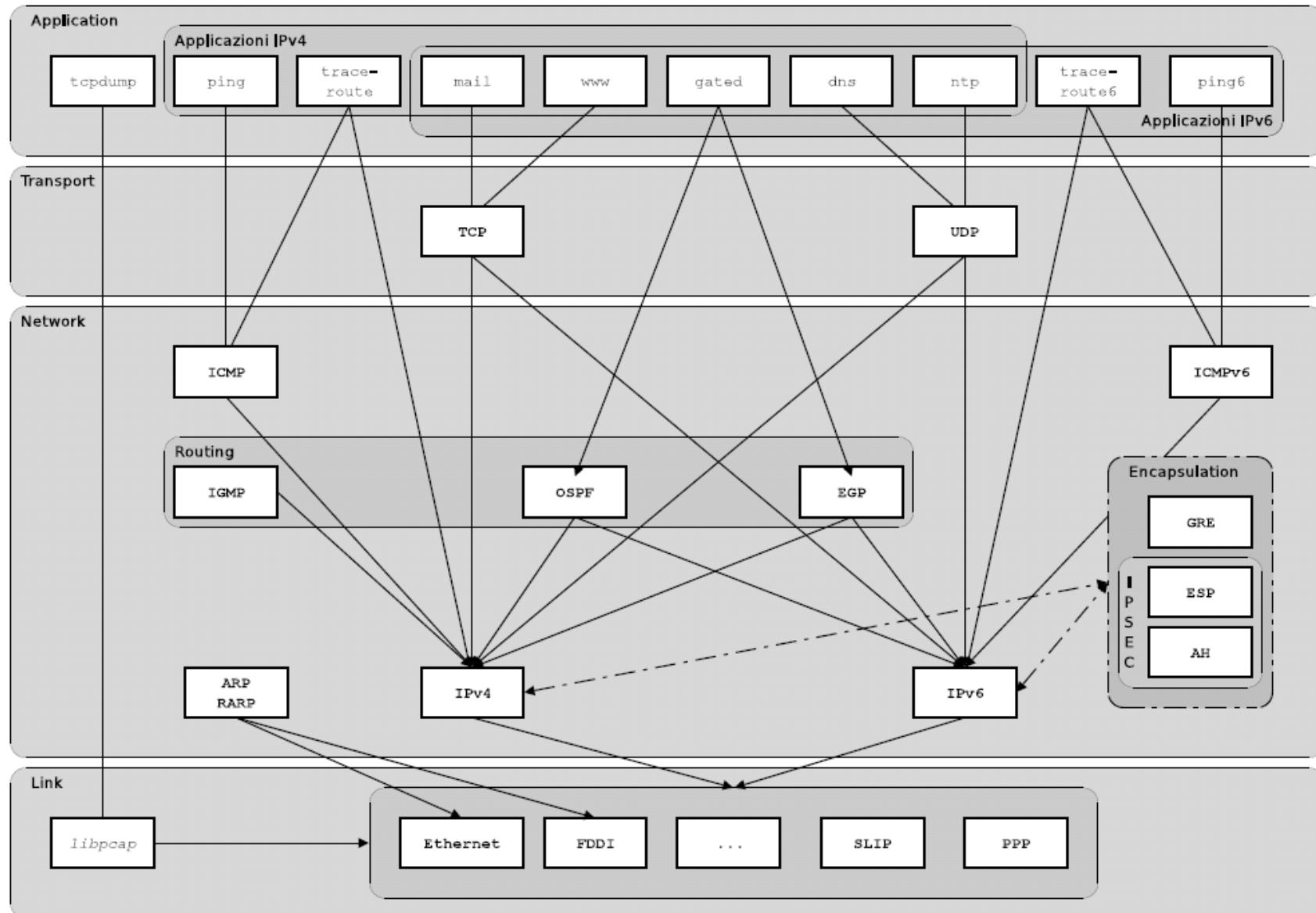


La famiglia dei protocolli TCP/IP

- ▶ Nessuna specifica per gli strati sotto IP, in quanto relativi alla singola sottorete.
- ▶ IP svolge funzioni di rete e instradamento dei pacchetti
- ▶ TCP (o UDP) svolge le funzioni di trasporto e di controllo della connessione end-to-end
- ▶ Lo strato di applicazione contiene applicativi utilizzati per fornire servizi all'utente



Quadro generale dei protocolli TCP/IP (da Gapil)



Gli Standard di Internet : Internet Society, RFC

Non esistono veri e propri enti che svolgono la funzione di gestione, ma solo enti di coordinamento delle attività di ricerca e di sviluppo che ora convergono nella **Internet Society**.

Dalla **IS** dipende l'**Internet Advisory Board** (IAB) e si compone di due sottogruppi:

- ▶ **Internet Research Task Force (IRTF)**: coordina le attività di ricerca
- ▶ **Internet Engineering Task Force (IETF)**: coordina le attività di ingegnerizzazione ed implementazione
 - IETF pubblica nei **Request For Comment (RFC)** - <http://www.ietf.org/rfc.html>
 - Tipi di RFC
 - Informational (FYI)
 - Best Current Practice (BCP)
 - Standard (STD) 3 stati : **Proposed Standard, Draft Standard, Standard**

Gli Standard di Internet : ICANN , IANA

ICANN (Internet Corp. for Assigned Names and Numbers - <http://www.icann.org/>) è l'ente no-profit che assegna gli indirizzi IP e l'identificatore di protocollo e gestisce il DNS di primo livello (Top-Level Domain).
Funzione svolta operativamente da IANA (www.iana.org) che è una sua emanazione.

Data la complessità della gestione, sono state individuate 5 organizzazioni denominate RIR (Regional Internet Registries) che in cooperazione con IANA hanno il compito di gestire le allocazioni a livello continentale.

Le RIR sono: ARIN, APNIC, RIPE, LACNIC e AFRINIC.



Principi architetturali di Internet

Descritti nell'RFC 1958 <http://www.ietf.org/rfc/rfc1958.txt>, in ordine di importanza:

- 1) Assicurarsi che funzioni
- 2) Mantenerlo semplice (nel dubbio, la soluzione più semplice)
- 3) Fare scelte chiare (se si può fare in diversi modi sceglierne uno)
- 4) Sfruttare la modularità
- 5) Aspettarsi l'eterogeneità
- 6) Evitare opzioni e parametri statici
- 7) Mirare ad un buon progetto (non necessariamente perfetto)
- 8) Essere rigorosi nell'invio e tolleranti nella ricezione
- 9) Pensare alla scalabilità (IPv4 e IPv6..)
- 10) Considerare le prestazioni e i costi

IP: Lo stato Network in Internet

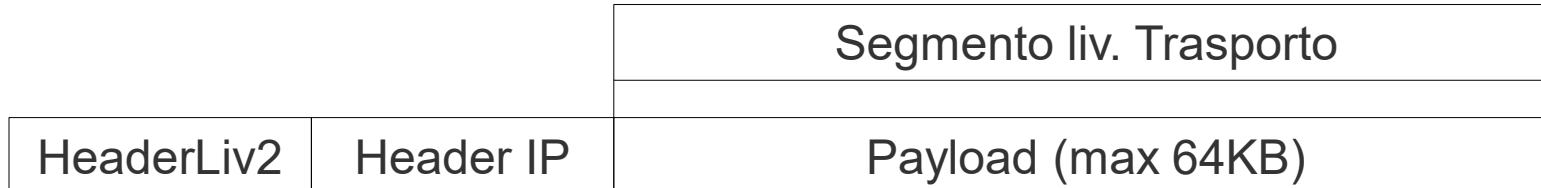
Interconnette più reti di livello Data-Link (LAN o connessioni punto-punto).
Fornisce uno servizio per il trasporto di datagrammi (pacchetti) tra mittente e destinatario indipendentemente dalle loro reti di appartenenza - <http://www.ietf.org/rfc/rfc791.txt>
La versione del protocollo IP attualmente in uso, descritta in queste slides, è IPv4.

Operazioni:

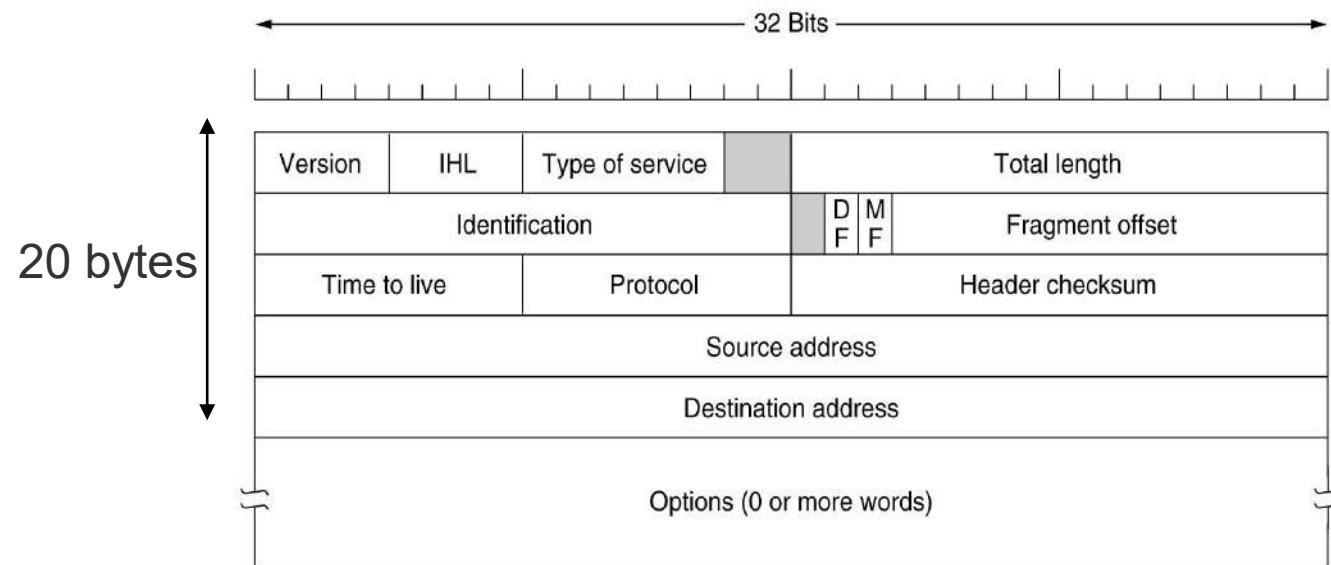
- ▶ Lo stato di trasporto prende il flusso di dati e li **divide in datagrammi** che passa allo stato IP. La dimensione massima è di 64KB, ma generalmente vengono scelti datagrammi non superiori a 1500 Byte (per compatibilità con Ethernet).
 - ▶ Il datagramma di trasporto (detto Segmento) **viene incorporato nella Trama IP** e trasferito da un router all'altro fino a destinazione.
 - ▶ Il datagramma può subire una **frammentazione** nel caso di passaggio attraverso un livello data-link con dimensione massima (MTU) inferiore. I frammenti vengono riassemblati a destinazione.
- Alcuni MTU (byte): 802.3=1500, 802.11=2312, PPP=576(tipico), FibreChannel=2112
- ▶ Il datagramma (eventualmente riassemblato) viene **estratto dalla trama IP** e passato al livello di trasporto, che **ricostruisce il flusso**.
 - ▶ **QoS:** La consegna è di tipo “**Best Effort**”.

La trama IP (1/3)

Il datagramma IP è costituito dall'intestazione (header) IP seguita dal segmento del livello di trasporto.



L'header ha una parte fissa e una parte opzionale variabile e viene trasmessa in ordine big endian.



La trama IP (2/3)

- ▶ **Version (4 bit):** i primi 4 bit di ogni pacchetto IP contengono il numero di versione.
- ▶ **HLEN (4 bit) :** dimensione dell'header espressa in parole di 4 byte (da 5 a 15)
- ▶ **Type of Service (6 bit):**
 - Inizialmente per controllo della rete (priorità e segnalazioni).
 - Con l'RFC 2474 diventa Servizi Differenziati per la codifica delle Classi di Servizio.
 - In realtà Internet è “best effort”: questo campo è quasi sempre inutilizzato.
- ▶ **Total Length (16 bit):** Numero di byte totali header+dati (fino a 64K)
- ▶ **Identification (16 bit):** Tutti i frammenti di datagramma hanno lo stesso valore
- ▶ **DF (1 bit):** Don't Fragment → ordina ai router di non frammentare
- ▶ **MF (1 bit):** More Fragments → 1 per tutti i frammenti tranne l'ultimo
- ▶ **Fragment Offset (13 bit):** Indica la posizione del frammento nel datagramma corrente, espressa in blocchi di 8 byte (max. 8192 frammenti).
- ▶ **TTL (8 bit):** Numero max di salti; si decrementa ad ogni passaggio.
Quando arriva a 0 il pacchetto viene eliminato.

La trama IP (3/3)

- ▶ **Protocol (8 bit):** Protocollo di livello superiore (ICMP=1, TCP=6, UDP=17, ...)
- ▶ **Header Checksum (16 bit):** Checksum dell'**header**
 - ricalcolato da ogni router , perché il TTL cambia ad ogni salto.
 - Aiuta a rilevare errori generati da locazioni di memoria difettose nei router.
 - Somma tutte le sequenze di 16 bit (con l'aritmetica del complemento a 1) e poi prende il complemento a 1 del risultato.
- ▶ **Source e Destination Address (32+32 bit):** Indirizzi di sorgente e destinazione
- ▶ **Options:** Pensato per poter aggiungere estensioni non previste.
Lista completa: <http://www.iana.org/assignments/ip-parameters>
Formato: Opt. code (1 byte) - Opt. Length (1byte) - Opt. Data (n Byte)
 - 0 - End of Option List
 - 130 - Security: lista di reti vietate, non usato.
 - 7 - Route record: ogni router aggiunge il proprio indirizzo
 - 68 - Time Stamp: ogni router aggiunge il proprio indirizzo e data/ora.
 - 137 - Source routing: lista dei router da percorrere
- ▶ **Padding:** bit aggiunti per rendere il campo Options multiplo di 32 bit.

Indirizzi IP

Indirizzi a 32 bit con notazione “**dotted decimal**”: 4 decimali (0-255) separati da punto

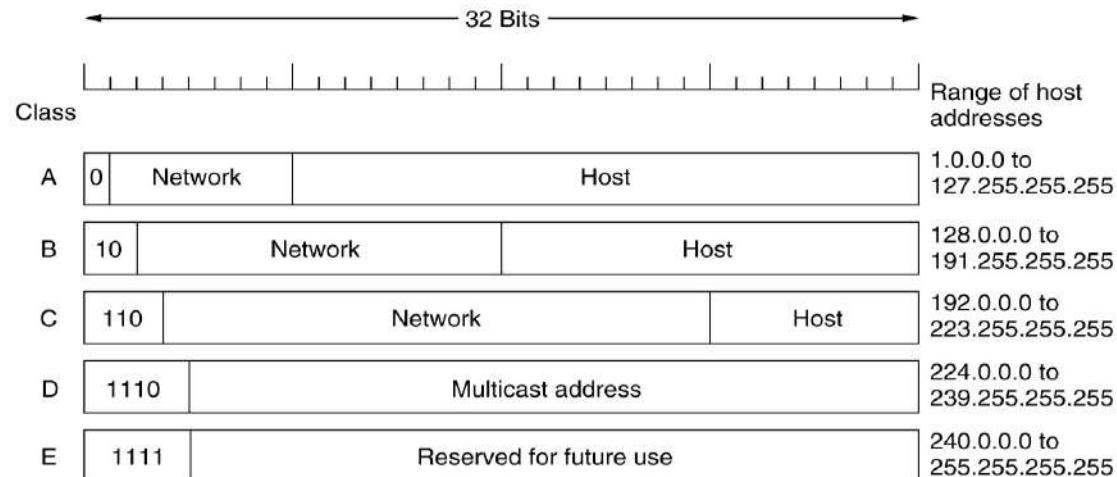
Esempio: $0x89CCD401 = 10001001.11001100.11010100.00000001 \rightarrow 137.204.212.1$

Numero max. di indirizzi $2^{32} = 4.294.967.296$

Per motivi di routing la sequenza è suddivisa in due parti:

- **NETid**: Identifica una Rete liv.2 Utilizzata dai router per l'instradamento dei pacchetti
- **HOSTid**: Distingue gli Host della stessa Rete.

Classfull Addressing:



Classi IP

Classe	bit-iniziali	inizio	fine	indirizzi	default-mask	CIDR-equiv	reti	host
A	0	0.x.x.x	127.x.x.x	2G	255.0.0.0	/8	126	16M
B	10	128.x.x.x	191.x.x.x	1G	255.255.0.0	/16	16K	64K
C	110	192.x.x.x	223.x.x.x	0.5G	255.255.255.0	/24	2M	254
D	1110	224.x.x.x	239.x.x.x	0.25G		Multicast		
E	1111	249.x.x.x	255.x.x.x	0.25G		Reserved		

La numerazione è gestita da [IANA](#) che delega gerarchicamente alle RIR:



Vedi ad esempio il comando: whois 160.78.0.0

Indirizzi IP di rete e di Broadcast

```
< network > 00000000000000000000  
< network > 11111111111111111111
```

Address of the network
Broadcast of a specific network

Se la parte Host è di N bit, il numero di indirizzi effettivamente assegnabili agli host è $2^N - 2$, poiché il primo indirizzo (tutti zeri nella parte host) identifica la rete, mentre l'ultimo indirizzo (tutti uni nella parte host) è l'indirizzo di broadcast.

Indirizzi IP per uso privato

Le seguenti reti sono riservate da ICANN per uso privato (Intranet) e gli indirizzi non possono essere annunciati dai Router (<http://www.ietf.org/rfc/rfc1918.txt>):

Classi	inizio	Fine	indirizzi
1 classe A	10.x.x.x		16M
16 classi B	172.16.x.x	172.31.x.x	1M
255 classi C	192.168.0.x	192.168.255.x	64K

LOOPBACK

La rete 127.0.0.0/16 è riservato per il loopback (RFC 3330).

Per convenzione su ogni host viene definita una interfaccia virtuale di loopback con indirizzo IP predefinito 127.0.0.1, con nome **localhost**, che consente la comunicazione TCP/IP tra due processi locali senza il coinvolgimento di interfacce fisiche.

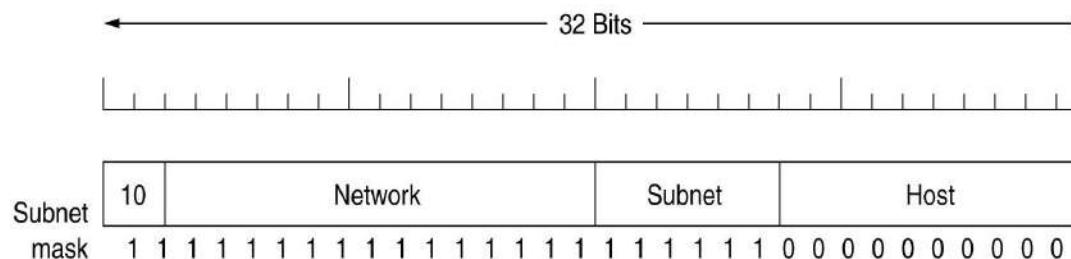
ZEROCONF

La rete 169.254.0.0/16 (IPv4 link-local) è utilizzata dal servizio Zeroconf (<http://www.ietf.org/rfc/rfc3927.txt>) per assegnare un indirizzo IP agli host di una LAN senza dipendere da una infrastruttura, ovvero quando non è possibile ottenere un indirizzo dinamico da un server DHCP.

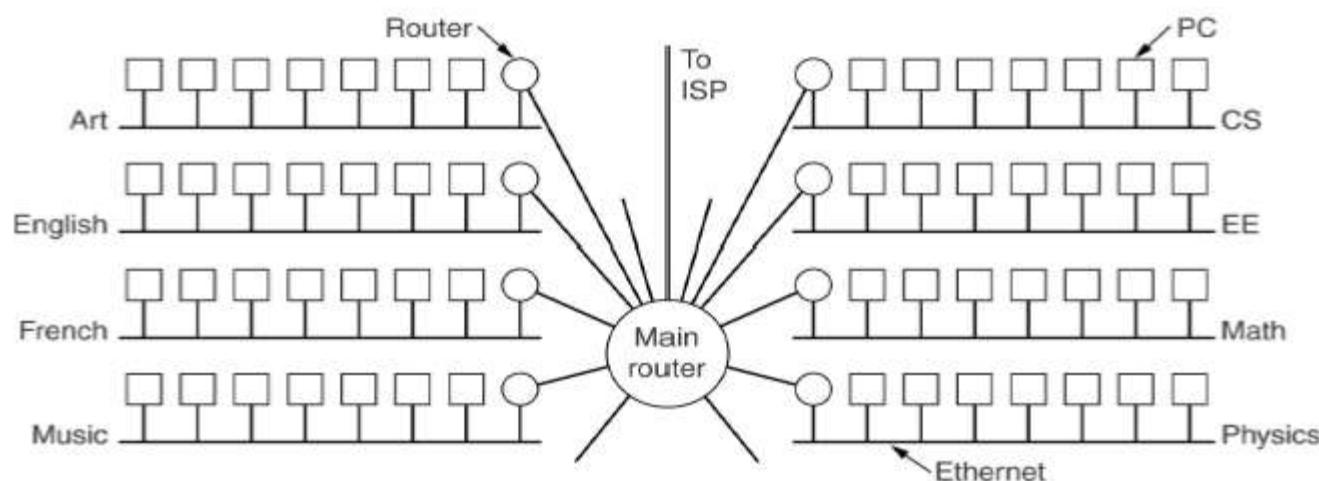
IP subnetting

Consente un ulteriore livello di gerarchia per gli indirizzi IP: NET-SUBNET-HOST
Il **NETMASK** è un parametro di 32 bit che stabilisce la suddivisione:

- ▶ Bit a 1 in corrispondenza del campo NET o SUBNET
- ▶ Bit a 0 in corrispondenza del campo HOST



Esempio: rete di classe B 160.78.0.0 partizionata in 256 Subnet da 256 indirizzi:
NETMASK 255.255.255.0 -> 11111111 11111111 11111111 00000000



CIDR – Classless Inter-Domain Routing

- Il numero di indirizzi IP (4G) è insufficiente
- Molte reti di classe B usano meno 50 indirizzi

CIDR <http://www.ietf.org/rfc/rfc1519.txt>

- soluzione temporanea in attesa di IPv6
- Assegna gli indirizzi IPv4 rimanenti in blocchi di dimensione variabile nella forma
netaddress/NetMaskBit
- routing più complicato (tabelle lunghe)

Il Supernetting (route aggregation)

consente di accorpare più reti contigue come fossero un'unica rete, per ottimizzare i tempi di routing

- ▶ In caso di sovrapposizioni tra 2 reti vince la netmask più lunga

no. of addrs	bits	pref	mask
1	0	/32	255.255.255.255
2	1	/31	255.255.255.254
4	2	/30	255.255.255.252
8	3	/29	255.255.255.248
16	4	/28	255.255.255.240
32	5	/27	255.255.255.224
64	6	/26	255.255.255.192
128	7	/25	255.255.255.128
256	8	/24	255.255.255
512	9	/23	255.255.254
1 K	10	/22	255.255.252
2 K	11	/21	255.255.248
4 K	12	/20	255.255.240
8 K	13	/19	255.255.224
16 K	14	/18	255.255.192
32 K	15	/17	255.255.128
64 K	16	/16	255.255
128 K	17	/15	255.254
256 K	18	/14	255.252
512 K	19	/13	255.248
1 M	20	/12	255.240
2 M	21	/11	255.224
4 M	22	/10	255.192
8 M	23	/9	255.128
16 M	24	/8	255
32 M	25	/7	254
64 M	26	/6	252
128 M	27	/5	248
256 M	28	/4	240
512 M	29	/3	224
1024 M	30	/2	192

Instradamento dei Datagrammi

La rete di appartenenza di un Host è fondamentale per determinare la modalità di consegna, che può essere **diretta** o **indiretta**.

Direct delivery : host sorgente e destinatario condividono la stessa rete.

- trova l'indirizzo fisico del **destinatario** (con ARP) che associa all'IP del destinatario
- inoltra il pacchetto al livello Link indirizzando il destinatario:

1)	ARPrequest	to:Broadcast from: MACmitt	Who has IPdest?
2)	ARPreply	to:MACmitt from: MACdest	
3)	Send IP	to:MACdest from: MACmitt	toIP:dest, fromIPmitt

Indirect delivery : sorgente e destinatario appartengono a reti IP diverse

- individua il router da contattare consultando la propria **Tabella di Routing**
- trova l'indirizzo fisico del **router** (con ARP) che associa all'IP del destinatario
- inoltra il pacchetto al livello Link indirizzando il router.

1)	ARPrequest	to:Broadcast from: MACmitt	Who has IProuter?
2)	ARPreply	to:MACmitt from: MACrouter	
3)	Send IP	to:MACrouter from: MACmitt	toIP:dest, fromIPmitt

La scelta del tipo di consegna avviene consultando la tabella di routing locale.

Tabella di routing

E' una tabella che contiene le destinazioni e i percorsi per raggiungerle.

Esempio di tabella di routing (comando "route" di linux) per l'host 160.78.124.1:

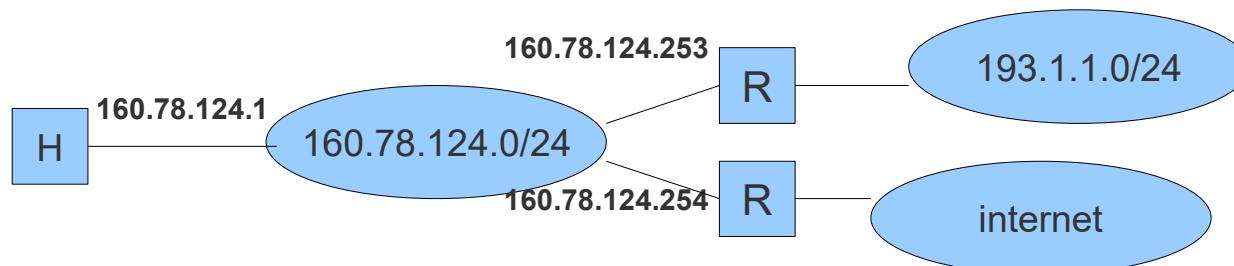
Destination	Router	Mask	Interface
160.78.124.0	*	255.255.255.0	eth0 (consegna diretta)
193.1.1.0	160.78.124.253	255.255.255.0	eth0 (consegna indiretta)
default	160.78.124.254	0.0.0.0	eth0 (consegna indiretta)

La prima riga (Router *) indica che gli host della rete 160.78.124.0/24 vengono raggiunti in consegna diretta.

La seconda riga (Router 160.78.124.253) indica che gli host della rete 193.1.1.0/24 sono raggiunti in modalità indiretta tramite il router 160.78.124.253

Generalmente le reti locali hanno al proprio interno un router di riferimento (indicato come "**Default Router**") a cui vengono consegnate tutte le destinazioni non note.

Nell'esempio tutte le destinazioni diverse da 160.78.124.0/24 e 192.1.1.0/24 vengono consegnate al router 160.78.124.254.



Ricerca nella tabella

La ricerca avviene utilizzando

- l'IP di destinazione (IPdest)
- La rete di destinazione e Netmask (Mask) di ciascuna riga della tabella

Procedura: IPdest AND Mask

- Se il risultato coincide con la rete presa in esame la riga è quella giusta
- Una volta trovato il risultato il lookup si ferma e il datagramma viene instradato
- Se nessuna riga corrisponde si usa il router di default

NAT (Network Address Translation)

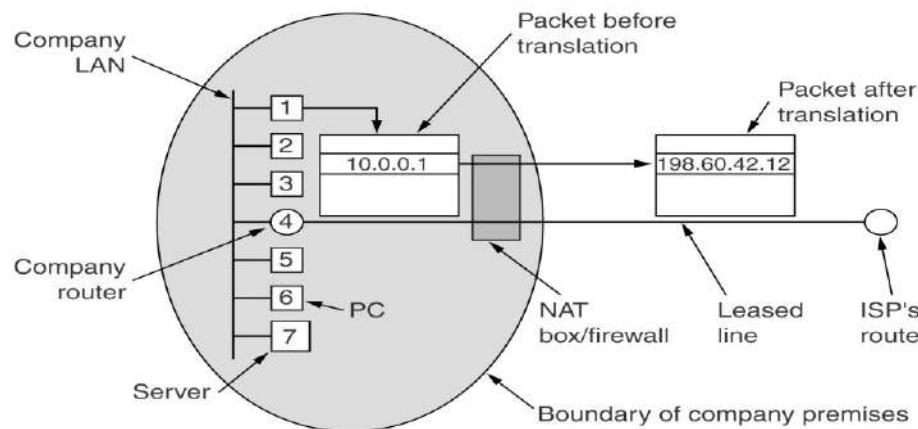
NAT (<http://www.ietf.org/rfc/rfc3022.txt>): dispositivo che consente agli host di una LAN (con indirizzi privati) di comunicare in Internet utilizzando un solo indirizzo pubblico. La linea verso Internet possiede un indirizzo IP pubblico e viene visto dagli host della LAN come Default Router.

Le operazioni del NAT sono distinte in base alla direzione:

SNAT (Source-NAT) è la funzionalità che consente di manipolare l'indirizzo sorgente ed è tipicamente utilizzato per consentire ai pacchetti di una LAN privata di uscire in internet. Quando un Host della LAN si rivolge al NAT per uscire in Internet il NAT trasforma l'indirizzo del mittente IP nell'indirizzo IP pubblico del NAT, quindi contatta il destinatario.

DNAT (Destination-NAT) è utilizzata per manipolare l'indirizzo di destinazione.

E' usata tipicamente per dirottare verso una destinazione interna (con indirizzo privato) i pacchetti provenienti da Internet.

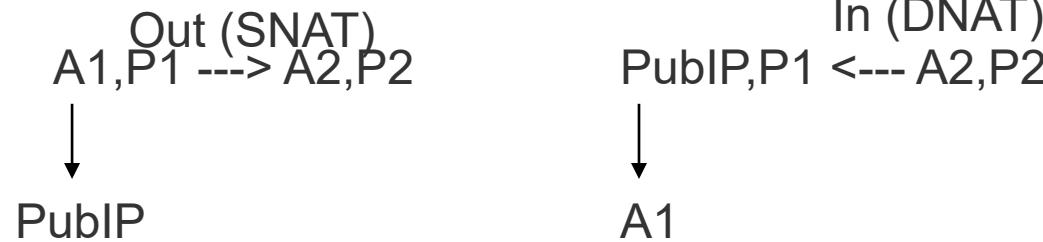


Le tabelle del NAT

Le manipolazioni SNAT e DNAT sono rappresentate in tabelle che vengono consultate per ogni pacchetto che attraversa il NAT. Le entry delle tabelle possono essere statiche o dinamiche.

NAT Dinamico

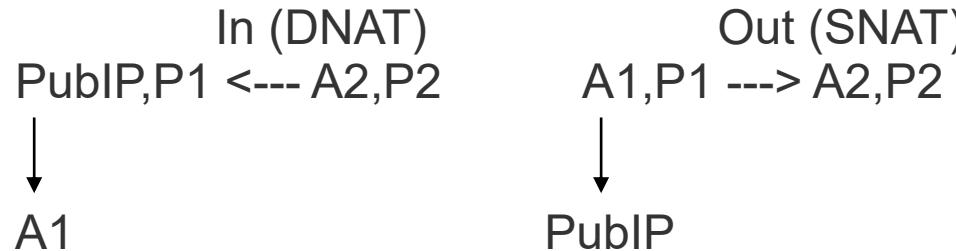
Cl. interno: A1,P1
Serv esterno: A2,P2



Quando un client della LAN si rivolge al NAT per contattare un server esterno, il NAT genera una entry dinamica associando IP/porta del client con la IP/porta del server quindi applica **SNAT**. L'entry viene utilizzata per il DNAT sulla risposta del server.

NAT Statico

Cl. esterno: A2,P2
Serv interno: A1,P1



Se vogliamo avere un server interno che deve essere contattato da un client esterno dobbiamo istruire il NAT mediante una entry statica che associa una porta del NAT con IP/porta del server interno.

Quando un client esterno contatta il NAT sulla porta viene consultata la entry statica e applicato **DNAT**. Viene inoltre creata una entry dinamica che verrà utilizzata per applicare SNAT sulla risposta.

Problemi dell'architettura NAT

- ▶ Violazione dell'univocità degli indirizzi: migliaia di Host usano gli stessi indirizzi privati.
- ▶ Sicurezza: è difficile tracciare l'identità dell'indirizzo IP pubblico.
- ▶ IP non è più connection-less
- ▶ IP non è più stratificato: Il Layer IP non dovrebbe entrare nei layer superiori
- ▶ Un guasto al NAT pregiudica tutte le connessioni che lo attraversano.

In realtà NAT ha avuto una grande diffusione e ha ridotto la spinta verso IPv6.

Protocollo ARP

Ogni interfaccia di rete di un nodo (Ethernet, LAN Wireless, Seriale, ecc) possiede un indirizzo fisico e, se utilizzata in internet, almeno un indirizzo IP.

Il protocollo **ARP (Address Resolution Protocol)** ha il compito di determinare l'indirizzo fisico di un nodo IP.

Quando un nodo mittente deve contattare un destinatario in **Direct Delivery** (Terminale o Router) di cui conosce solo l'indirizzo IP utilizzerà il protocollo ARP:

- ▶ Il nodo sorgente invia un pacchetto (**ARP Request**) con destinazione Broadcast sulla LAN, contenente l'indirizzo IP del destinatario.
- ▶ I terminali con indirizzo IP diverso ignoreranno il Pacchetto, mentre il nodo in oggetto risponderà (**ARP Replay**) con un Unicast inviando il proprio indirizzo fisico.
- ▶ Ogni host mantiene una tabella (**ARP Cache**) con le corrispondenze ottenute (comando arp –a). Ogni entry ha un tempo di vita tipicamente di 20 minuti.

Il Frame ARP contiene:

- ▶ Un campo codice 1=ARPPrequest, 2=ARPPreplay
- ▶ indirizzo IP e Indirizzo HW di partenza e destinazione

Protocollo RARP

In determinate situazioni alcuni nodi IP al momento dell'attivazione della rete non conoscono il loro indirizzo IP (ad esempio perché non hanno memoria permanente). Esistono diverse soluzioni, tra cui **RARP** (Reverse ARP) - <http://www.ietf.org/rfc/rfc903.txt>. E' un protocollo ideato da SUN per risolvere il problema.

Il client invia in modalità Broadcast la richiesta:

“Questo è il mio indirizzo MAC: xx-xx-xx-xx-xx-xx, Qualcuno conosce il mio indirizzo IP?”. Un server RARP, con la tabella MAC-IP , risponderà con l'informazione richiesta.

Svantaggi:

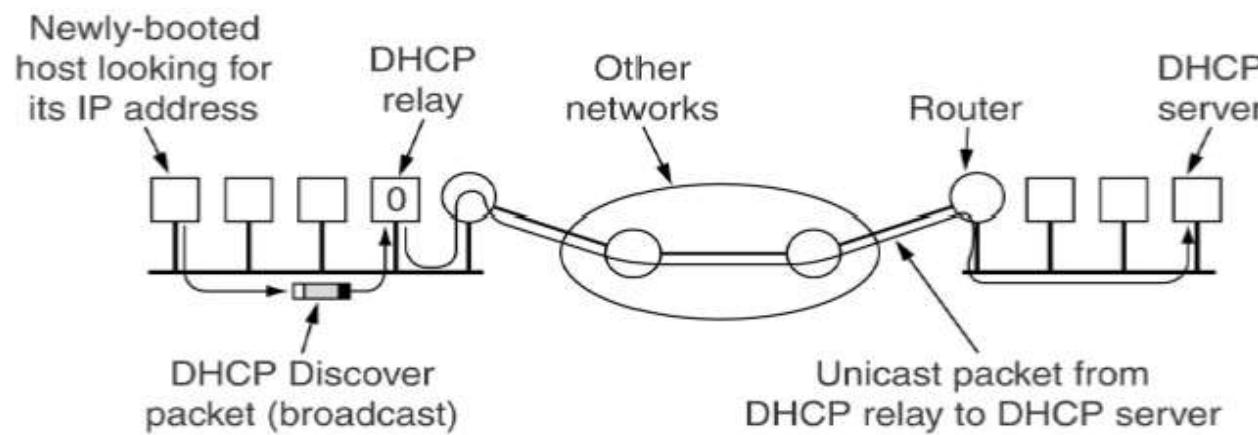
- ▶ la richiesta Broadcast non passa i router
- ▶ le associazioni MAC-IP sono statiche
- ▶ non sono previste altre informazioni

RARP è reso obsoleto dal suo successore DHCP.

Protocollo DHCP

DHCP (Dynamic Host Configuration Protocol, rfc2131.txt e rfc2132.txt) risolve lo stesso problema di RARP aggiungendo nuovi servizi.

- ▶ Il server DHCP può fornire più informazioni al client: indirizzo IP, NetMask, Default Router, DNS server, NTP server, ecc.
- ▶ L'indirizzo IP fornito può essere statico o dinamico (assegnato al momento della richiesta sulla base di un pool di indirizzi disponibili)
- ▶ Il server può risiedere in una LAN diversa dalla LAN del client (tramite relay)



E' un protocollo applicativo: utilizza la porta 67/UDP per il server e la 68/UDP per il client.

Funzionamento del DHCP

1) Il client DHCP invia in modalità broadcast

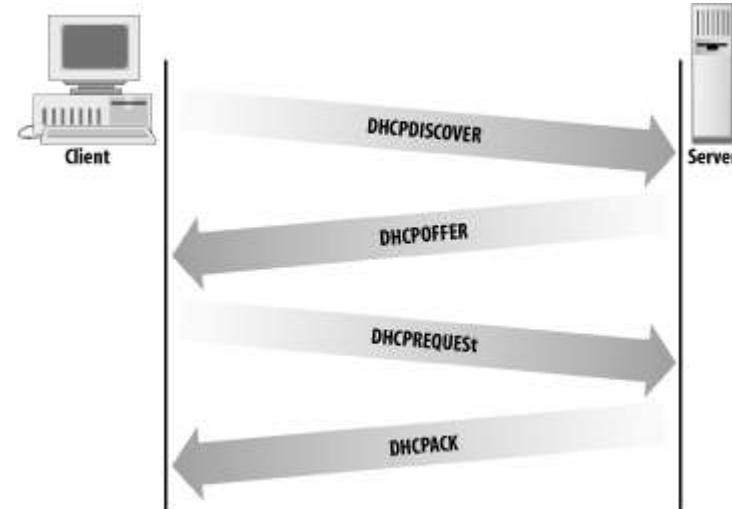
un pacchetto **DHCP Discover**

$0.0.0.0:68 \rightarrow 255.255.255.255:67$

2) Il server risponde (tramite l'eventuale Agent)

un pacchetto **DHCP Offer** che contiene l'indirizzo
richiesto più eventuali altre informazioni.

$\text{Ipserver:67} \rightarrow \text{Ipclient:68}$



3) Il client accetta la prima risposta che ottiene e

invia in Broadcast un **DHCP Request** in cui dice
da quale server ha ricevuto l'indirizzo.

$0.0.0.0:68 \rightarrow 255.255.255.255$

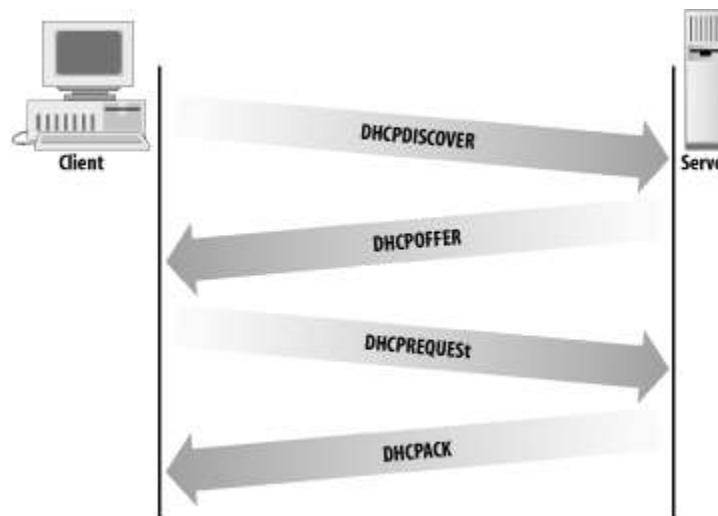
4) Infine il server manda un **DHCP ACK** al client per conferma.

$\text{Ipserver:67} \rightarrow \text{Ipclient:68}$

Funzionamento del DHCP: Rinnovo

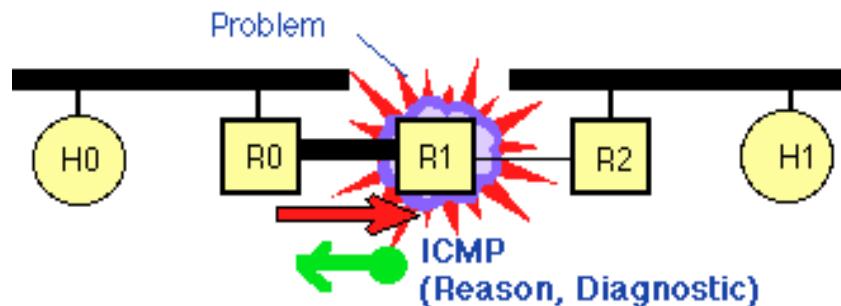
Il server gestisce una tabella in cui gli indirizzi IP possono essere associati staticamente a indirizzi MAC oppure possono essere “**affittati**” dinamicamente al momento della richiesta. Il “leasing” ha un termine; il client deve chiederne un eventuale rinnovo, altrimenti l’indirizzo viene ritirato ed assegnato ad un altro client.

Le operazioni **DHCPREQUEST/DHCPACK** vengono ripetute per prolungare l’assegnazione dell’indirizzo. La richiesta avviene con 3 tentativi: 2 volte al 50% del tempo utilizzato e un’ultima volta all’ 87,5%



Protocollo ICMP

ICMP - Internet Control Message Protocol (<http://www.ietf.org/rfc/rfc792.txt>) è un protocollo di servizio di IP per lo scambio di messaggi di errore o di controllo che consentono agli Host e ai Router di accorgersi di eventuali malfunzionamenti della rete. Spedisce i messaggi di notifica dell'errore sempre al mittente del datagramma per il quale si è verificato l'errore.

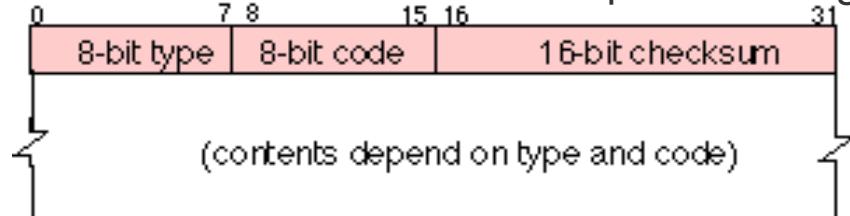


Formato del Frame ICMP

Il formato del frame è costituito da una intestazione e da un'area dati. La prima è composta da tre campi:

- **TIPO** è un numero di 8 bit che identifica il messaggio Tipi principali:
 - 0 = Risposta di ECHO
 - 3 = Destinazione irraggiungibile (esempio datagramma troppo grande, ma DF settato)
 - 4 = Rallentamento della sorgente (Il router informa che il pacchetto è stato eliminato e che la sorgente deve rallentare)
 - 8 = Richiesta di ECHO (comando ping)
 - 11 = TTL scaduto per un datagramma
 - 12 = Problema di parametri (argomento di un opzione scorretto)
 - 13 = Richiesta di contrassegno temporale (per sincronizzare gli orologi)
 - 14 = Risposta di contrassegno temporale
- **CODICE:** Info aggiuntive. Ad esempio se il Tipo è 3 il Codice dice qual'è il tipo di errore
- **CHECKSUM**, di 16 bit, è il CRC del frame ICMP (header+data)

L'area Dati varia in funzione del tipo di messaggio.



Il Frame ICMP è inserito direttamente nel payload di IP:





UNIVERSITÀ
DI PARMA

DIPARTIMENTO DI SCIENZE MATEMATICHE, FISICHE ED INFORMATICHE
Corso di Laurea in Informatica

Il Livello Network

Parte II : IPv6

RETI DI CALCOLATORI - a.a. 2023/2024

Roberto Alfieri

Il livello Network: sommario

PARTE I

- ▶ Scopi del livello Network
- ▶ Commutazione di circuito e di pacchetto
- ▶ La famiglia dei protocolli TCP/IP
- ▶ Il protocollo IP: trama indirizzi, instradamento
- ▶ Protocolli di servizio: ARP, ICMP, DHCP

PARTE II

- ▶ IPv6

PARTE III

- ▶ Routing: Algoritmi e protocolli. Distance Vector e Link State.

IPv6

Necessità di un nuovo layer IP:

- ▶ Supportare molti miliardi di host
- ▶ Semplificare il routing per avere backbone veloci
- ▶ Offrire meccanismi di sicurezza
- ▶ Offrire qualità di servizio (multimedialità)
- ▶ Gestire bene multicast e broadcast
- ▶ Consentire la mobilità
- ▶ Consentire future evoluzioni e garantire compatibilità col passato

Nel 1993 tra varie proposte venne scelta SIPP (Simple Internet Protocol Plus) che prese il nome di **IPv6**.

Indirizzi IPv6

Spazio degli indirizzi grande a sufficienza (16 byte → 128 bit)

Notazione: 8 quaterne di numeri esadecimali separati da ":"

Esempio: 8000:0000:0000:0000:0562:CDAF:2DAF:0001

Notazione compatta: è possibile omettere gli zeri iniziali di ogni quaterna.

Gruppi di 4 zeri possono essere sostituiti con ::

Esempio precedente: 8000::562:CDAF:2DAF:1

Gli indirizzi IPv4 possono essere compresi tra gli indirizzi IPv6 con un prefisso di 96 zeri, mantenendo la notazione dotted decimal. Esempio: ::192.31.20.46

Indirizzi Broadcast: non esistono in IPv6

Indirizzi Speciali: Loopback (127.0.0.1 di IPv4) ::1

Indirizzi Multicast: Indirizzi assegnati a più interfacce (come IPv4)

Indirizzi Anycast (novità): Sono indirizzi assegnati a più interfacce.

Il pacchetto anycast viene consegnato solo all'interfaccia più vicina

Gli indirizzi IPv6 in URL devono essere scritti tra parentesi quadre. Esempio:

[http://\[2001:1:4F3A:206:AE14\]:8888/index.html](http://[2001:1:4F3A:206:AE14]:8888/index.html)

Prefix	Hex	Size	Allocation
0x0000 0000 0000 0000 0000 0000		2	Ipv4 compatible
0000 0000	0000-00FF		Reserved
0000 0001	0100-01FF		Unassigned
0000 001	0200-03FF	2	NSAP
0000 010	0400-05FF		Unassigned
0000 011	0600-07FF		Unassigned
0000 1	0800-0FFF		Unassigned
0001	1000-1FFF		Unassigned
001	2000-3FFF	2	IANA to registers
010,011,100,101,110	4000-CFFF		Unassigned
1110	D000-EFFF		Unassigned
1111 0	F000-F7FF		Unassigned
1111 10	F800-FBFF		Unassigned
1111 110	FC00-FDFF		Unassigned
1111 1110 0	FE00-FE7F		Unassigned
1111 1110 10	FE80-FEBF	2	Link-local
1111 1110 11	FEC0-FEFF	2	Site-Local
1111 1111	FF00-FFFF	2	Multicast

IPv6: Indirizzi Global Unicast

IANA 2000::/3

Gli indirizzi Unicast globali di IPv6 hanno prefisso 001 (2000::/3) e sono gestiti da IANA. IANA ha frammentato questo spazio in diverse reti più piccole che ha poi assegnato in gestione alle RIR continentali (APNIC, ARIN, RIPE, LACNIC e AFRINIC)

<http://www.iana.org/assignments/ipv6-unicast-address-assignments/ipv6-unicast-address-assignments.xhtml>

RIPE 2001:600::/23

La rete 2001:0600::/23 (2001:06xx: e 2001:07xx:) è stata assegnata a RIPE, che ha suddiviso in reti più piccole (tipicamente /32). <http://www.ripe.net/ripe/docs/ripe-510#2e>

GARR 2001:760::/32

La rete 2001:760::/32 è stata assegnata da RIPE a GARR, che ha suddiviso in reti più piccole (/48).

UNIPR 2001:760:2E04::/48 - INFN-Parma 2001:760:4207::/48

Il GARR ha assegnato la rete 2001:760:2E04::/48 a UNIPR e la rete 2001:760:4207::/48 a INFN-Parma. UNIPR dispone di 64K reti /64 da ripartire alle proprie strutture.

IPv6: possibile ripartizione della rete in UNIPR

Rete di ateneo: 2001:760:2e04::/48

2001:760:2e04:0000::/52	0000 --> 0fff	4k Reti /64 destinate ad usi futuri
1 2001:760:2e04:1000::/52	1000 --> 1fff	4k Reti /64 destinate alla Sede 1 (Campus)
2 2001:760:2e04:2000::/52	2000 --> 2fff	4k Reti /64 destinate alla Sede 2 (Sede Centrale)
3 2001:760:2e04:3000::/52	3000 --> 3fff	4k Reti /64 destinate alla Sede 3 (Via Gramsci)
4 2001:760:2e04:4000::/52	4000 --> 4fff	4k Reti /64 destinate alla Sede 4 (B.go Carissimi)
5 2001:760:2e04:5000::/52	5000 --> 5fff	4k Reti /64 destinate alla Sede 5 (Via Kennedy/via D'Azeglio)
6 2001:760:2e04:6000::/52	6000 --> 6fff	4k Reti /64 destinate alla Sede 6 (S. Francesco)
7 2001:760:2e04:7000::/52	7000 --> 7fff	4k Reti /64 destinate alla Sede 7 (Momentaneamente dismessa)
8 2001:760:2e04:8000::/52	8000 --> 8fff	4k Reti /64 destinate alla Sede 8 (Via del Taglio)
9 2001:760:2e04:9000::/52	9000 --> 9fff	4k Reti /64 destinate Alla Sede 9 (Via Volturro)

Le sedi in servizio denominate 10 11 12 e 13 sono sedi con un numero di host allocati sensibilmente inferiore a 100 è quindi plausibile supporre che non abbiano grosse esigenze di indirizzamento futuro pertanto si propone di continuare l'allocazione con il seguente schema

2001:760:2e04:a000::/52

10 2001:760:2e04:a100::/60	a100 --> a10f	16 Reti /64 destinate alla Sede 10 (Via S. Michele)
11 2001:760:2e04:a110::/60	a110 --> a11f	16 Reti /64 destinate alla Sede 11 (via Farini)
12 2001:760:2e04:a120::/60	a120 --> a12f	16 Reti /64 destinate alla Sede 12 (Pilotta)
13 2001:760:2e04:a130::/60	a130 --> a13f	16 Reti /64 destinate alla Sede 13 (Paradigna)
14 2001:760:2e04:a140::/60	a140 --> a14f	16 Reti /64 destinate alla Sede 10 (Beni Teatrali)
15 2001:760:2e04:a150::/60	a seguire per le altri sedi attive	
16 2001:760:2e04:a160::/60	a seguire per le altri sedi attive	
17 2001:760:2e04:a170::/60	a seguire per le altri sedi attive	
18 2001:760:2e04:a180::/60	a seguire per le altri sedi attive	
19 2001:760:2e04:a190::/60	a seguire per le altri sedi attive	

InterfaceID

Le reti assegnate alle strutture per le reti locali sono generalmente di 64 bit.

Gli ultimi 64 bit dell'indirizzo IPv6 possono essere assegnati in vari modi:

- ▶ Assegnati via DHCPv6
- ▶ Configurati manualmente
- ▶ Autogenerati con numeri pseudo-random
- ▶ Autoconfigurati utilizzando **l'interfaceID**, ovvero una sequenza di 64 bit, univoci di ogni interfaccia di rete, ottenuta partendo dai 48 bit del MAC address

Da MAC48 a InterfaceID

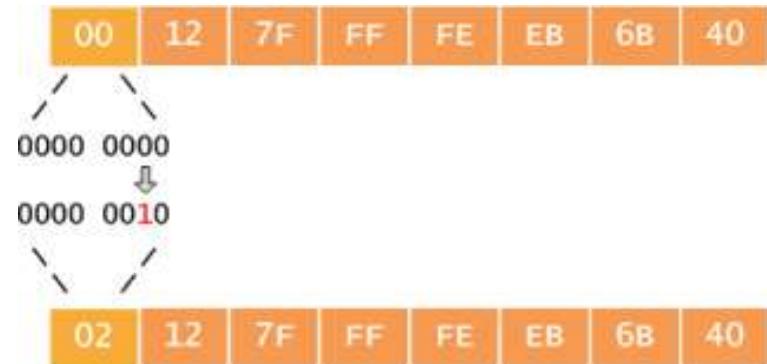
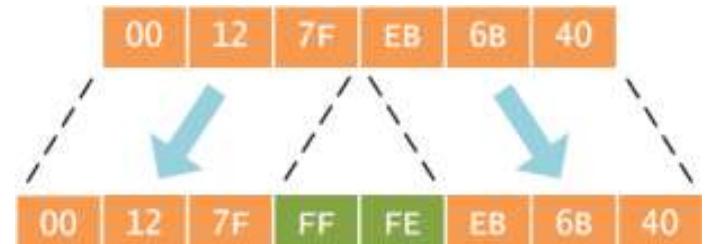
- ▶ Gli indirizzi MAC 48 bit utilizzati da Ethernet (MAC48) sono gestiti da IEEE e non si esauriranno prima del 2100.
- ▶ IEEE gestisce anche una numerazione a 64 bit, EUI64 (Extended Unique Identifier). La numerazione MAC48 è integrata in EUI64 inserendo 16 bit (FFFE) al centro.
- ▶ L' Interface-ID utilizzata per gli indirizzi IPv6 è una versione modificata di EUI64 (mEUI64) in cui si pone ad 1 il bit 7 di EUI64.

Esempio:

MAC48: 00-12-7F-EB-6B-40

EUI64: 00-12-7F-FF-FE-EB-6B-40

mEUI64: 02-12-7F-FF-FE-EB-6B-40 (intID)



IPv6: Indirizzi Link-Local

Per Link si intende una rete di livello 2 (LAN o punto-punto). Nodi sullo stesso link sono detti Neighbor (vicini)

Indirizzo Link locale: Destinati ai terminali della stessa rete locale.

Hanno come prefisso 1111 1110 10 (FE80::/10)

I pacchetti con questa destinazione non attraverseranno mai un router.

E' un tipo di indirizzo attribuito inizialmente alle interfacce IPv6 con configurazione automatica e viene utilizzato per il processo di Neighbor Discovery.

La configurazione automatica ha il seguente formato:

**FE80:0000:0000:0000:xxxx:xxxx:xxxx:xxxx
--interfaceID---**

IPv6: Indirizzi Site-Local

Un **Site** è un gruppo di Link gestiti da un'unica autorità (esempio Campus).

Gli indirizzi Site-local sono indirizzi per **uso privato**, analoghi alle reti 10.0.0.0/8, 172.16.0.0/12, e 192.168.0.0/16 di IPv4.

Hanno come prefisso 1111 1110 11 (FEC0::/10)

Rispetto a un indirizzo Link-local cambia il prefisso di formato, aggiungendo la possibilità e la convenienza di suddividere lo spazio di indirizzi in sottoreti.

A differenza dagli indirizzi Link-local non sono configurati automaticamente.

IPv6: Indirizzi Multicast e Anycast

Un indirizzo **IPv6 multicast** serve a identificare e a raggiungere un gruppo di nodi simultaneamente. Normalmente il multicast non viene propagato dai router a meno di configurazioni specifiche.

Il prefisso di formato è 1111 1111 (ovvero FF) a cui seguono 4 bit di opzione, 4 bit di ambito e 112 bit per identificare il gruppo.

Vedi http://wwwcdf.pd.infn.it/AppuntiLinux/introduzione_a_ipv6.htm

Gli **indirizzi anycast** sono degli indirizzi con le caratteristiche di quelli unicast che, in base al contesto, sono attribuiti a più interfacce di rete differenti, appartenenti ad altrettanti componenti di rete distinti.

Anycast viene gestito automaticamente dai router, i quali includono in tabella la destinazione che si raggiunge a minor costo.

Trova applicazione per servizi con alto tasso di utilizzo, che vengono replicati in punti diversi della rete con lo stesso indirizzo Anycast.

Esempi sono i root server dell'architettura DNS, oppure applicazioni in Cloud.

Altri Indirizzi IPv6

Loopback

0:0:0:0:0:0:1 (oppure ::1) identifica lo stesso nodo, come 127.0.0.1 in IPv4

Per controllare se lo stack IPv6 funziona: ping6 ::1

IPv4 compatible

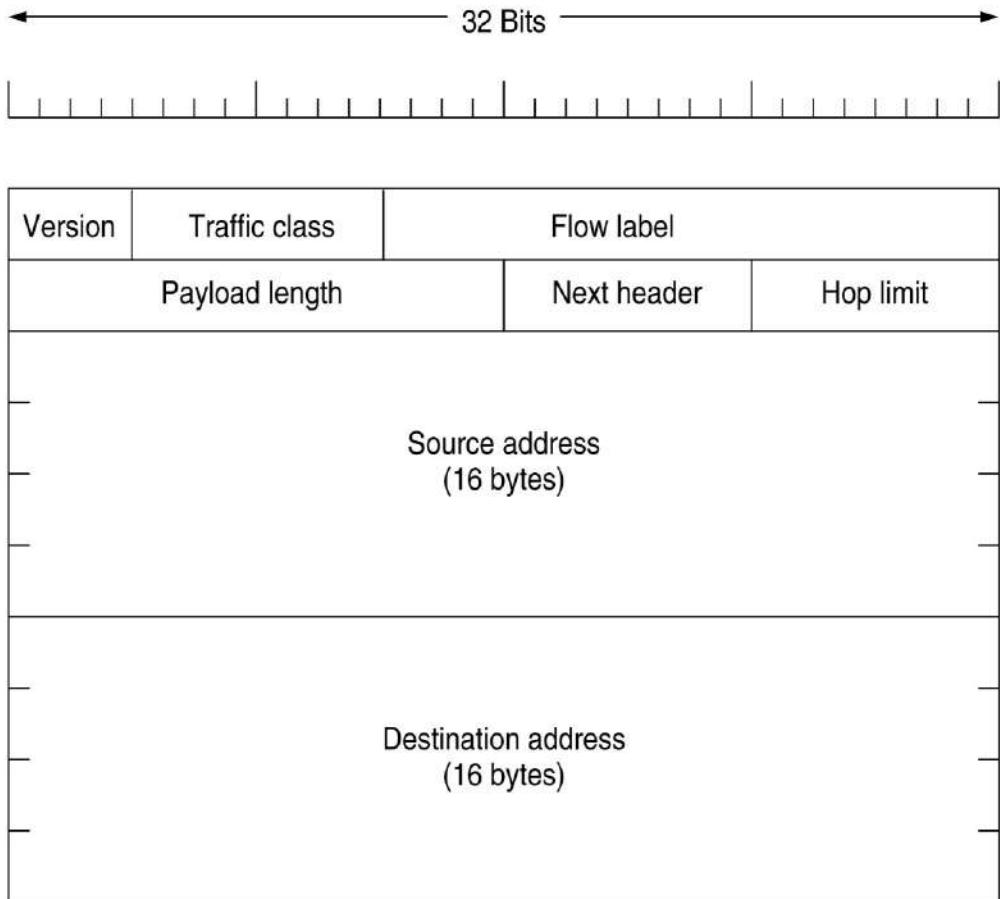
Permettono di inserire indirizzi IPv4 in indirizzi IPv6 anteponendo 96 zeri:

Esempio: 10.0.0.1 -> ::A001

vale anche la notazione ::10.0.0.1

Utilizzati per la transizione IPv4-IPv6

La trama IPv6

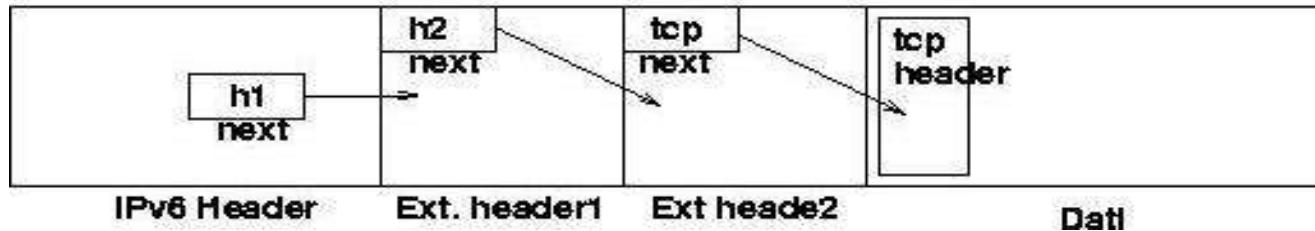


Cosa è stato eliminato da IPv4

- La frammentazione è stata rimossa perché IPv6 determina dinamicamente la dimensione del datagramma (Path MTU Discovery – [rfc 1191](#))
- Il campo Checksum è stato eliminato perché la sua elaborazione riduce le prestazioni.
- Il campo Protocol è stato rimosso perché questa info è contenuta nel Next Header.

Header Fields

- ▶ **Version** (4 bits) -> 0110
- ▶ **Traffic Class** (8 bits). E' un campo utilizzato per supportare la QoS basata sulle Classi. Corrisponde al Type of Service di Ipv4 utilizzato solo sperimentalmente)
- ▶ **Flow Label** (20 bits) – Label Switching, per QoS basata sui flussi (nuovo campo).
- ▶ **Payload Length** (16 bits) – Lunghezza del payload (esclusa l'intestazione)
- ▶ **Next Header** (8 bits) – Per snellire l'intestazione molti campi sono resi opzionali mediante Header numerate che possono essere concatenate



- ▶ Hop Limit (8 bits) – Era il TTL che ora assume il nome corretto.
- ▶ Source address (128 bits)
- ▶ Destination address (128 bits)

Extension Header

Estensioni opzionali nel formato Type-Lunghezza-Valore

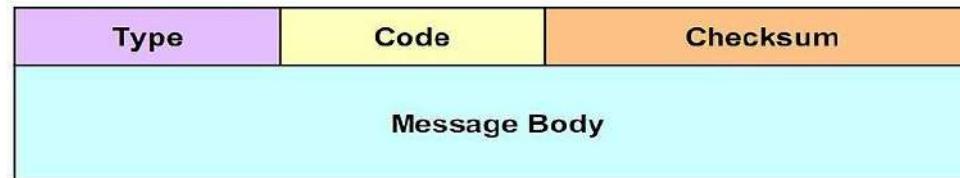
L'ultimo NextHeader indica il protocollo del Payload (stessi codici del campo Protocol di IPv4)

Code	Header Type
0	Hop-By-Hop options – Informazioni per i router attraversati
43	Routing Header – Lista di router da visitare nell'ordine indicato
44	Fragmentation Header – In alcuni casi la frammentazione è necessaria
50	Encapsulating Security Payload (ESP – IPsec) – Cifratura del datagramma
51	Authentication Header (AH – IPsec) – Integrita' del datagramma
60	Destination Options – Informazioni per il destinatario
1	ICMPv4
58	ICMPv6
6	TDP
17	UCP

ICMPv6

Equivale a ICMP per IPv4, con alcune nuove funzionalità:

- ▶ Path MTU discovery
- ▶ Neighbor discovery (equivalente in IPv6 di ARP)
- ▶ Router Discovery



Formato del pacchetto:

Type: indica il tipo - Code: specifica meglio il tipo - Checksum: dell'intero pacchetto

Tipi principali:

1=dest unreachable (no route to dest, address unreachable, ...)

2= packet too big (per il discovery automatico dell'MTU ottimale)

3=time exceeded (superato il numero massimo di hop consentiti)

128=echo req (ping)

129=echo replay (risposta al ping)

133=router solicitation (ricerca automatica dei router della LAN)

134=router advertisement

135=neighbor solicitation (sostituisce arp request)

136=neighbor advertisement (sostituisce arp response)

Path MTU discovery

E' un protocollo basato su ICMPv6 che consente di determinare l'MTU ottimale per connessioni TCP.

- Il nodo manda il primo pacchetto con una dimensione pari all'MTU del proprio link
- Se riceve un messaggio ICMPv6 "Packet too big" (tipo 2) manda un nuovo pacchetto con le dimensioni indicate nel messaggio
- Ripete finché non trova più errori

Neighbor discovery

Sostituisce ARP per determinare l'indirizzo di rete LAN.

- ▶ Usa pacchetti ICMPv6 anziché ARP, multicast anziché broadcast
- ▶ Per ottenere un indirizzo fisico di un altro nodo:
 - Calcola l'indirizzo Solicited-Node (multicast) corrispondente all'indirizzo IPv6 del destinatario, formato aggiungendo gli ultimi 24 bit dell'indirizzo IP (ultime 6 cifre esadecimali del dest) al prefisso ff02::1:ff00:/104
 - Invia all'indirizzo multicast un pacchetto ICMPv6 “Neighbor Solicitation” (125)
 - Il destinatario risponde con un pacchetto ICMPv6 “Neighbor Advertisement” (136)
 - Il nodo memorizza l'indirizzo della Neighbor Cache

Router discovery

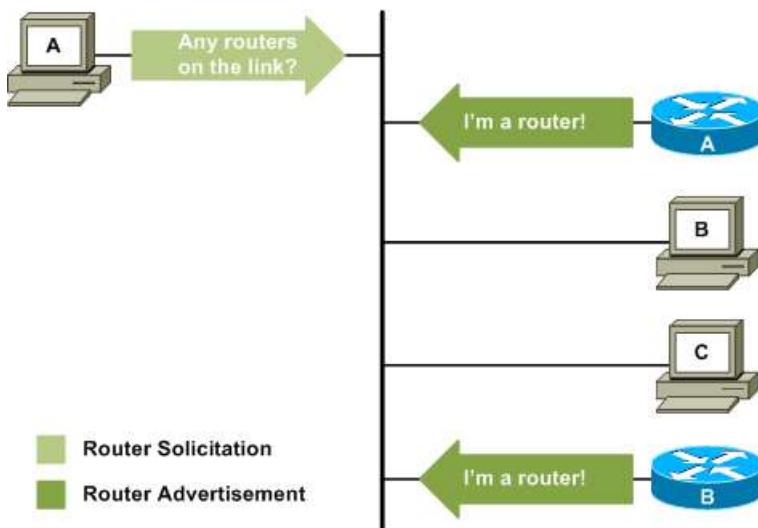
In IPv4 il default router deve essere configurato manualmente o via DHCP.

Con IPv6 gli host possono individuare automaticamente i router in un link.

Questo avviene attraverso 2 messaggi ICMPv6:

Router Solicitation (RS, type 133) e **Router Advertisement** (RA, type 124)

Quando un host entra in Link manda un Router Solicitation in multicast all'indirizzo [FF02::2] e ogni router risponde con un Router Advertisement contenente il suo indirizzo e altre informazioni necessarie per il routing.



DHCPv6 (statefull autoconfiguration)

E' il protocollo dhcp per IPv6 descritto nell'RFC 3315 e consiste nello scambio dei seguenti segmenti UDP:

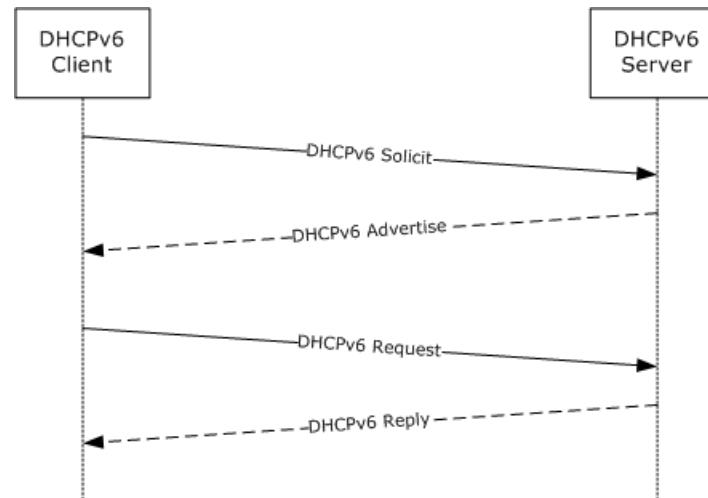
Il client manda un **"Solicit"** dalla porta 546 a [ff02::1:2]:547 (multicast)

Il server risponde con un **"Advertise"** unicast dalla porta 547 verso la porta 546.

Il client risponde con un **"Request"** dalla porta 546 a [ff02::1:2]:547 (multicast)

Il server completa il protocollo con un **"Reply"** unicast dalla porta 547 verso la 546.

Nota: Per identificare gli host DHCP6 usa il DUID (DHCP UID) che è unico per ogni Host.



Stateless Address AutoConfiguration (SLAAC)

SLAAC è definito nell'RFC 2462

Combinando il protocollo di router discovery con l'autoconfigurazione degli indirizzi Link-local (FE80:0000:0000:0000:mEUI64) è possibile assegnare un indirizzo Global unicast in modalità plug & play, senza la necessità di avere un servizio DHCP.

Al momento del boot l'host ottiene dalla rete il default router ed il prefisso IPv6, quindi genera il Global address combinando LinkPrefix:mEUI64

- Adatto per i client (i server devono essere configurati manualmente)
- Il nome del DNS deve essere ottenuto in altro modo (esempio DHCPCv6)
- L'indirizzo non viene automaticamente registrato nel DNS.

Nota: Nei sistemi Linux l'attivazione di SLAAC è controllata dall'opzione IPV6_AUTOCONF

Esempio: IPV6_AUTOCONF=YES



UNIVERSITÀ
DI PARMA

DIPARTIMENTO DI SCIENZE MATEMATICHE, FISICHE ED INFORMATICHE
Corso di Laurea in Informatica

II Livello Network

Parte III : Routing

RETI DI CALCOLATORI - a.a. 2023/2024

Roberto Alfieri

Il livello Network: sommario

PARTE I

- ▶ Scopi del livello Network
- ▶ Comutazione di circuito e di pacchetto
- ▶ La rete ATM
- ▶ La famiglia dei protocolli TCP/IP
- ▶ Il protocollo IP: trama indirizzi, instradamento
- ▶ Protocolli di servizio: ARP, ICMP, DHCP

PARTE II

- ▶ IPv6

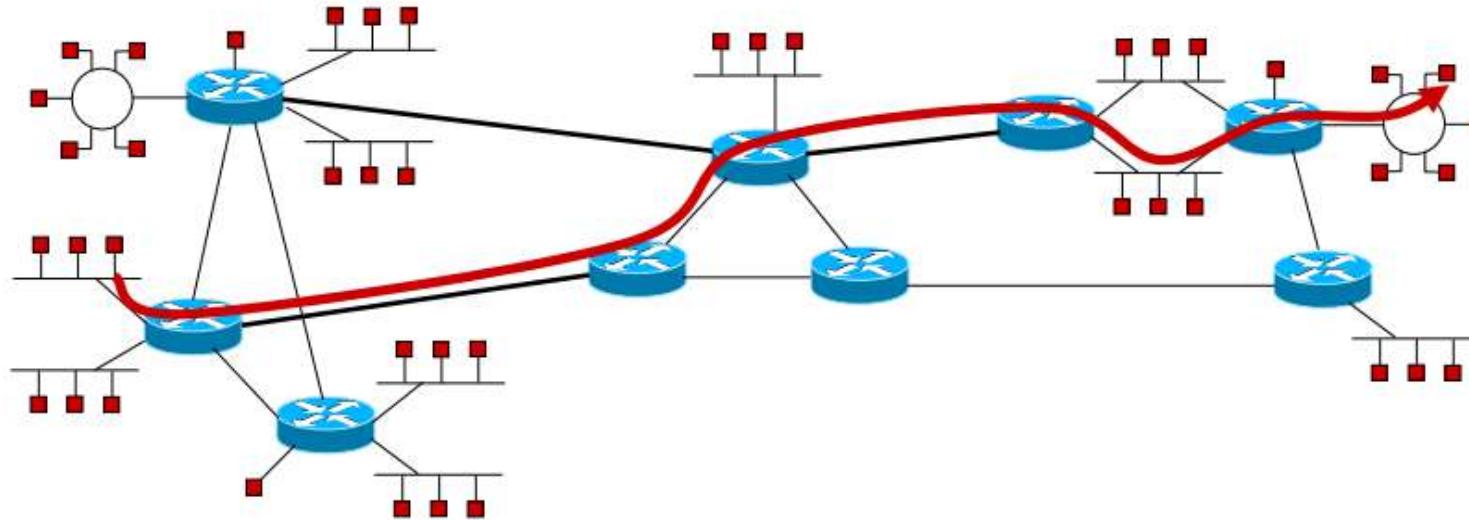
PARTE III

- ▶ Routing: Algoritmi e protocolli. Distance Vector e Link State.

Routing

E' la scelta del percorso su cui inviare i dati quando mittente e destinatario appartengono a 2 reti diverse e quindi la consegna non può avvenire direttamente a livello Link.

In questo caso il mittente affida la consegna ad una struttura interconnessa di **Router** i quali passano i datagrammi dall'uno all'altro finché raggiungono quello che può consegnarli direttamente al destinatario.

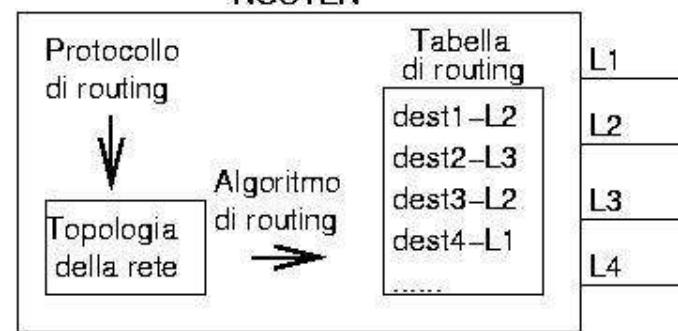


Router

I Router sono dotati di due componenti funzionali: **instradamento e inoltro**.

Instrandamento: Creazione di **una tabella di routing (RT)**, che contiene le informazioni riguardo la porta di uscita per le destinazioni dei Frame. Componenti:

- ▶ **Algoritmi di routing (RA)** si preoccupano di scegliere lungo quale linea di uscita vanno instradati i pacchetti in arrivo. Sono usati per il calcolo della tabella di routing in base alla topologia della rete.
- ▶ **Protocolli di routing (RP)** utilizzati per lo scambio delle informazioni necessarie per determinare la topologie della rete.



Inoltro: Applicazione dell'instrandamento sui singoli datagrammi ricevuti.

- ▶ **Lettura dell'intestazione IP** ed estrazione dell'indirizzo di destinazione.
- ▶ **Look-up della tabella di routing** ed identificazione dell'interfaccia di uscita
- ▶ **Switching:** trasferimento fisico dei datagrammi da ingresso a uscita.

Algoritmo di Flooding

Flooding è un semplice algoritmo di routing in cui ogni pacchetto entrante è spedito verso tutte le linee di uscita eccetto quella di cui è arrivato.

Per gestire la propagazione di pacchetti duplicati le possibili tecniche sono:

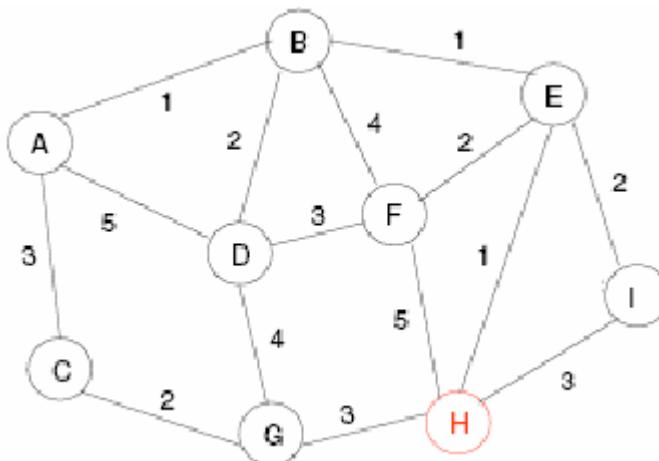
- Conteggio del numero di salti, decrementando il valore ad ogni salto.
- Numero di sequenza assegnato dal mittente ad ogni messaggio. Ogni nodo deve gestire una lista con i messaggi già trasmessi, scartando eventuali repliche.

Flooding è utilizzato

- nei bridge per l'invio di broadcast e nella fase di autoapprendimento
- nei protocolli link state

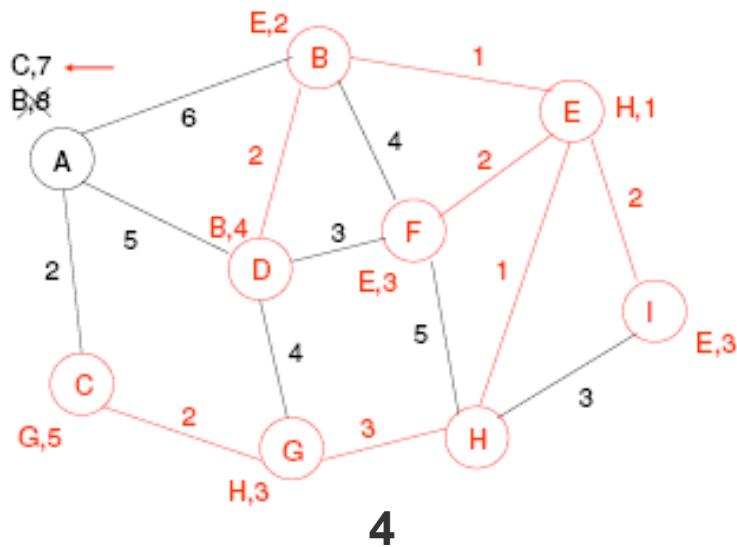
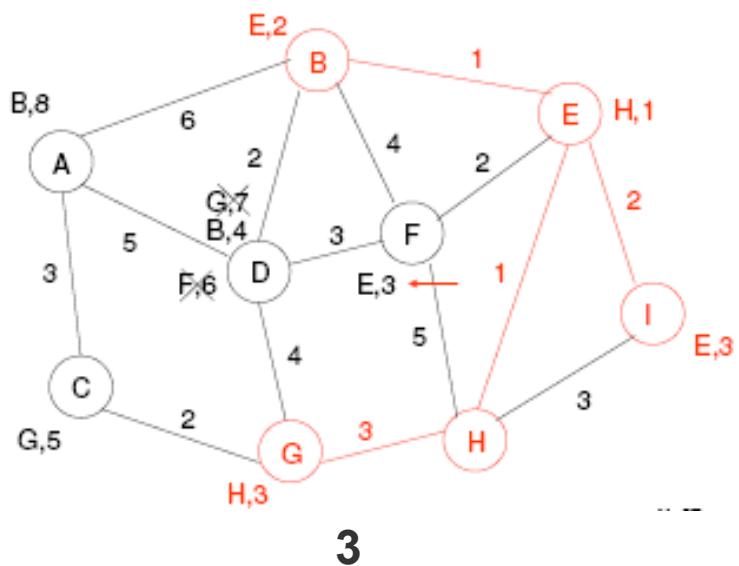
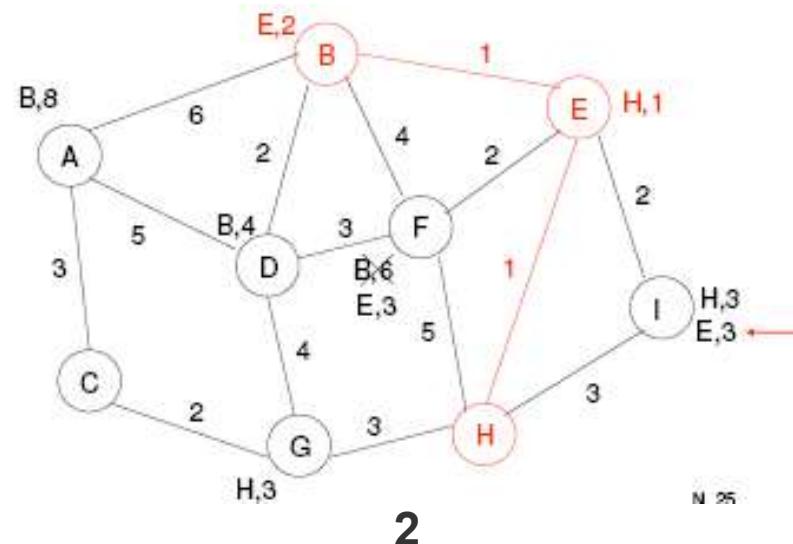
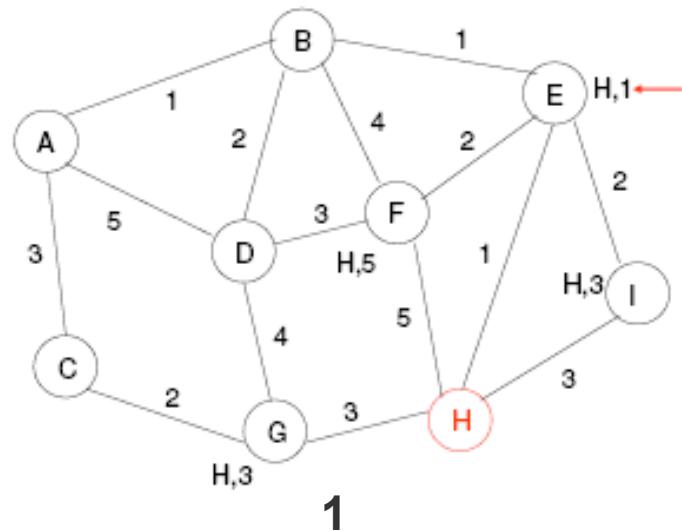
Algoritmo Shortest Path First

- ▶ La topologia della rete può essere rappresentata da un grafo pesato non orientato. Il peso attribuito ad ogni arco viene determinato mediante l'attribuzione di una metrica che tiene conto vari parametri di rete (velocità, latenza, ..)
- ▶ Per il principio di ottimalità il cammino minimo che un nodo deve percorrere per raggiungere qualsiasi altro nodo del grafo è un albero detto “**Sink Tree**” (a partire dalla destinazione) o **Source Tree** (a partire dall'origine).
- ▶ Si conoscono diversi algoritmi per elaborare il percorso più breve tra due nodi. Il più utilizzato è stato ideato da Dijkstra nel 1959 ed è noto con il nome di **Shortest Path First (SPF)**.



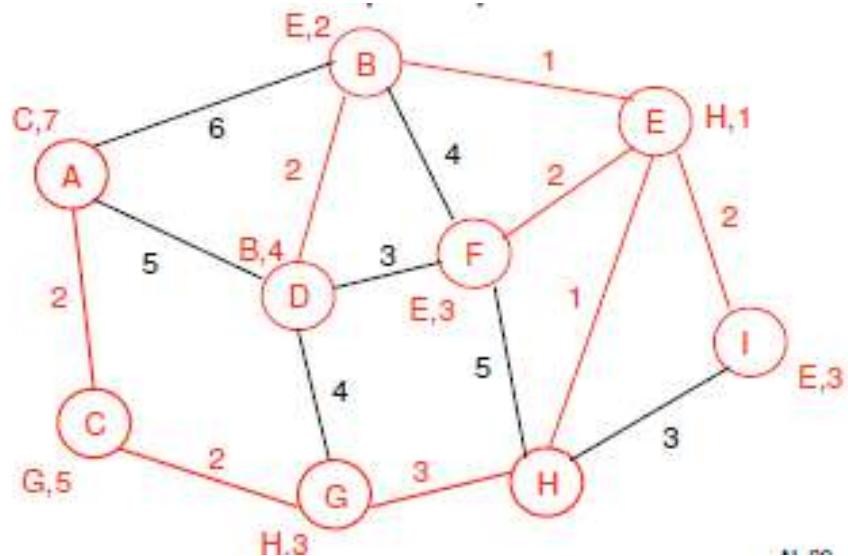
Ricerca del cammino minimo da A verso H. A partire da H (Sink Tree) mettiamo etichette provvisorie sui nodi adiacenti. Scegliamo la più piccola delle distanze.

Sink Tree del nodo H con SPF



Risultato finale

- ▶ A va ad H attraverso C con distanza 7.



Protocolli di Routing (RP)

I RP stabiliscono le modalità di comunicazione tra i router per la costruzione della topologia di rete. Possono essere statici o dinamici (adattivi).

► Routing Statico:

- La topologia e tabella di routing vengono definite in fase di setup della rete.
- In caso di variazioni (inserimento o eliminazione di nodi o collegamenti) è necessario l'intervento dell'operatore.

► Routing Dinamico:

- La topologia della rete è costruita dinamicamente in modo automatico, in base ai cambiamenti della topologia di rete o al traffico.

■ **Routing Dinamico Centralizzato:**

- - un nodo centrale raccoglie le informazioni sullo stato della rete
- - calcola (RA) la tabella per ogni nodo e la spedisce.
- - tabelle consistenti, ma abbiamo un punto di criticità

■ **Routing Dinamico Distribuito:**

- - i nodi si scambiano informazioni sullo stato della rete
- - ogni nodo calcola la propria tabella sulla base delle informazioni ricevute.
- - tre categorie di protocolli: **Distance Vector, Link State e Gerarchici**

Protocolli Distance Vector

- ▶ Nel grafo ogni coppia di nodi ha una **distanza** che dipende dalla “metrica” utilizzata.
- ▶ Una metrica ragionevole tra 2 nodi adiacenti potrebbe dipendere dalla velocità, la latenza e il Throughput del canale.
- ▶ La distanza di un percorso potrebbe dipendere dalla somma delle singole distanze e/o dal numero di salti.
- ▶ **Nel Protocollo Distance Vector (DV) ogni nodo invia ai primi vicini l'elenco delle distanze (a lui note) con tutti gli altri nodi (ovvero il DV), periodicamente e ogni volta che c'è un cambiamento.**
- ▶ Le distanze con i primi vicini vengono misurate (ad esempio con un ECHO), mentre le altre distanze sono derivate dalle informazioni ricevute
- ▶ Tutte le volte che un Router calcola una nuova tabella di instradamento, la invia agli IS adiacenti (cioè quelli collegati da un cammino fisico diretto) sotto forma di DV
- ▶ **La tabella** contiene una entry per ogni nodo presente in rete
- ▶ Ogni **entry** è composta da quattro parametri:
 - Indirizzo (del nodo remoto)
 - Hops (numero di salti per raggiungerlo)
 - Costo (determinato in base alla metrica)
 - Linea
- ▶ Il DV inviato contiene Indirizzo-Hops-Costo di ogni entry (non la linea).

Protocolli Distance Vector

Il router che riceve il DV prima di tutto verifica se vi sono delle modifiche dal precedente e, in caso affermativo, aggiorna i campi hops e costo, sommando 1 a tutti gli hops e sommando il costo della linea da cui è arrivato il messaggio al campo costo.

Il passo successivo è l'aggiornamento della propria tabella tramite un processo di **fusione** (merge) di tutti i Distance Vector a lui pervenuti da ogni linea attiva.

Nella fusione vengono esaminate le entry con lo stesso indirizzo di destinazione, scartando quelle con i costi maggiori.

A parità di costo si seleziona quella che ha il **minor numero di hops**.

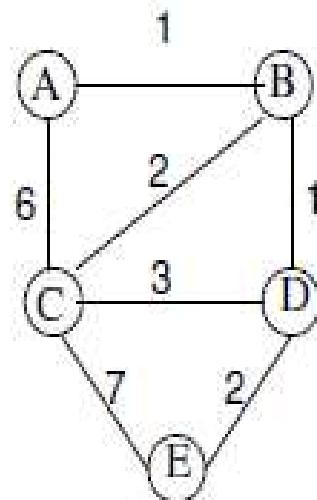
Il protocollo è semplice ma a lenta convergenza: l'informazione di una modifica della topologia (linea interrotta, router spento, ..) si propaga lentamente.

Indirizzo	Hops	Costo	Linea
1	3	25	3
2	5	35	2
3	9	50	6
4	1	5	7
5	0	0	0

DV



Distance Vector: Esempio di calcolo



Distance Vector A = {(A,0), (B,1), (C,6), (D, ∞), (E, ∞)}

DV B = {(A,1), (B,0), (C,2), (D,1), (E, ∞)}

DV C = {(A,6), (B,2), (C,0), (D,3), (E,7)}

DV D = {(A, ∞), (B,1), (C,3), (D,0), (E,2)}

DV E = {(A, ∞), (B, ∞), (C,7), (D,2), (E,0)}

Tabella di routing alla fine del periodo di convergenza (al termine dello scambio dei DV)

1. A riceve DV di B

dest	Costo, next hop
A	0
B	1, A
C	3, B
D	2, B
E	∞

2. A riceve DV di C

dest	Costo, next hop
A	0
B	1, A
C	3, B
D	2, B
E	13, C

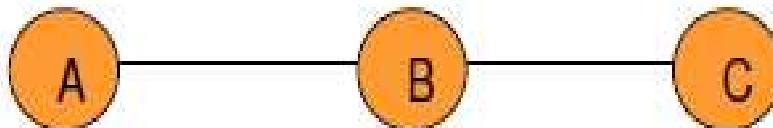
3. B riceve DV di D

dest	Costo, next hop
A	1, B
B	0
C	2, B
D	1, B
E	3, D

4. A riceve DV di B

dest	Costo, next hop
A	0
B	1, A
C	3, B
D	2, B
E	4, B

Distance Vector: il problema “count to infinity”



- **Situazione iniziale:** $D_{AC} = 2$ e $D_{BC} = 1$
 - Link BC va fuori servizio
 - B riceve il DV di A che contiene l'informazione $D_{AC} = 2$, per cui esso computa una nuova $D_{BC} = D_{BA} + D_{AC} = 3$ e la comunica ad A
 - A calcola la nuova distanza $D_{AC} = D_{AB} + D_{BC} = 4$
 - Il processo può continuare all'infinito
- **Vari rimedi sono stati proposti, nessuno risolutivo**

RIP protocol

RIP (Routing Information Protocol) è la prima implementazione di un protocollo DV.

Ne esistono 3 versioni:

- ▶ RIPv1 (RFC 1058) usa il routing "classful" (reti senza NetMask)
- ▶ RIPng (RFC 2080) estensione del protocollo RIPv1 per supportare IPv6.
- ▶ RIPv2 (RFC 2453) usa il routing “classless” (CIDR)

Caratteristiche:

- ▶ Metrica: basata solo sulla minimizzazione degli hops (max 15)
- ▶ Nodi RIP Attivi (tipicamente Router): annunciano il loro percorsi
 - **ogni 30 secondi e quando si verificano cambiamenti di topologia**
- ▶ Nodi RIP Passivi (tipicamente Host): aggiornano senza annunciare

Protocolli Link State

E' un RP con cui ogni nodo determina e mantiene aggiornata la topologia della rete da cui calcola la Tabella di Routing applicando un RA.

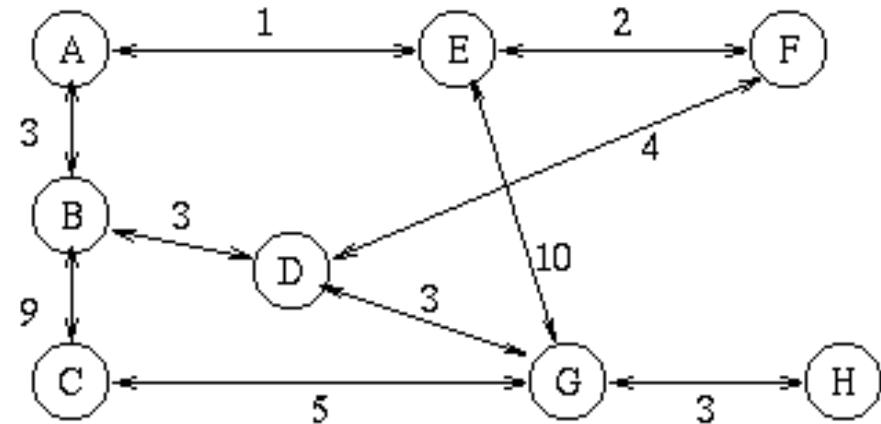
Il protocollo si sviluppa nelle seguenti fasi:

- 1. Scoperta dei vicini** (neighbor greetings) : invio di un pacchetto HELLO su tutte le linee.
- 2. Misurazione costo linea**: invio di un ECHO ai router che hanno risposto all'HELLO.
- 3. Costruzione di un pacchetto** (Link State Packet - LSP) con tutte le informazioni ricavate nella fase 2: l'identità del trasmittente, numero di sequenza, dall'età e lista di vicini con il relativo ritardo misurato.
- 4. Distribuzione periodica del LSP a TUTTI i nodi della rete di router**, con un numero di sequenza, utilizzando il Flooding. Se arriva un pacchetto con un numero di sequenza inferiore al numero più alto visto fino a quel momento, il pacchetto viene scartato (ritenuto obsoleto).
- 5. Ogni nodo**, dopo aver ricevuto gli LSP da tutti gli altri, **costruisce la topologia della rete** e applica un RA (Shortest Path First di Dijkstra) per il **calcolo della tabella**.

Protocolli Link State: esempio 1/2

- ▶ Raccolta Info (HELLO - ECHO)
- ▶ Propagazione info (LSP) in flooding multicast
- ▶ Per il nodo D il pacchetto LSP sarebbe:

Adiacente	Costo
B	3
F	4
G	3



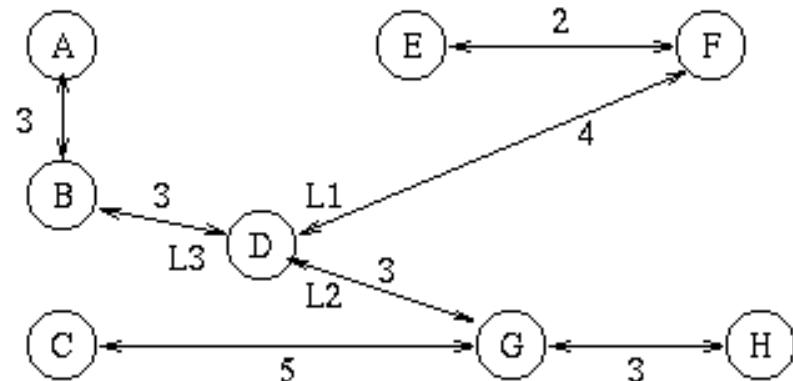
- ▶ Ogni nodo ricostruisce la mappa della rete fondendo i LSP ricevuti in una tabella come quella a fianco.

A	B/3	E/1	
B	A/3	C/9	D/3
C	B/9	G/5	
D	B/3	F/4	G/3
E	A/1	F/2	G/10
F	D/4	E/2	
G	C/5	D/3	E/10
H	G/3		

H/3

Protocolli Link State: esempio 2/2

- ▶ Il calcolo della tabella di instradamento si riduce ora al calcolo dello spanning tree di tipo SPF (Shortest Path First) e lo si effettua tramite il noto algoritmo di Dijkstra
- ▶ Lo spanning tree ad esempio del nodo D risulterà come nella figura a lato e, a seguire la relativa tabella di instradamento
- ▶ L'algoritmo può gestire reti di grandi dimensioni grazie alla sua rapida convergenza ed il suo comportamento è prevedibile, poiché ogni nodo ha in memoria la mappa intera della rete.
- ▶ Difficilmente si generano loop e, comunque, risulta facile identificarli ed eliminarli.



A	L3
B	L3
C	L2
E	L1
F	L1
G	L2
H	L2

OSPF

OSPF (Open Shortest Path First, RFC2328)

È un protocollo IGP di tipo **Link State Packet** ed è raccomandato da IETF per Internet.

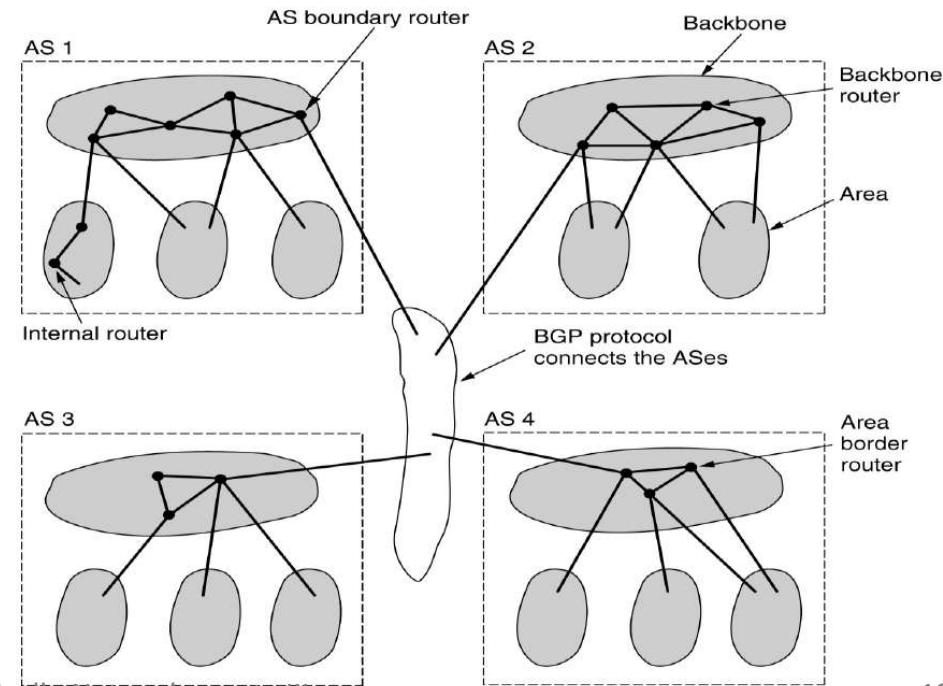
- ▶ Ciascun router emette periodicamente (default 10 s) dei **pacchetti Hello** multicast (244.0.0.5) , per valutare possibili modifiche topologiche
- ▶ Ogni Router costruisce un pacchetto con l'elenco delle linee attive e dei loro costi (1/larghezza di banda della linea)
- ▶ Invia in **flooding** pacchetti **Link State Update (LS-Update)** multicast 244.0.0.5
- ▶ Questi vengono riscontrati con un **Link State Ack (LS-Ack)**
- ▶ Se, in base ai pacchetti di update ricevuti, si sono verificate modifiche della topologia ricalcola la tabella di routing con l'algoritmo **Shortest Path First (SPF)**
- ▶ Scalabilità:
 - Il calcolo del SPF ha complessità $O(N \log N)$, dove N è il numero di router e reti.
 - Il problema viene risolto suddividendo un AS in aree:
 - Ogni AS-OSPF contiene almeno un'area: l'area di Backbone (area 0)
 - Le eventuali altre aree sono connesse al Backbone
 - Ogni router mantiene informazioni solo riguardo la topologia della propria area.

Protocolli Gerarchici

Nel caso di reti di grandi dimensioni non è possibile gestire le tabelle di routing per l'intera rete in tutti i router, in questo caso il routing deve essere gerarchico:

- ▶ la rete viene ripartita in aree, chiamate Autonomous-System
- ▶ i router all'interno di un area sono in grado di effettuare l'instradamento relativamente alla sola area
- ▶ per destinazioni al di fuori dell'area si limitano ad inviare i pacchetti a dei router "di bordo" che sono a conoscenza della topologia esterna dell'area
- ▶ i router "di bordo" si occupano solamente dell'instradamento dei pacchetti fra aree

In linea di principio la ripartizione può essere effettuata tante volte quante si vuole creando più livelli nella gerarchia di routing



Protocolli di Routing in Internet

In TCP/IP i router sono suddivisi in due classi, **Exterior Router** ed **Interior Router**. I primi interconnettono due insiemi di reti distinti. Ogni insieme di reti, gestito da una singola autorità amministrativa, è un Autonomous System ed i router interni ad essi sono proprio gli Interior.

Un sistema è detto autonomo, poiché è libero di scegliere un'architettura di instradamento interna, ma deve raccogliere informazioni su tutte le sue reti e progettare uno o più gateway, gli Exterior Router, che passino le informazioni di raggiungibilità ad altri sistemi autonomi.

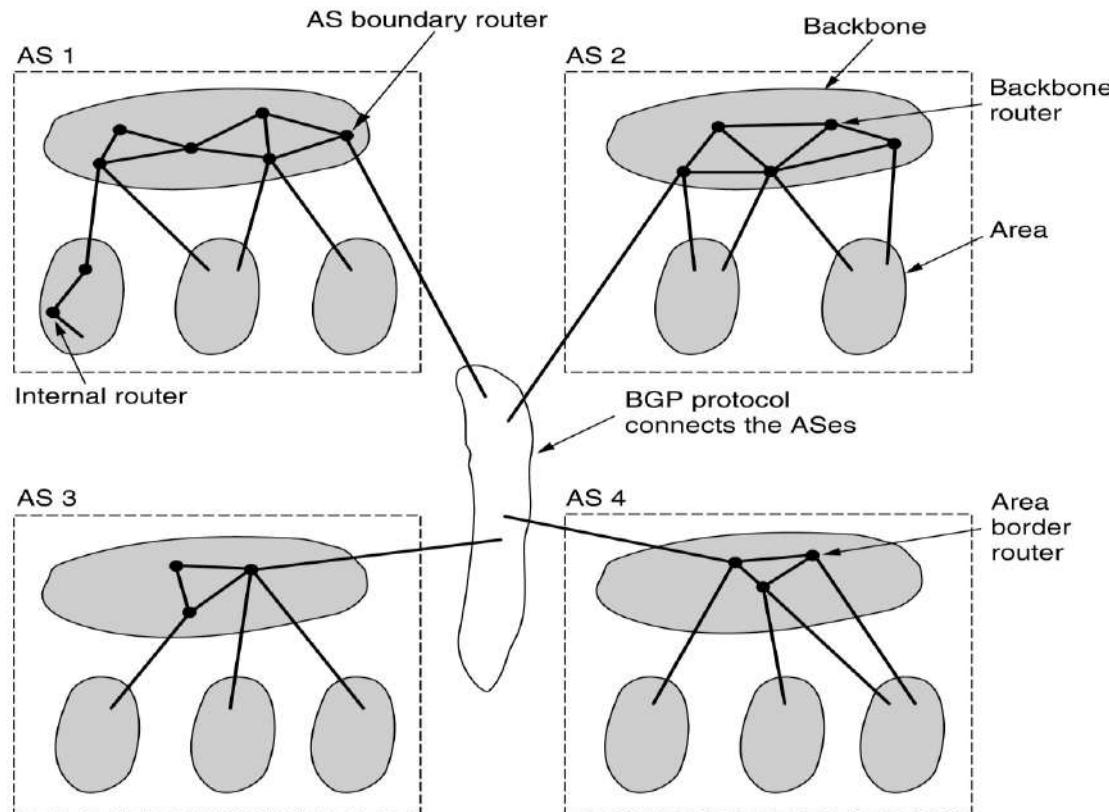
Gli Exterior Router utilizzano protocolli denominati EGP (Exterior Gateway Protocol), mentre gli Interior Router scambiano informazioni di instradamento tramite gli IGP (Interior Gateway Protocol).

I protocolli IGP più utilizzati sono **RIP** (Distance Vector) e **OSPF** (Link State)
BGP è il protocollo raccomandato in Internet per l'interconnessione di Autonomous System.

BGP

BGP (Border Gateway Protocol, RFC 1771).

Protocollo Path Vector : invece di propagare i costi propaga la sequenza di AS da attraversare per arrivare a destinazione

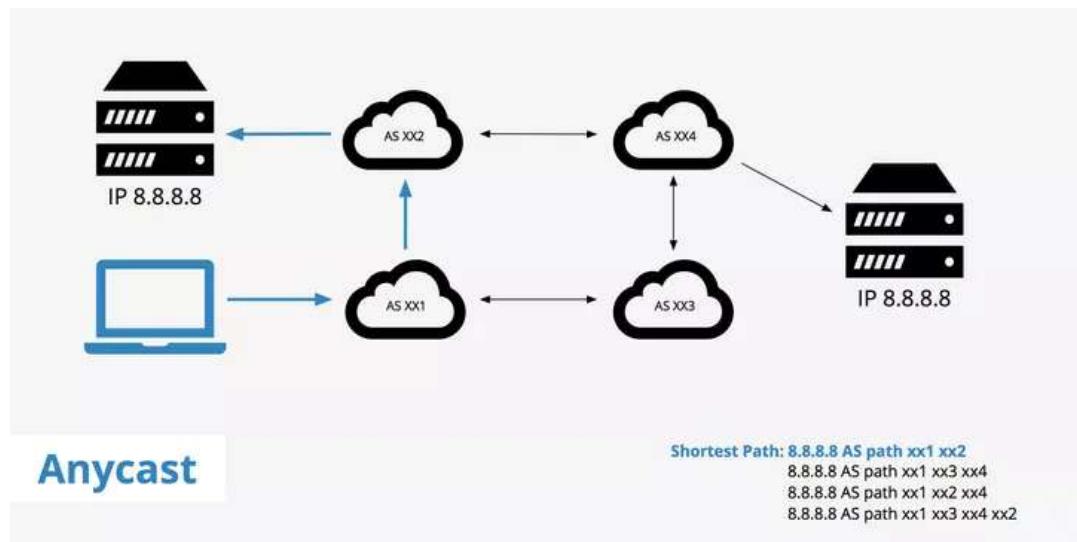


Routing Anycast

IP Anycast è una tecnica che consente a diverse macchine (server) di condividere lo stesso indirizzo IP, in modo che un client raggiunga il server più vicino (a minore costo) per ridurre la latenza e aumentare la ridondanza.

Gli algoritmi di routing basati sui protocolli Distance Vector o Link State gestiscono automaticamente percorsi multipli per raggiungere una destinazione, selezionando il percorso a minor costo e quindi la destinazione più conveniente.

Ovviamente il routing non è stabilito sugli indirizzi IP, ma sulle reti, quindi sarà necessario definire una rete Anycast, (anche piccola) replicata in diversi punti della rete.





UNIVERSITÀ
DI PARMA

DIPARTIMENTO DI SCIENZE MATEMATICHE, FISICHE ED INFORMATICHE
Corso di Laurea in Informatica

Il Livello Trasporto

RETI DI CALCOLATORI - a.a. 2023/2024

Roberto Alfieri

Il Livello Trasporto: sommario

PARTE I

- ▶ Scopo del livello Trasporto
- ▶ L'indirizzamento
- ▶ Il modello client/server
- ▶ Il protocollo UDP
- ▶ I servizi orientati alla connessione
- ▶ Il protocollo TCP

PARTE II

- ▶ Congestione, Qualità del Servizio
- ▶ Algoritmi Slow start, Tahoe, Fast Recovery
- ▶ QoS e controllo del traffico
- ▶ Servizi differenziati e integrati

RIFERIMENTI

- ▶ *Reti di Calcolatori, A. Tanenbaum, ed. Pearson*
- ▶ *Reti di calcolatori e Internet, Forouzan , Ed. McGraw-Hill*

Scopo del livello di Trasporto

- ▶ Fornire al **livello applicazione** paradigmi astratti per la comunicazione tra 2 processi: flusso di byte, scambio di messaggi, chiamata a funzione, ecc.
- ▶ Offre al livello applicativo una **interfaccia indipendente** dalle diverse tecnologie dello strato di rete (es IPv4, IPv6).
- ▶ Per assolvere le sue funzioni utilizza i **servizi** dello **strato di rete**
- ▶ **Presupposti della rete sottostante:**
 - I pacchetti possono andare perduti
 - arrivare in ordine modificato
 - consegnati in più copie
 - con ritardi indefiniti

Servizi del livello di Trasporto

I servizi di trasporto di Internet si basano sulla Socket Library introdotta dall'Università di Berkeley nel 1982

Fornisce 2 tipi di servizi di Trasporto:

- **Servizio affidabile orientato alla connessione: stream sockets (TCP)**

- Garanzia di integrità, completezza e ordine

- Gli utenti TCP vedono la connessione come una Pipe

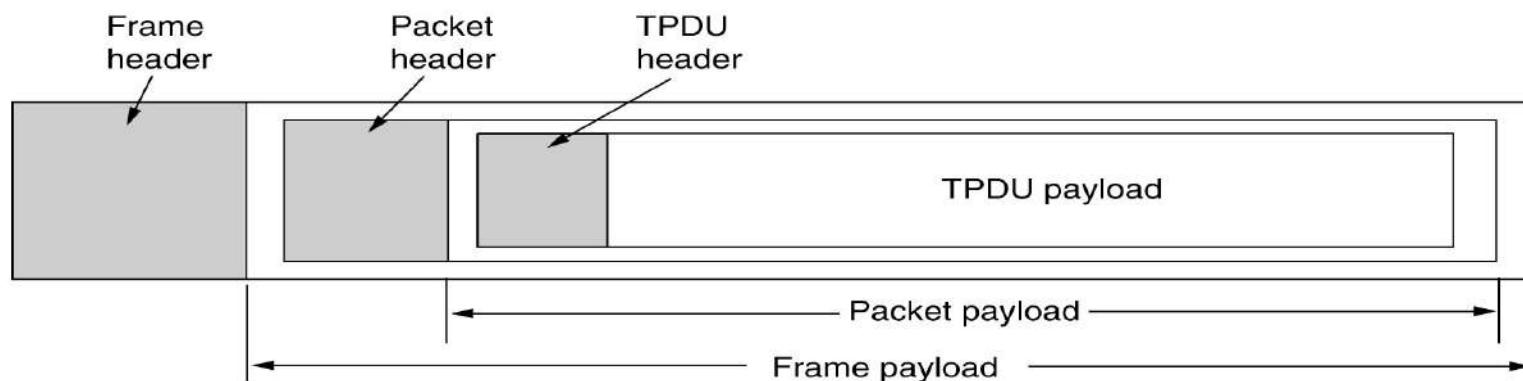
- **Scambio (inaffidabile) di Datagrammi: datagram sockets (UDP)**

- Ogni datagramma viene inviato senza garanzia di consegna

- Eventuali ordinamenti nell'invio di una successione di datagrammi devono essere gestiti dall'applicazione.

Processi di ricezione e di trasmissione

- 1) Nel processo di **ricezione** dei pacchetti dal livello rete (**Demultiplexing**) il livello di trasporto gestisce un indirizzamento (porta) che serve per associare il pacchetto IP in arrivo al processo applicativo a cui è destinato: analizza la porta di destinazione indicata nel pacchetto e smista il pacchetto processo applicativo corretto.
- 2) Nel processo di **spedizione** (**Multiplexing**) il dato viene eventualmente ridotto in **segmenti** (detti anche TPDU – Transport Protocol Data Unit) che vengono imbustati nell'header di trasporto con l'indicazione della porta di destinazione.



Demultiplexing: Le porte di Berkeley Socket Library

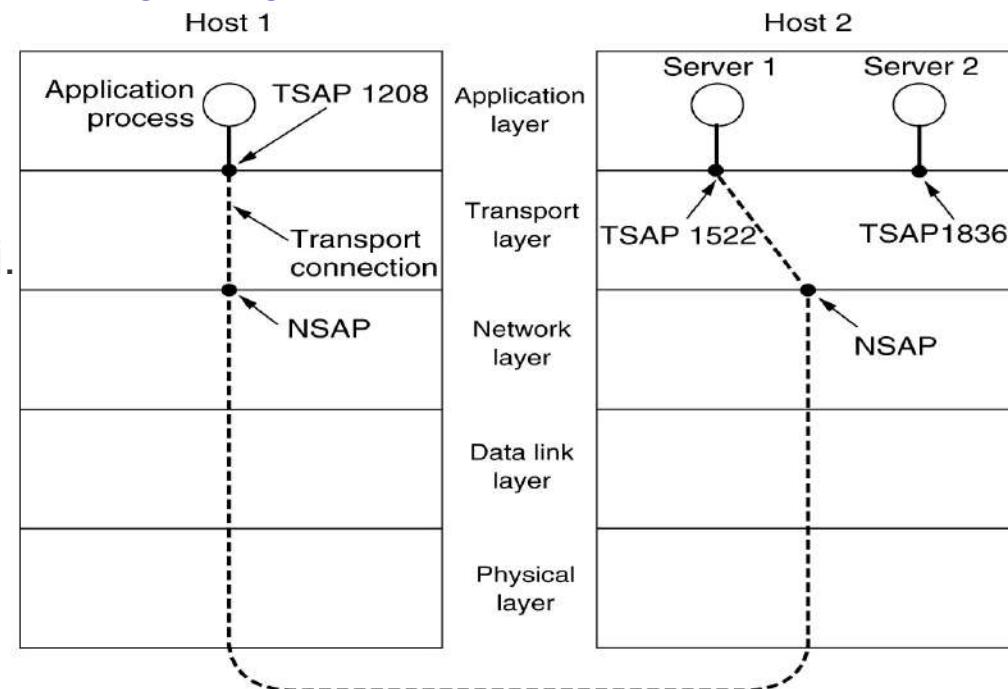
La porta è un identificativo numerico (16 bit , 64K porte) che rappresenta il punto di arrivo di una connessione su di un host. La coppia (IPaddr, Port) identifica quindi univocamente un estremo di una connessione ed è detta **Socket**.

Una connessione tra gli host A e B è identificata da una coppia di socket

IPaddrA,PortA – IPaddrB, portB

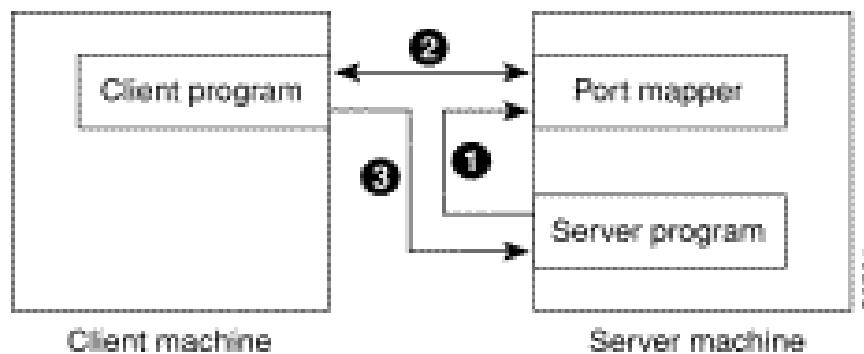
Le porte inferiori a 1024 sono dette **“Well-Known-Port”** e vengono universalmente associate alle principali applicazioni server da **IANA**, per agevolare l'identificazione del Socket server. Vedi <http://www.iana.org/assignments/port-numbers>

In diversi sistemi operativi le Well-Known-Ports possono essere assegnate solo da processi con privilegi. I processi server creati da utenti senza privilegi possono usare le porte non privilegiate (da 1025 a 32768). Le porte da 32768 a 61000 sono dette **effimere**, assegnate dinamicamente ai processi client.



Come trovo la porta?

- ▶ Se il **servizio è standard** il server utilizza una “Well known port”, che tutti conoscono, o una porta non privilegiata.
- ▶ Per **servizi di rete dinamici** si può utilizzare un Name Server (Directory Server) con un servizio di PortMapper in ascolto su una Well Known Port, su cui i servizi di rete registrano la porta di ascolto (1) . Il client interroga il Portmapper per conoscere la porta del Server (2), quindi contatta il Server (3). Questo meccanismo è utilizzato dal protocollo RPC.



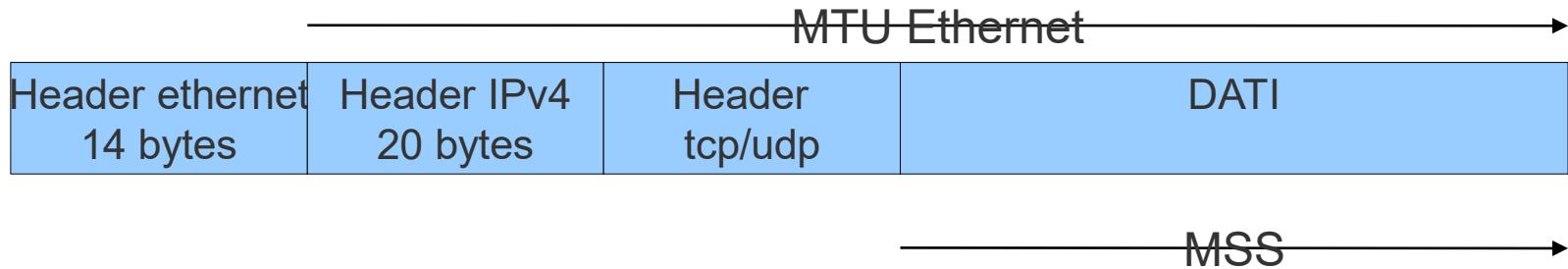
Segmentazione

Il mittente fraziona il flusso dell'applicazione in segmenti che avranno una dimensione massima detta MSS (Maximum Segment Size).

I segmenti vengono consegnati al layer Network (IP) il quale si occuperà della consegna all'host di destinazione.

Se durante il tragitto viene incontrato un Link con MTU inferiore alla dimensione del pacchetto il protocollo IP frammenterà il pacchetto in 2 o più parti, per poi ricomporle a destinazione.

Per evitare la frammentazione normalmente viene definito l'MSS in base al MTU dell'interfaccia locale (meno i byte dell'header TCP/UDP e i byte dell'header IP).



Il modello client/server

La Berkeley Socket Library utilizza un modello di comunicazione di tipo **Client/Server**:

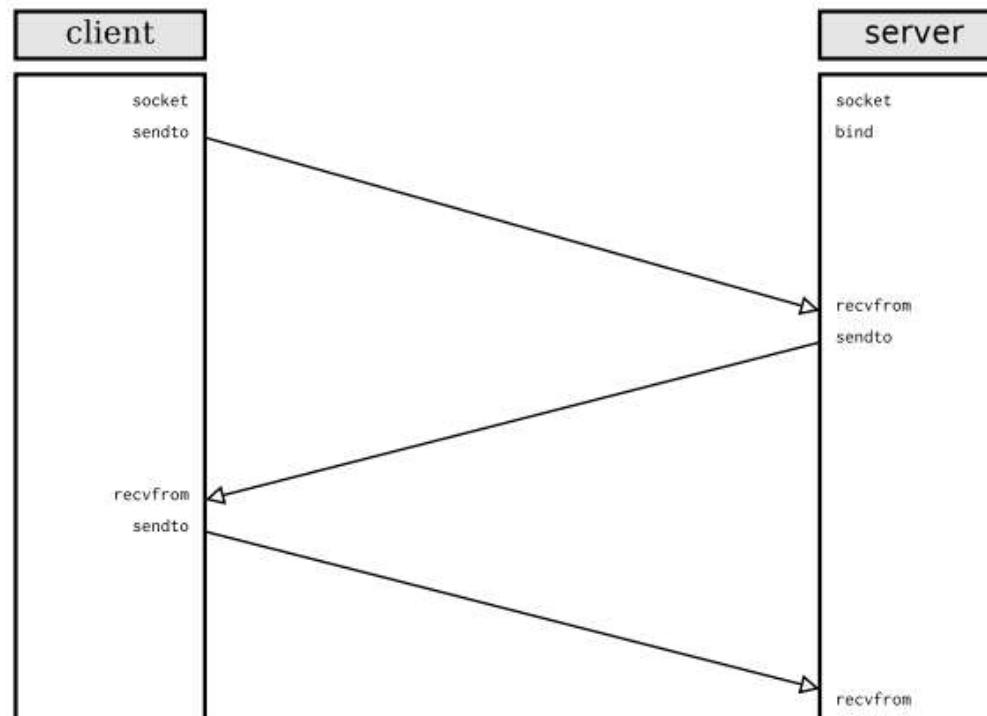
- ▶ un estremo “**Server**” è sempre in ascolto su una porta stabilita.
 - ▶ La primitiva **bind()** assegna un indirizzo locale (Porta) ad un socket.
- ▶ L'altro estremo “**Client**” prenderà contatto con il server specificandone il socket.
 - ▶ Il client per poter contattare il server deve quindi conoscerne indirizzo IP e porta.
 - ▶ La porta utilizzata dal client apparirà al server nell'intestazione di trasporto, quindi la porta del client non deve essere nota a priori. Generalmente viene determinata dinamicamente dal sistema operativo al momento della richiesta di connessione.

Programmazione Servizi a Datagrammi

Quando **il client** invia il messaggio con **sendto()** riceve dal sistema operativo un numero di porta dinamico.

Al successivo (eventuale) **recvfrom()** il client si mette in ascolto sulla stessa porta, nota al server poiché contenuta nel messaggio inviato dal client.

Il server inizia con un **recvfrom()** e la sua porta di ascolto deve essere stabilita dall'applicazione mediante la primitiva **bind()**.



<https://gapil.gnulinux.it/fileshare/gapil.pdf>

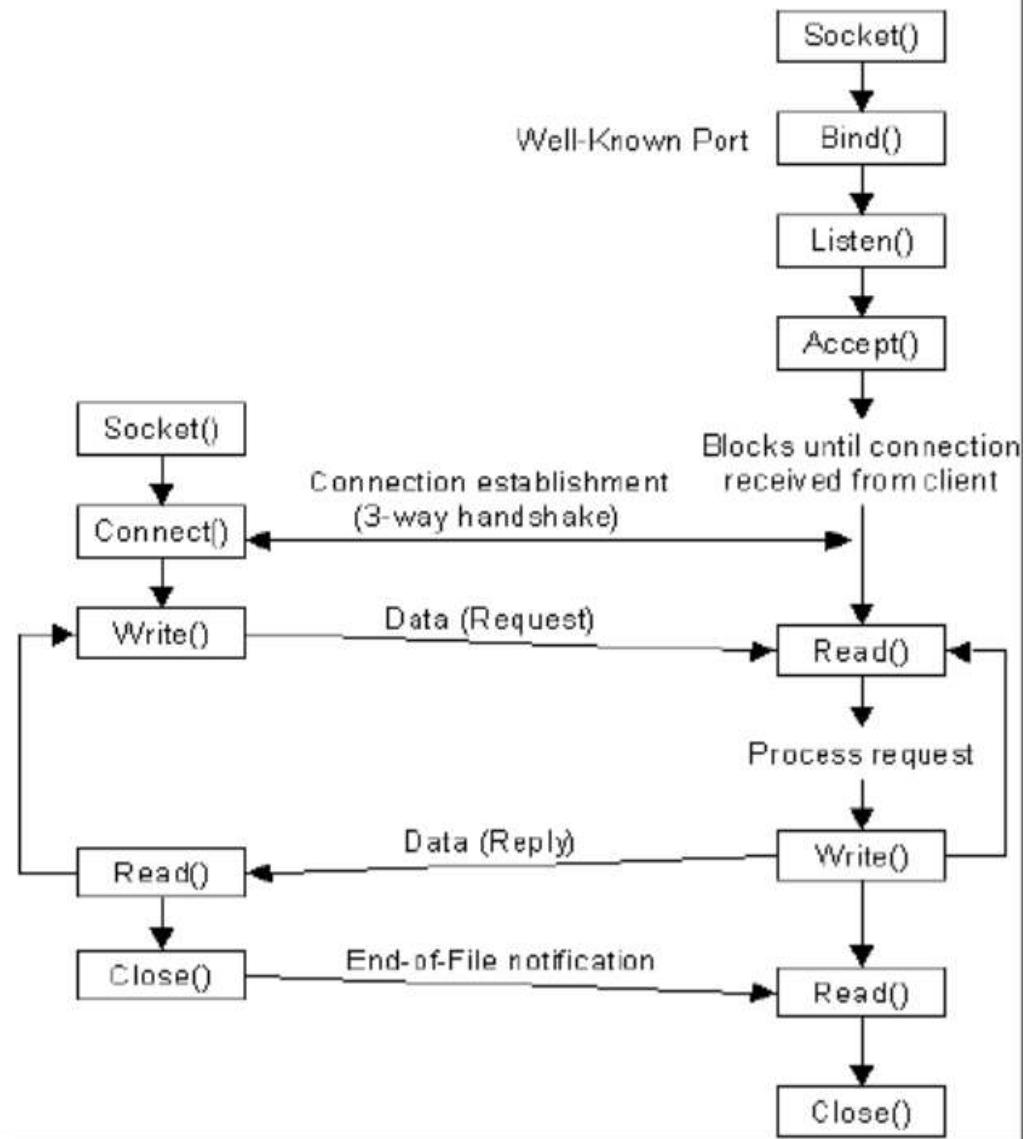
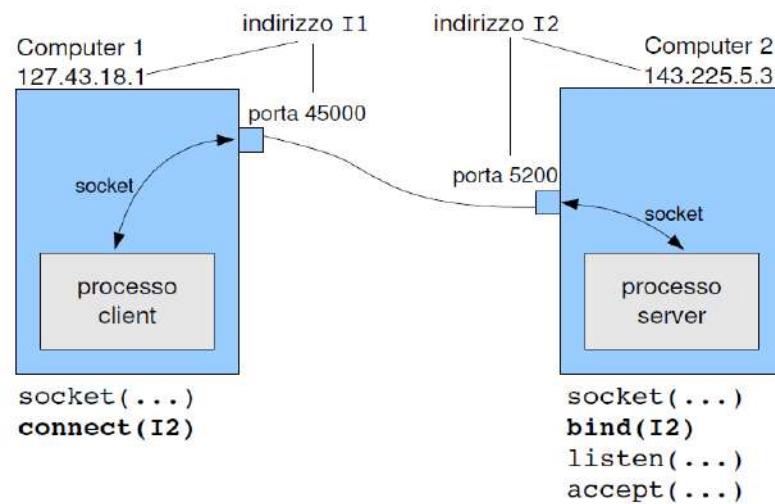
Programmazione Servizi Connection-Oriented

Per i servizi “**Connection Oriented**” (**TCP**) la libreria fornisce:

- ▶ la primitiva **listen()** predisponde le code di attesa per i processi client che accederanno contemporaneamente al servizio
- ▶ **accept()** è una primitiva bloccante che consente al server di mettersi in ascolto sulla porta. Quando arriva una TPDU il server crea un nuovo socket con le stesse proprietà di quello originale e ritorna un file descriptor per esso. Il server può creare un nuovo processo (fork) o un nuovo thread per gestire la connessione sul nuovo socket e tornare ad aspettare la prossima connessione.
- ▶ **connect()** è utilizzata dal client per aprire una connessione.
- ▶ Quando la connessione è instaurata la distinzione tra client e server non esiste più anche se normalmente il primo invio di dati viene fatto dal client con la primitiva **send()** la cui corrispondente **recv()** deve essere attivata dall'altro estremo.

*Nota: In Unix/Linux si possono usare anche le primitive **read()/write()***

Programmazione Servizi Connection-Oriented



Il protocollo UDP

- ▶ Descritto in [RFC768](#), offre alle applicazioni un modo per inviare datagrammi senza dover stabilire una connessione.
- ▶ L'unica differenza importante rispetto a IP è l'aggiunta delle porte di origine e destinazione necessarie per il demultiplexing.
- ▶ L'intestazione UDP contiene inoltre la lunghezza del segmento (header+dati) e il checksum facoltativo che è la somma delle sequenze di 16 bit in complemento a 1.
- ▶ UDP Viene utilizzato per
 - ▶ l'implementazione di protocolli applicativi che richiedono lo scambio di brevi messaggi (esempi: **DHCP**, **DNS**, **TFTP**)
 - ▶ la costruzione (a livello applicativo) di servizi di trasporto più astratti denominati “protocolli Middleware”. (esempi: **RPC** e **RTP**).
 - ▶ Comunicazioni multicast o broadcast

Bit	0	16	31
Porta di provenienza UDP		Porta di destinazione UDP	
Lunghezza del messaggio UDP		Checksum UDP	
Dati			
• • •			

Il protocollo TCP

- ▶ Descritto in [RFC 793](#) per fornire un **flusso di Byte** end-to-end affidabile a partire da un servizio di rete inaffidabile (IP).
- ▶ Le connessioni TCP sono **full-duplex e Unicast** (no multicast, no broadcast)
- ▶ TCP riceve flussi di byte dai processi locali, li spezza in segmenti e li spedisce in datagrammi IP separati.
- ▶ L'applicazione che spedisce (`send()`) consegna i dati in un buffer di spedizione. I byte possono essere raggruppati (o frazionati) in segmenti da consegnare al livello rete.
- ▶ Segmenti di max 64KB, ma quasi sempre MSS=1460 byte che, con le aggiunte degli header TCP e IP, arriva a 1500 che è l'MTU di Ethernet
- ▶ Il flag “PUSH” può essere usato per l'invio non ritardato.
- ▶ Il livello TCP di destinazione scrive i segmenti nel buffer di destinazione e consegna all'applicazione (`recv()`) tutti i byte riscontrati (ricevuti in ordine), ricostruendo il flusso originale.

Servizio orientato alla connessione

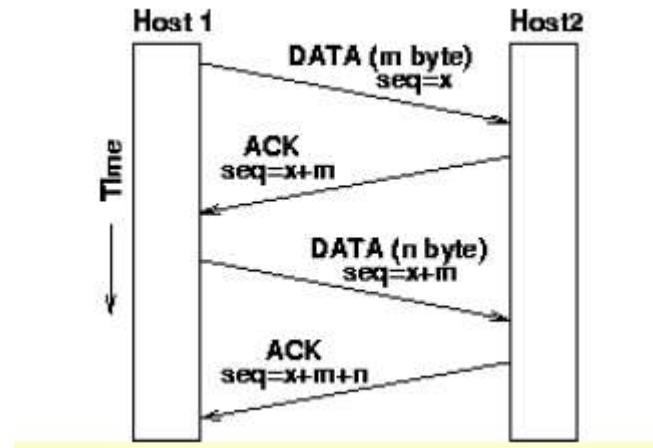
Le problematiche per questo tipo di servizio (analogamente al livello Data_Link) sono le seguenti:

- ▶ **Come si attiva una connessione**
- ▶ **Come si chiude una connessione**
- ▶ **Come si controlla l'ordinamento dei dati**
- ▶ **Come si controlla il flusso**
- ▶ **Come si gestiscono gli eventuali errori o perdite di pacchetti**

Corretta consegna e ordinamento dei segmenti

Il **riscontro** (conferma dell'avvenuta ricezione di un segmento di dati) o ACKnowledgement , abbinato al **numero di sequenza** attribuito ad ogni pacchetto dati (esempio tftp) o a ogni byte del flusso (esempio tcp), rappresentano un strumento molto utilizzato per la verifica della **corretta consegna e dei pacchetti e del relativo ordinamento**. Il protocollo consiste nei seguenti passi:

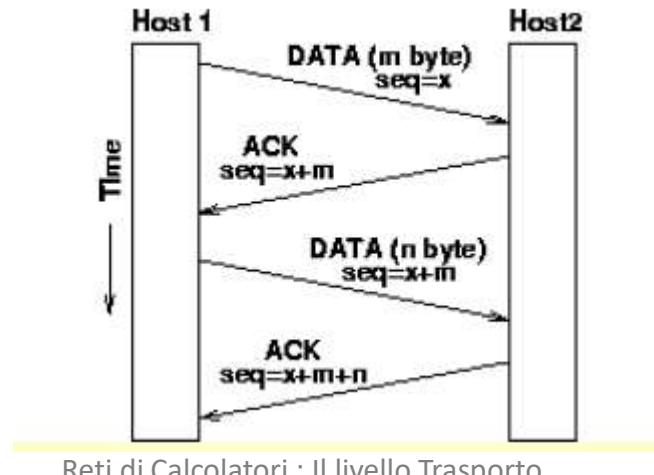
- ▶ Il mittente invia un **segmento di dati** assieme ad un numero progressivo
- ▶ Il destinatario invia un pacchetto con un Flag (**ACK**) attivo e il numero del byte (o del segmento) ricevuto correttamente
- ▶ Il mittente attiva un **Retransmission Time Out (RTO)** per ogni segmento inviato.



Corretta consegna e ordinamento dei segmenti in TCP

Per l'ordinamento dei dati TCP usa la numerazione dei Byte:

- ▶ Il numero di sequenza è il numero del primo byte dei dati contenuti nel segmento
- ▶ Il numero di riscontro, che accompagna l'ACK, è il numero del prossimo byte che il destinatario si aspetta di ricevere
- ▶ Il numero iniziale della sequenza non è 0, ma è determinata in modo da evitare che in seguito alla reinizializzazione di una connessione si faccia confusione tra vecchi e nuovi pacchetti
- ▶ Il mittente gestisce **un unico timer** per la ritrasmissione (**Retransmission Time Out - RTO**), basato sul RTT e associato al più vecchio segmento non riscontrato. Quando arriva una notifica intermedia, si riavvia il timer sul più vecchio segmento non riscontrato.
- ▶ Se non riceve ACK di un segmento ricomincia a spedire dall'ultimo byte riscontrato (GoBack-N) a meno che non sia concordato il Selective ACK (TCP con SACK).



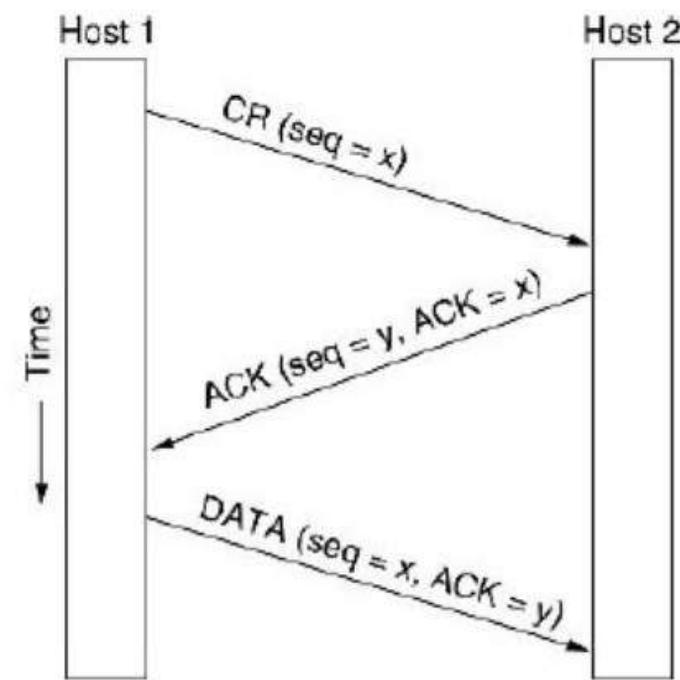
Attivazione della connessione

Il dialogo tra client e server per l'attivazione di una connessione deve tenere conto dell'inaffidabilità della rete sottostante.

Il problema maggiore è dato dai possibili duplicati ritardati che non devono essere confusi con nuove connessioni.

La soluzione proposta da **Tomlinson** (1975) è un meccanismo di HandShaking a 3 vie:

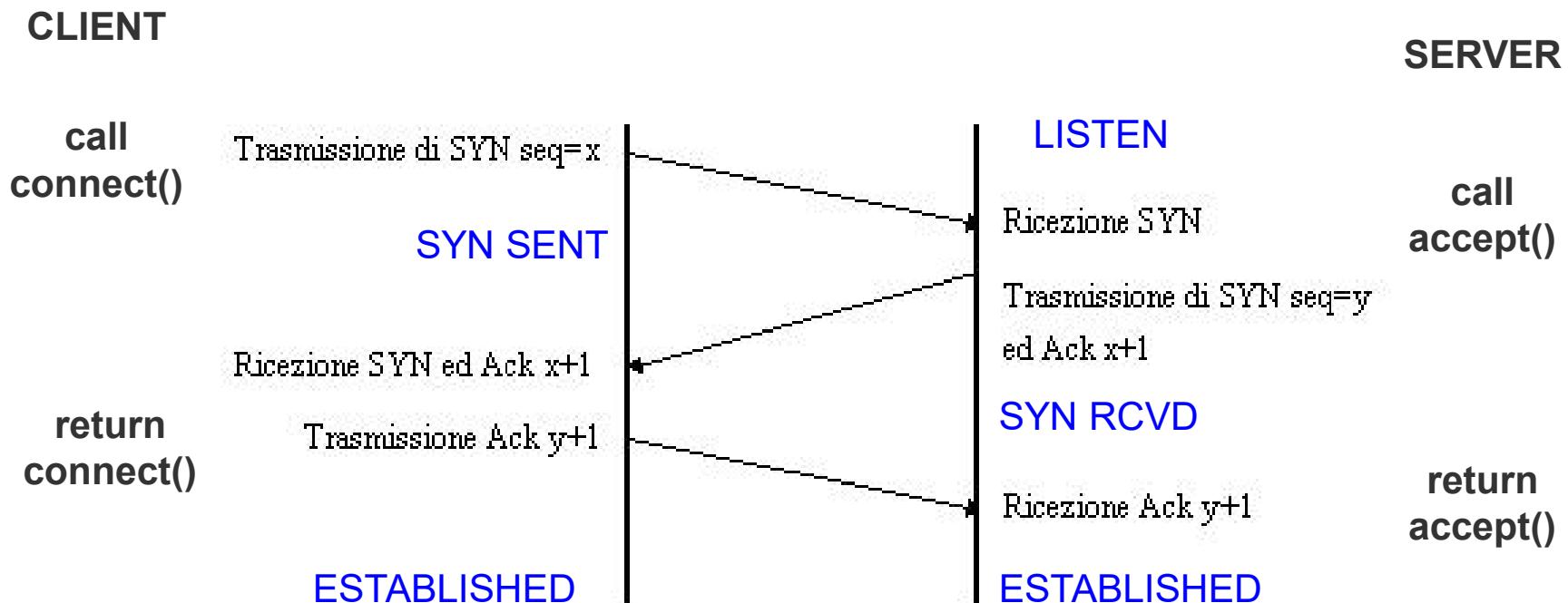
- 1) Il client invia un segmento di Connection Request (CR) con un valore iniziale di sequenza
- 2) Il server risponde con un ACK che riscontra il valore di sequenza proposto dal Client e propone un valore iniziale di sequenza per il senso inverso (da server a client).
- 3) Il client invia un terzo segmento con ACK e il riscontro della sequenza del server (che eventualmente può anche trasportare i primi dati)



Apertura di una connessione TCP

La soluzione in uso in TCP è derivata dall'algoritmo di Tomlinson:

- 1) La CONNECT sul **client** invia un segmento con SYN=1, ACK=0, seq=x (random)
- 2) Se il **server** è in ascolto (LISTEN) e accetta la connessione, risponde con un segmento in cui ACK=1, SYN=1, ACKseq=x+1 (il destinatario riscontra il byte numero x e dichiara che x+1 è il prossimo byte che si aspetta di ricevere) e seq=y (random)
- 3) Il **client** termina l'apertura riscontrando la sequenza del server: ACK=1, ACKseq=y+1



Chiusura della connessione TCP

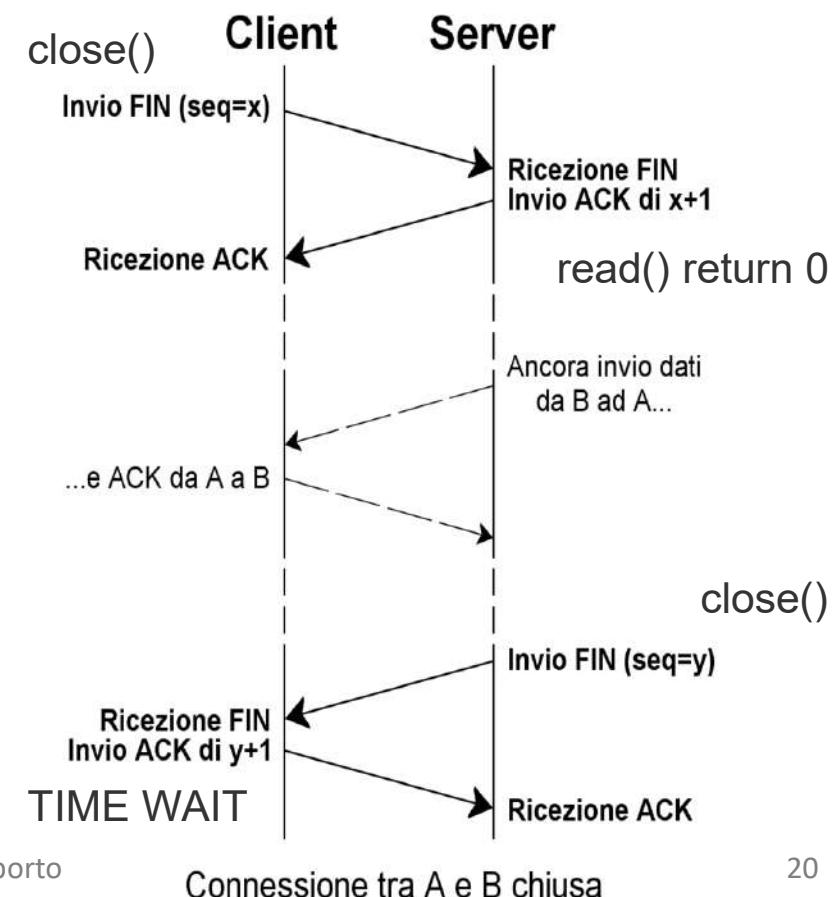
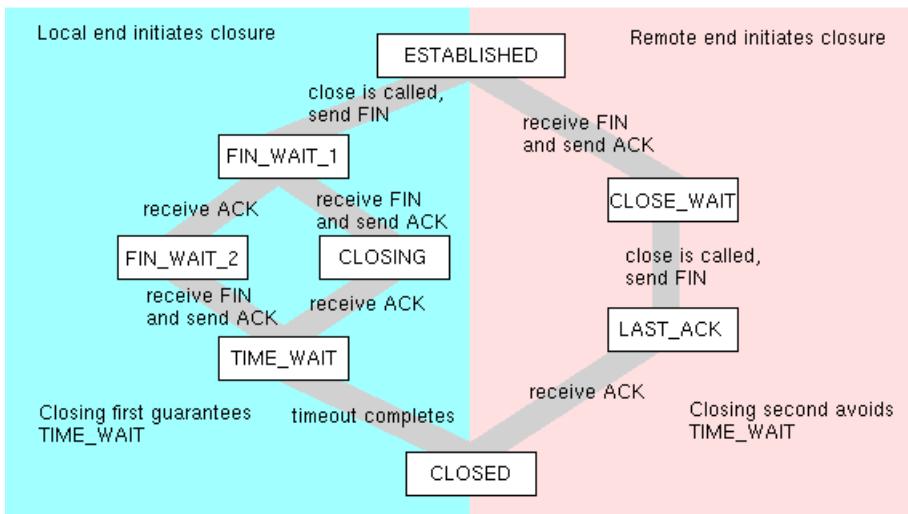
Si utilizza un handshake a 2 vie per ogni direzione.

Generalmente viene fatto con 3 segmenti, inviando il secondo FIN assieme all'ACK

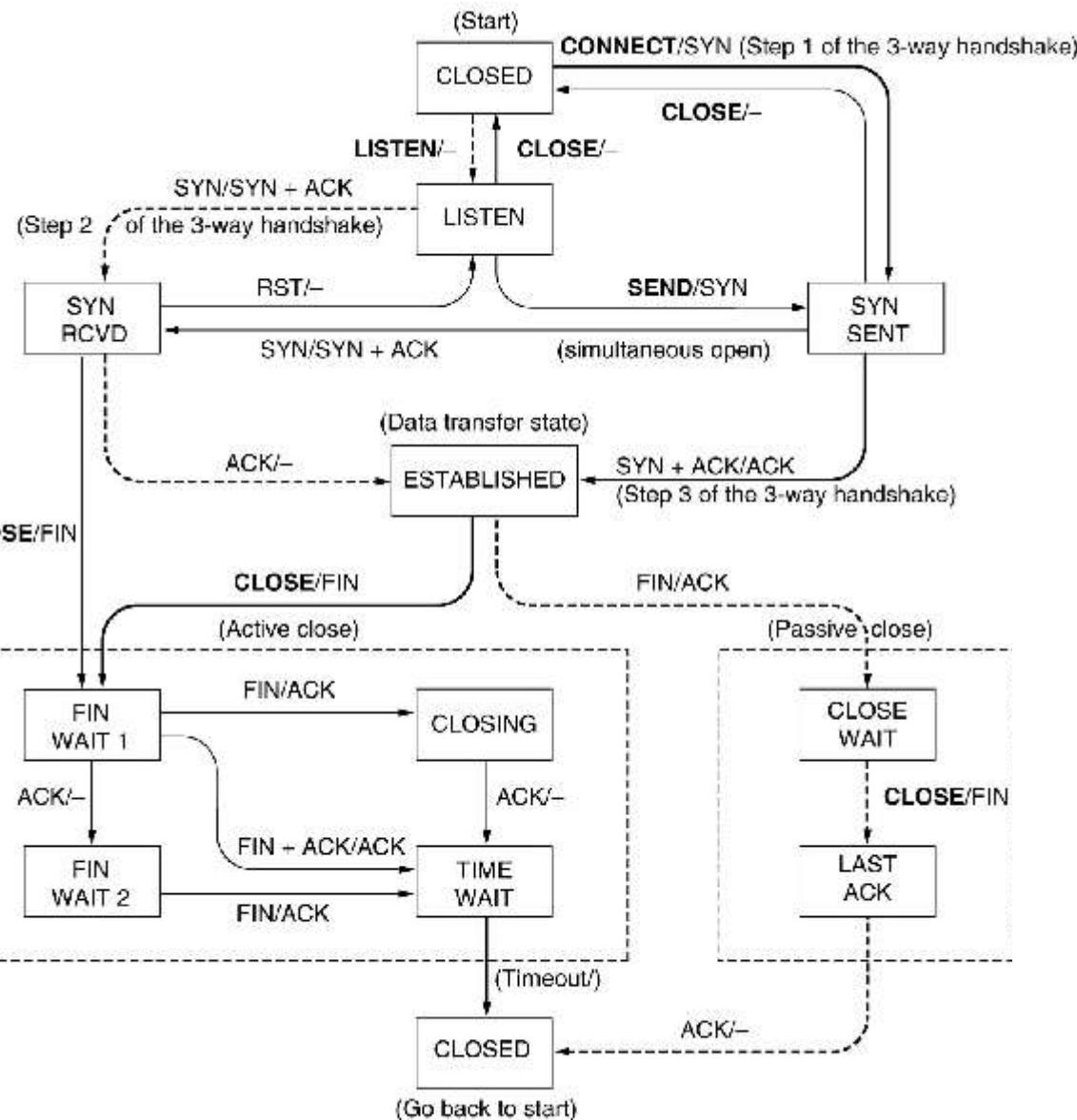
La primitiva `close()` determina l'invio del FIN, marca come chiuso il canale e ritorna immediatamente. Il canale non è più utilizzabile con `read()` o `write()`.

Se una risposta FIN non arriva entro 2 RTT il mittente FIN rilascia la connessione.

TIME WAIT attende per 2 MSL (Maximum Segment Lifetime) l'arrivo di eventuali pacchetti ancora in rete.
MSL è una stima del tempo di vita di un segmento. In Linux è tipicamente 30 s.



Gli stati della connessione TCP



Linee continue grosse: client
 Linee tratteggiate: server
 Linee sottili: eventi inusuali

State	Description
CLOSED	No connection is active or pending
LISTEN	The server is waiting for an incoming call
SYN RCV	A connection request has arrived; wait for ACK
SYN SENT	The application has started to open a connection
ESTABLISHED	The normal data transfer state
FIN WAIT 1	The application has said it is finished
FIN WAIT 2	The other side has agreed to release
TIMED WAIT	Wait for all packets to die off
CLOSING	Both sides have tried to close simultaneously
CLOSE WAIT	The other side has initiated a release
LAST ACK	Wait for all packets to die off

Buffering

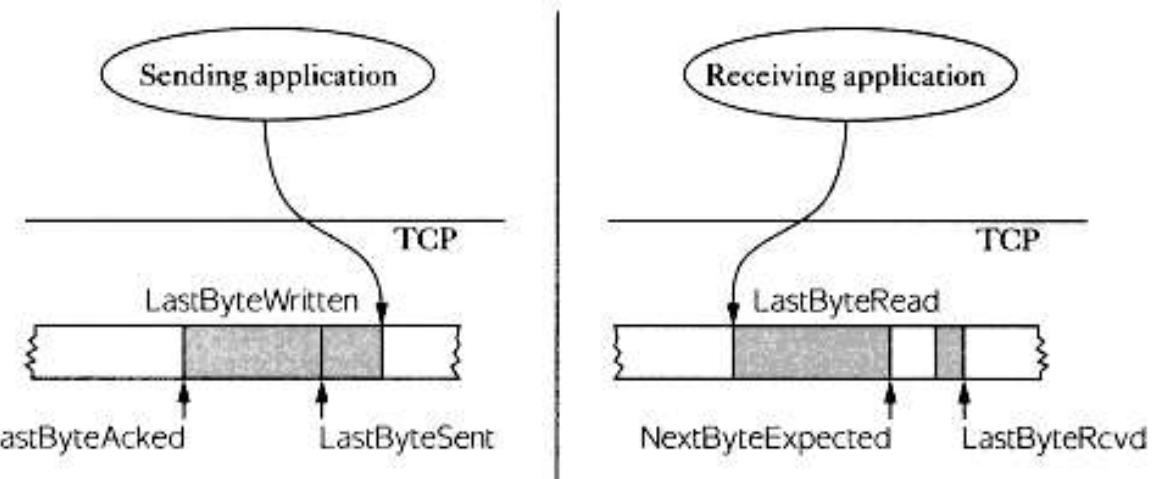
Per ogni connessione TCP è necessario un buffer (coda circolare) di trasmissione e un buffer di ricezione poiché i segmenti potrebbero andare perduti / fuori ordine e perché i processi di scrittura e lettura potrebbero lavorare a diverse velocità

Il buffer di trasmissione contiene:

- dati spediti ma non ancora riscontrati (tra LastByteSent e LastByteAcked)
- dati ancora da spedire (dopo LastByteSent)
- Spazio libero

Il buffer di ricezione contiene:

- dati ricevuti e riscontrati non ancora letti dall'applicazione
- dati ricevuti non ancora riscontrati (tipicamente dati ricevuti fuori ordine)
- spazio libero



I buffer in Linux

In ambiente Linux possiamo controllare i buffer nel seguente modo:

```
sysctl -a | grep tcp  
net.ipv4.tcp_rmem= 4096 87380 174760 (buffer ric. min-init-max)  
net.ipv4.tcp_wmem= 4096 16384 131072 (buffer sped. min-init-max)
```

La dimensione dei buffer può essere modificata con setsockopt().

SO_RCVBUF SO_SNDBUF Imposta dimensioni del Buffer di Ricezione/Trasm

```
//Esempio di raddoppio del buffer del ricevente  
int rcvBufSize;  
int sockOptSize;  
sockOptSize = sizeof(rcvBufSize); //preleva la dimensione di origine del buffer  
if(getsockopt(m_socket, SOL_SOCKET, SO_RCVBUF, &rcvBufSize, &sockOptSize) < 0)  
/*gestione errore*/  
printf("initial receive buffer size: %d\n", rcvBufSize);  
RcvBufSize *=2; //raddoppia la dimensione del buffer  
if(setsockopt(m_socket, SOL_SOCKET, SO_RCVBUF, &rcvBufSize, sizeof(rcvBufSize)) < 0)  
/*gestione errore*/
```

Socket Non Blocking

La **send()** è per default bloccante; si blocca quando il buffer in trasmissione è pieno e ritorna quando si libera spazio nel buffer di trasmissione. Se lo spazio nel buffer è insufficiente per i dati da spedire viene effettuata una scrittura di una porzione di dati minore o uguale alla dimensione del buffer libero, e la send() restituisce il numero di byte scritti.

Se il **buffer è pieno** e il socket è impostato come **non bloccante** non ci sarà nessun blocco ma ritornerà un -1 settando la variabile di errore **EWOULDBLOCK**.

La **recv()** è per default bloccante; si blocca quando il buffer in ricezione è vuoto e ritorna quando ci sono dati nel buffer. Il numero di byte letti può essere inferiore al numero di byte richiesto.

Ritorna 0 quando non ci sono dati nel buffer e l'altro peer ha chiuso la connessione.

Se il **buffer è vuoto** e il socket è impostato come **non bloccante** non ci sarà nessun blocco ma ritornerà un -1 settando la variabile di errore **EWOULDBLOCK**

Per mettere il socket in modalità non bloccante:

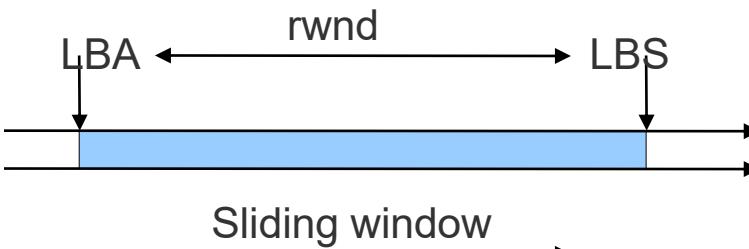
```
int flags, sockfd;
sockfd = socket(...);
if ( (flags=fcntl(sockfd,F_GETFL,0)) <0 ) exit(1);
flags |= O_NONBLOCK;
if ( fcntl(sockfd,F_SETFL,flags) <0 ) exit(1);
```

Controllo di Flusso : Sliding Window

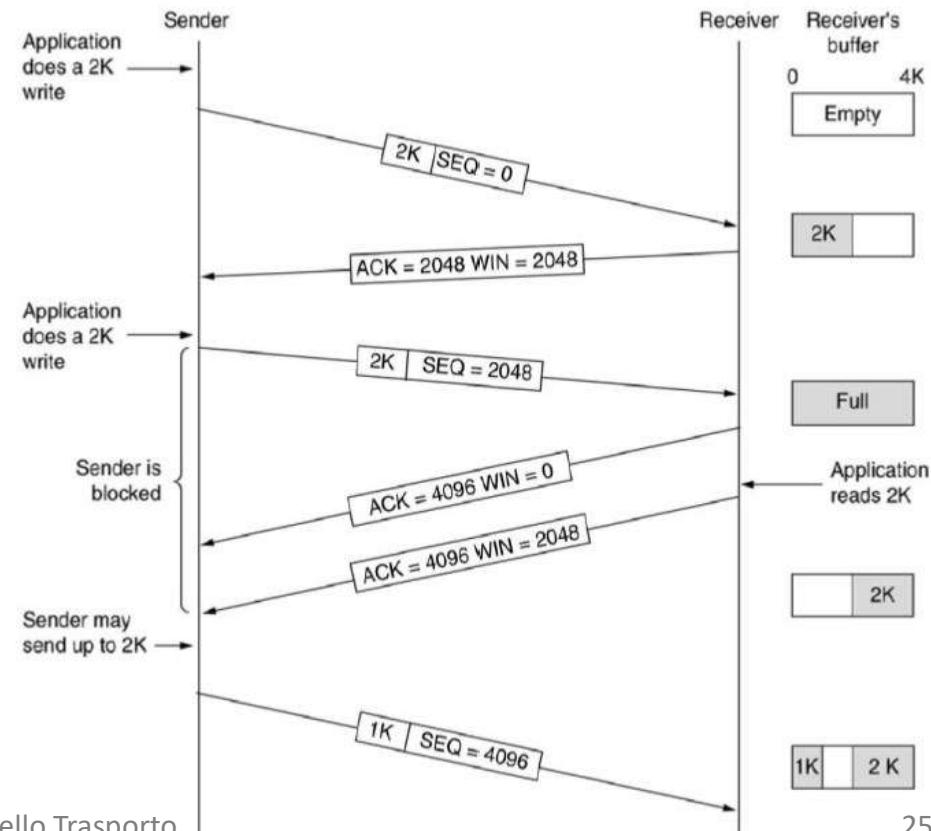
Per il controllo del flusso e l'ottimizzazione del throughput della rete in TCP si utilizza il meccanismo denominato **Sliding Window** (finestra scorrevole):

- ▶ Il **Ricevente** annuncia al trasmettitore la **Receiver Window Size (rwnd)**, che generalmente corrisponde al numero di byte liberi sul buffer di ricezione e indica quanti byte possono essere inviati a partire dall'ultimo riscontrato.
- ▶ Il trasmettitore può inviare anche più dati senza riscontro, purché il numero di byte non riscontrati non ecceda rwnd:

$$\text{LastByteSent} - \text{LastByteAcked} < \text{rwnd}$$



Il ricevitore può bloccare la trasmissione (ad eccezione degli Urgent Data) inviando $\text{rwnd}=0$ (Stop-and-wait)



La Trama TCP

Porta di provenienza e destinazione identificano gli estremi della connessione.

Il Numero Sequenziale è il contatore del flusso di byte spediti. Indica il numero del **primo byte** di dati contenuto nel segmento.

Il Numero di Riscontro è il contatore del numero di byte ricevuti. Indica il numero del **prossimo byte che il destinatario si aspetta di ricevere**.

HLEN (parole di 32 bit dell'Header) è necessario perché il campo opzioni è variabile.

Bit	0	4	10	16	24	31		
	Porta di Provenienza				Porta di Destinazione			
	Numero Sequenziale							
	Numero Riscontro							
HLEN	Riservato	Bit Codice			Finestra			
	Checksum		Puntatore Urgente					
	Options				Padding			
	Dati							
	• • •							

La Trama TCP : I bit di codice

Dopo HLEN ci sono 4 bit riservati per sviluppi futuri, poi abbiamo 8 bit di codice.

Se attivi (posti a 1) significano:

- **CWR e ECE** vengono utilizzati quando è attivo ECN (gestione della congestione - RFC 3168)
 - **ECE (ECN-Echo)**: usato per mandare ad un host l'indicazione di rallentare
 - **CWR** e' generato dall'host per indicare che ha ridotto la finestra di congestione
- **URG**: si deve considerare il campo “**puntatore urgente**”
- **ACK** : si deve considerare il “**numero di riscontro**”
- **PSH** : il ricevente non deve bufferizzare, ma renderli subito disponibili all'applicazione
- **RST**: reset della connessione a causa di qualche tipo di problema.
- **SYN**: utilizzato nella fase di attivazione di una connessione
- **FIN** : utilizzato nella fase di rilascio di una connessione

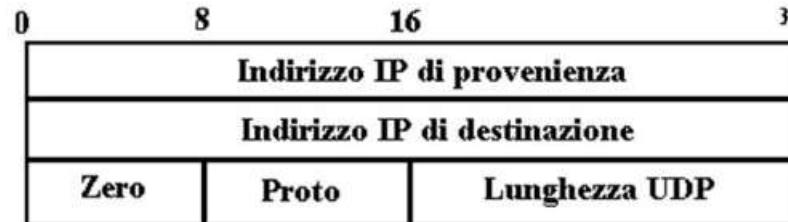
La Trama TCP: finestra, checksum e urgente

Finestra (16 bit): è la dimensione della Sliding Window, ovvero il numero di byte che il destinatario è in grado di ricevere a partire dall'ultimo byte riscontrato.

La dimensione massima sarebbe di 64KB , ma può essere aumentata attraverso il fattore di Scala della Finestra (vedi opzioni).

Il **CheckSum** (16 bit): somma in complemento a 1 delle sequenze di 16 bit del segmento TCP (header e dati) e la “pseudo-intestazione”

La **pseudo-intestazione** include ulteriori informazioni importanti di IP e TCP (IP source, IP dest, 0x00, 0x06, TCP Segment length), violando però l'indipendenza dei protocolli perché include dati del Layer IP.



Puntatore URGENTE (16 bit): Puntatore a un dato urgente, ha significato solo se il flag URG è impostato a 1 ed indica lo scostamento in byte dell'ultimo dato urgente. Tipicamente sono messaggi di controllo. Usato raramente.

La Trama TCP: opzioni

I campi opzionali dell'intestazione TCP vengono principalmente utilizzati nella fase di Handshake, nei segmenti SYN, per comunicare all'altro capo una serie di parametri utili a regolare la connessione.

Normalmente vengono usate le seguenti opzioni (dettagli nelle slide successive):

- **MSS**: massima dimensione accettata del segmento
- **SACK**: Selective ACKnowledgement
- **Fattore di scala della finestra**
- **timestamp**: (**TSval** , timestamp value)

Se i byte delle opzioni non sono multipli di 4 (parole di 32 bit) viene aggiunto un padding opportuno.

La Trama TCP: Opzione MSS

Questa opzione viene utilizzata per **concordare l' MSS ottimale (RFC 1191)**

La frammentazione introduce un **overhead sull'attività dei router**. Frammenti troppo piccoli determinano un **overhead sul traffico di rete**.

L'MSS (Maximum Segment Size) **ottimale** dipende dall'MTU minimo tra tutti gli MTU incontrati nel tragitto, ma questo dato non è noto quando si inizia una trasmissione e potrebbe cambiare nel tempo.

L'algoritmo generalmente utilizzato è descritto nell'RFC1191:

- Ciascun host determina la propria MSS ottimale in base alla MTU dell'interfaccia locale (meno 20 byte dell'header TCP e 20 dell'header TCP) e lo comunica all'altro host attraverso le **Opzioni dell'header TCP (*)**
- Il primo segmento dati viene inviato con il bit DF (Don't Fragment) settato a uno. Se durante il cammino si incontra un router con MTU inferiore questo invierà al mittente un pacchetto ICMP di errore che verrà utilizzato per correggere l'MSS.

(*) E' possibile determinare il valore dell'MSS locale con l'opzione del Socket **TCP_MAXSEG**

La Trama TCP: Opzione SACK

Per default TCP funziona con **GoBackN**: se il ricevente ottiene un segmento errato e N segmenti validi, riscontra sempre l'ultimo segmento valido prima dell'errore. Questo manda in TimeOut il mittente che rimanda tutti i Segmenti a partire da quello errato.

Per migliorare le prestazioni evitando la ritrasmissione di segmenti validi è stata proposta la tecnica **SACK (Selective ACK, RFC 2108 e 2883)**: il ricevitore indica al trasmettitore quali segmenti sono arrivati correttamente in modo che possa determinare quali segmenti devono essere rispediti.

Funziona con **2 opzioni dell'header TCP**:

- SACK Permitted : Viene incluso in un segmento SYN per indicare la capacità di gestire la tecnica SACK.
- SACK: viene utilizzato dal ricevente per comunicare le informazioni SACK (i segmenti ricevuti correttamente).

Ad esempio: blocco1 (primo-ultimo), blocco2 (primo-ultimo) , . . .

La Trama TCP: opzioni Window Scale e timestamp

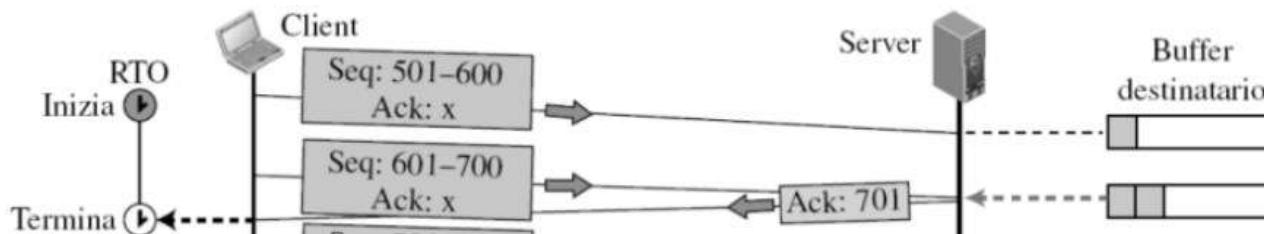
- **Window Scale:** La dimensione della finestra è scritta in un campo di 16 bit, consentendo quindi un valore massimo di 64KB. Nelle reti moderne questa dimensione massima è insufficiente. L'opzione window scale determina numero di shift a sinistra da applicare nell'interpretare il valore ricevuto. Ogni shift a sinistra corrisponde ad una moltiplicazione 2x.
- Ad esempio con Window Scale = 2 il valore massimo delle finestra è di 256KB.
- **timestamp:** (**TSval** , timestamp value) è un marcatore temporale spedito dal mittente e rimbalzato poi dal destinatario (**TSecr** , timestamp echo reply), per il calcolo del RTT (RTT = current time – TSecr).

Ottimizzazioni TCP: ACK delayed, ACK cumulativo

Quando il destinatario riceve un segmento in ordine può attendere fino a 500ms l'arrivo del prossimo segmento (**delayed acknowledgement, RFC 1122**)

Se durante l'attesa arriva un altro segmento in ordine risponde con un singolo **ACK cumulativo** che riscontra l'ultimo byte della sequenza.

Questa tecnica è utilizzata in molte implementazioni di TCP.

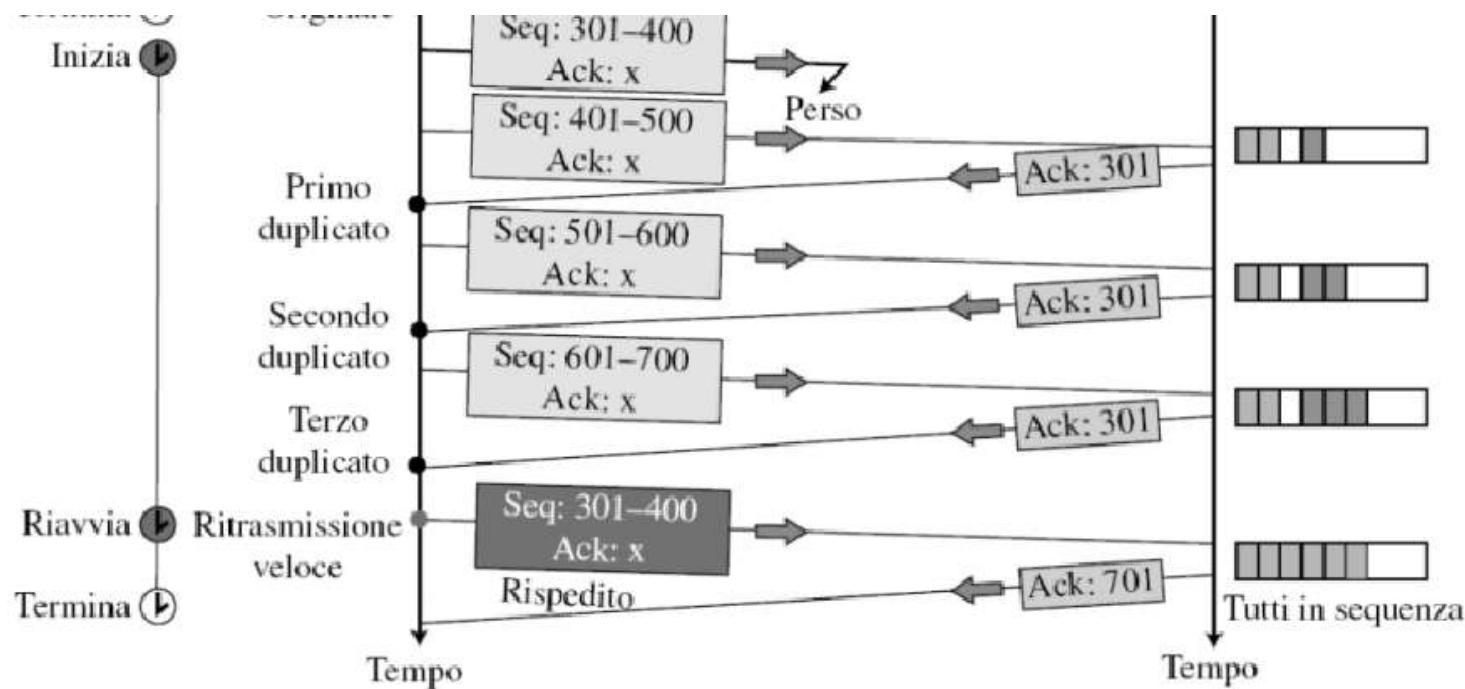


Ottimizzazioni TCP: fast retransmission

Se il destinatario riceve un segmento fuori ordine (successivo ad altri segmenti non ricevuti) invia un ACK in cui viene riscontrato l'ultimo segmento ricevuto in ordine.

Quando il mittente riceve 3 ACK che riscontrano lo stesso numero capisce che un segmento è andato perduto e lo ritrasmette, senza attendere lo scadere del timer (Fast Retransmission).

Il destinatario risponde con un singolo ACK che riscontra anche i successivi segmenti ordinati (ACK cumulativo).



Ottimizzazioni TCP: Algoritmo di Nagle e soluzione di Clark

Le prestazioni possono degradare in alcuni casi particolari quali:

- **il trasmettitore genera dati lentamente**; ad esempio quando si edita un file per ogni tasto premuto girano 4 pacchetti IP per un totale di 162 byte.
- **il ricevitore consuma dati lentamente**; ad esempio il destinatario pubblica finestre di pochi byte, perché l'applicazione legge pochi byte per volta, il mittente è costretto a spezzare il flusso in tanti segmenti (problema della finestra futile)

Per attenuare il problema **lato mittente** si usa **l'algoritmo di Nagle**:

- se il mittente ha **pochi byte da spedire** (a causa dell'applicazione o della finestra del destinatario) e **ci sono dati non riscontrati** → aspetta ACK, anche se la finestra scorrevole consentirebbe l'invio di altri dati.
- se il mittente ha **molti byte da spedire** oppure i **segmenti piccoli sono riscontrati** → spedisci subito

Nota: Questo algoritmo può essere disabilitato con l'opzione **TCP_NODELAY**

Esempio in C: `setsockopt (sock, SOL_TCP, TCP_NODELAY, ...);`

Per attenuare il problema **lato ricevente** si usa la **soluzione di Clark**:

- Se il ricevente pubblica finestre troppo piccole l'algoritmo forza il ricevitore ad attendere che la finestra raggiunga un valore minimo prima di comunicarlo al mittente.

Il Retransmission TimeOut (RTO) di TCP

Serve per decidere quando un pacchetto deve considerarsi perduto.

Deve essere almeno pari a RTT, ma deve aggiornarsi dinamicamente e deve gestire situazioni di congestione (backoff).

Algoritmo di Jacobson (1988):

Se l'ACK torna indietro prima dello scadere dell'RTO l'algoritmo calcola il valore del RTT (Round Trip Time) e aggiorna le variabili:

$$\text{RTTMedio} = \alpha \text{ RTTMedio} + (1-\alpha) \text{ RTT}$$

$$\text{DevMedia} = \alpha \text{ DevMedia} + (1-\alpha) \text{ abs}(\text{RTT} - \text{RTTMedio})$$

(α è il peso che si vuole dare ai precedenti valori medi, valore tipico 0.9)

$$\text{RTO} = \text{RTTMedio} + 4 \times \text{DevMedia}$$

Reti congestionate: Algoritmo di Backoff (di Karn)

Se l'RTO scade significa che la rete è congestionata. In questo caso l'algoritmo di Karn prevede di non aggiornare L'RTTMedio e raddoppiare l' RTO fino a quando i segmenti non arrivano a destinazione al primo tentativo

$$\text{RTO (i+1)} = 2 * \text{RTO(i)} \quad (\text{backoff esponenziale binario})$$

I timer di TCP in Linux

Oltre a RTO, che è il più importante, TCP gestisce altri Timer:

- **Timer di Persistenza:** viene attivato quando la finestra viene chiusa ($rwnd=0$). Se il pacchetto che riapre va perduto, allo scadere del timer il mittente invia una “window probe” che sollecita la rispedizione della finestra. Se la finestra è ancora 0 il timer viene reimpostato.
- **Timed Wait:** Tempo di attesa dopo un FIN. Prima di rilasciare la connessione viene attivato questo timer per gestire eventuali pacchetti circolanti dopo la chiusura. Generalmente corrisponde a doppio del tempo di vita massimo di un pacchetto.
- **Timer di Keepalive:** parte quando la linea TCP è inattiva. Se arriva a zero TCP invia un ACK; se non riceve risposta la connessione viene considerata interrotta.

Esempi con Linux:

```
sysctl -a | grep tcp | grep time  
net.ipv4.tcp_fin_timeout = 60          # Time wait: 1 minuto  
net.ipv4.tcp_keepalive_time = 7200      # keepalive: 2 ore  
  
echo 900> /proc/sys/net/ipv4/tcp_keepalive_time # keepalive a 15 min.
```



UNIVERSITÀ
DI PARMA

DIPARTIMENTO DI SCIENZE MATEMATICHE, FISICHE ED INFORMATICHE
Corso di Laurea in Informatica

Congestione e QoS

RETI DI CALCOLATORI - a.a. 2023/2024

Roberto Alfieri

Il livello trasporto: sommario

PARTE I

- ▶ Scopo del livello Trasporto
- ▶ L'indirizzamento
- ▶ Il modello client/server
- ▶ Il protocollo UDP
- ▶ I servizi orientati alla connessione
- ▶ Il protocollo TCP

PARTE II

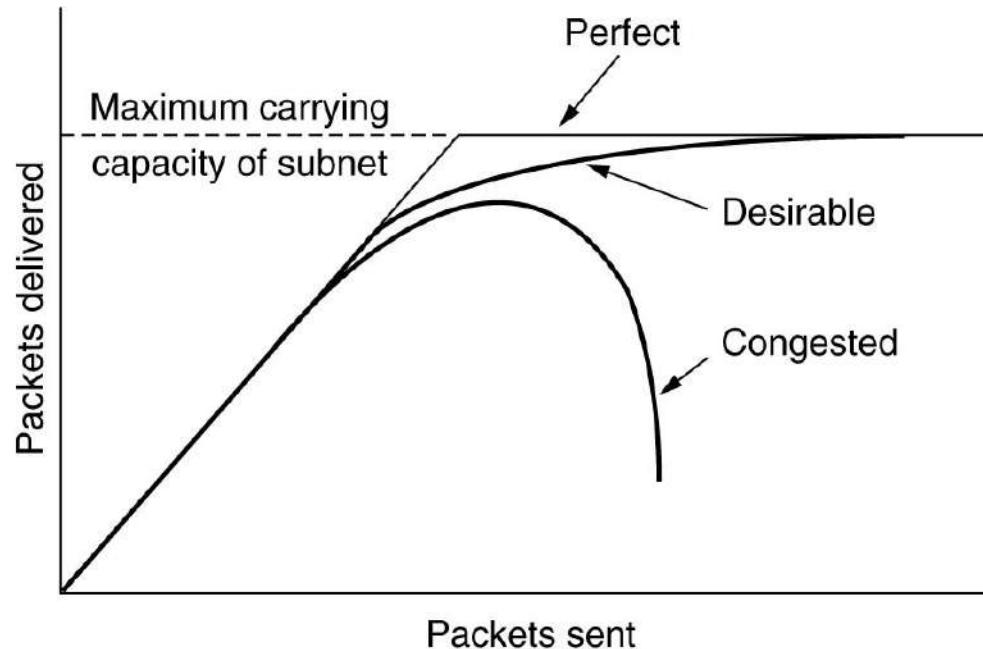
- ▶ Congestione, Qualità del Servizio
- ▶ Algoritmi Slow start, Tahoe, Fast Recovery, RED, segnalazione esplicita
- ▶ QoS e controllo del traffico
- ▶ Servizi differenziati e integrati

RIFERIMENTI

- ▶ *Reti di Calcolatori, A. Tanenbaum, ed. Pearson*
- ▶ *Reti di calcolatori e Internet, Forouzan , Ed. McGraw-Hill*

Controllo della Congestione

Quando troppi pacchetti sono presenti in una porzione della rete le prestazioni si degradano e la rete si dice congestionata. La congestione ha effetto su tutti i parametri della rete: velocità, ritardo, jitter e affidabilità.



Per **Controllo della Congestione** si intendono le procedure per prevenire la congestione prima che si verifichi (controllo proattivo) o gestirla quando si è verificata (controllo reattivo).

Coinvolge il comportamento dei terminali (Host) e dei nodi di transito (Router).

Può essere svolto a vari livelli (generalmente a livello Rete e a livello Trasporto in TCP).

Controllo della congestione di TCP

In TCP l'host gestisce la congestione mediante l'introduzione di una ulteriore finestra denominata **Congestion Window** (cwnd).

La finestra effettivamente utilizzata in trasmissione (awnd) è la finestra più piccola tra la finestra indicata dal ricevente (rwnd) e la finestra di congestione (cwnd)

$$\text{awnd} = \min(\text{cwnd}, \text{rwnd}) \geq \text{LastByteSent} - \text{LastByteAcked}$$

La Congestion Window **cwnd** dovrà essere regolata mediante opportuni algoritmi.

I principali sono:

- controllo proattivo: Algoritmo Slow Start. Nelle nuove connessioni TCP la cwnd inizia con un valore basso e cresce lentamente.
- controllo reattivo: Algoritmi Tahoe e Reno. Si assume che i pacchetti persi siano causati da congestione. La cwnd viene ridotta a seguito di un time-out.
- controllo reattivo: La cwnd viene ridotta a seguito di segnalazione implicita (algoritmo RED) o esplicita (ECN) di congestione.

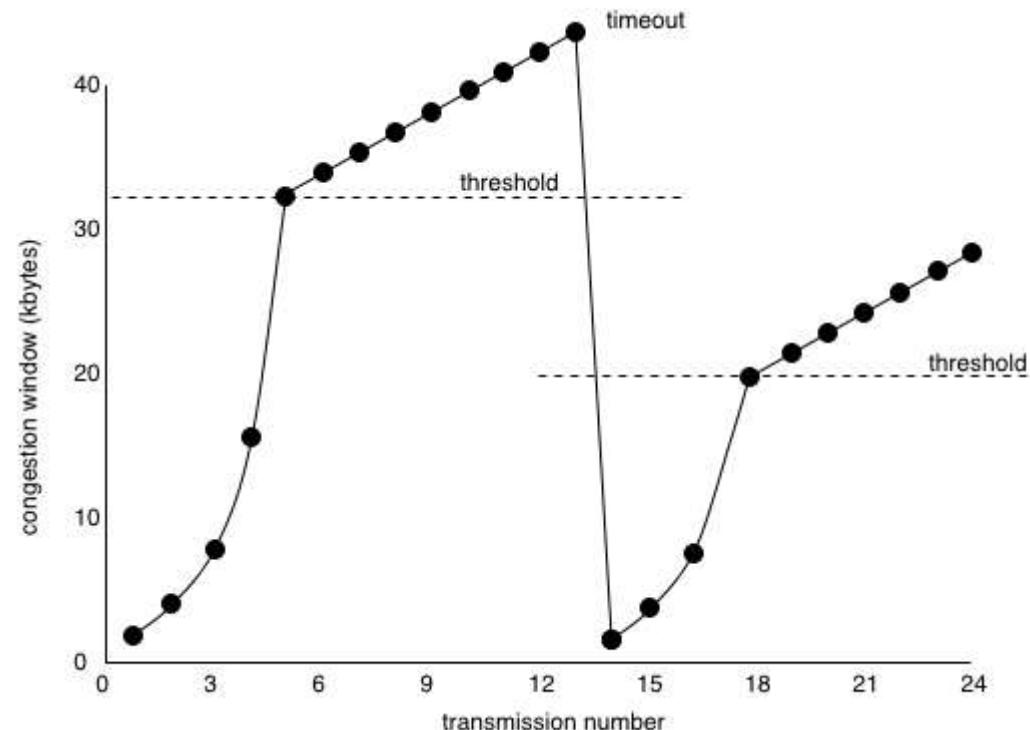
Gestione della cwnd: Algoritmo Slow Start

1) Controllo proattivo: **Slow Start**

Il mittente imposta $cwnd=MSS$ e la raddoppia ad ogni invio fino a quando:

- $cwnd$ raggiunge la dimensione della finestra del ricevente $rwnd$
- $cwnd$ raggiunge una **soglia** (**threshold**, valore iniziale tipico 64KB)

Raggiungendo la soglia l'aumento diventa lineare



2) Controllo reattivo: **TAHOE**

Se scade il timer siamo in congestione e si ritorna a Slow Start ($cwnd=MSS$) con soglia portata alla metà della corrente $cwnd$

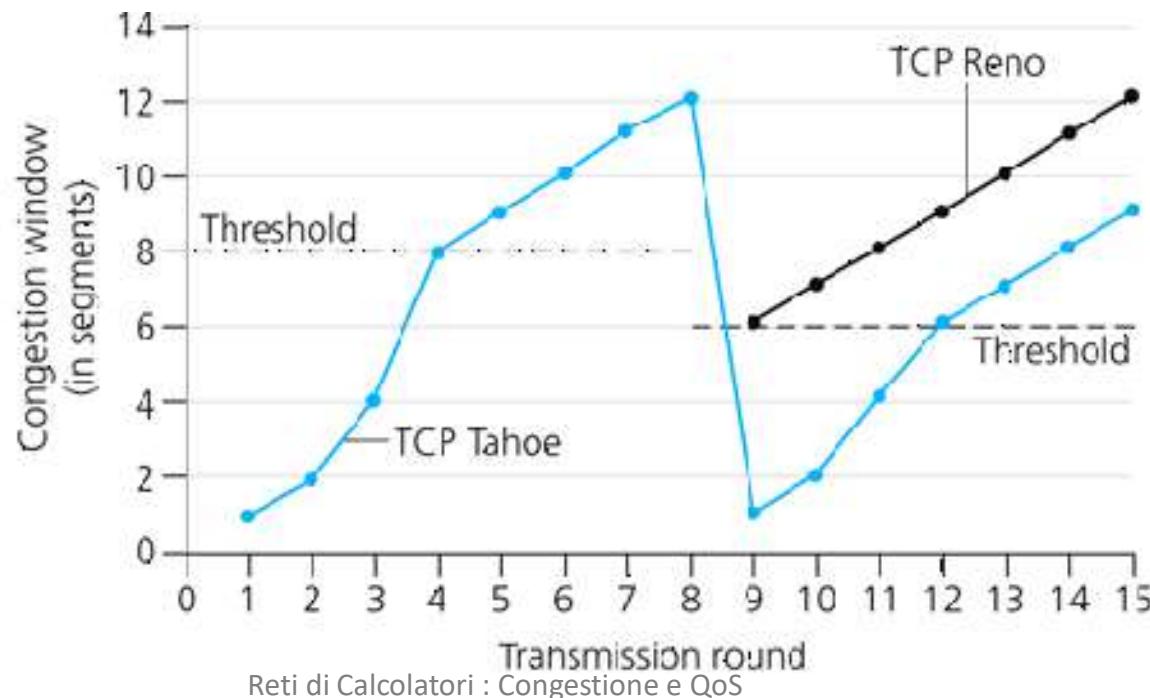
Il crollo repentino del $cwnd$ decongestiona la rete ma limita temporaneamente la velocità di ricezione / trasmissione dei dati.

Controllo reattivo con fast recovery (TCP Reno)

TCP Reno migliora il Tahoe distinguendo tra 2 motivi per la perdita di dati:

- 1) Dovuto al timeout del Timer → rete molto congestionata: cwnd= MSS
- 2) Ricezione di 3 ACK duplicati → Fast recovery: cwnd = new threshold

Infatti, se un segmento viene perduto ma non i successivi, ogni segmento arrivato fuori sequenza comporta la generazione di un ACK con la **conferma degli ultimi dati ricevuti in sequenza corretta**; siccome in virtù del meccanismo a finestra i segmenti sono spesso inviati uno di seguito all'altro in gruppi; se quello perso non è l'ultimo, è probabile che gli ACK arrivino prima dello scadere del timer.



Algoritmo RED (Random Early Detection)

RED è metodo (sia proattivo che reattivo) per gestire la congestione a livello Rete.

L' algoritmo interviene sulla coda di pacchetti nel buffer di spedizione dei router, definendo 2 soglie T_{min} e T_{max} .

Quando arriva un nuovo pacchetto P il router controlla il numero X di pacchetti in coda:

- Se $X < T_{min}$ P viene accodato
- Se $T_{min} < X < T_{max}$ P viene scartato (probabilità $p=f(X)$) o accodato ($p=1-f(X)$)
- Se $X > T_{max}$ P viene scartato

Segnalazione implicita di congestione

L'eliminazione precoce dei pacchetti comporta l'invio di 3 ACK duplicati o lo scadere dell'RTO del mittente, quindi rappresenta anche una **segnalazione implicita del router al mittente** riguardo una situazione di allarme, il quale interviene con un controllo reattivo (e.g. TCP Reno).

Segnalazione esplicita di congestione: ECN

Il router può avvisare esplicitamente il mittente mediante l'invio di un pacchetto speciale detto Choke packet. Quando il mittente riceve il Choke packet deve ridurre il traffico inviato.

La tecnica attualmente in uso è denominata ECN (Explicit Congestion Notification) ed è definita nell'RFC 3168 (vedi http://en.wikipedia.org/wiki/Explicit_Congestion_Notification)
ECN può lavorare sia a livello IP che TCP.

ECN in IPv4: per la segnalazione da Router a mittente

- utilizza 2 bit del campo DiffServ nell'intestazione IP (11 = Congestion Encountered)
Il router può utilizzare questo metodo anziché scartare il pacchetto con RED

ECN in TCP: per le segnalazioni end-to-end

- utilizza i bit ECN-Echo (ECE) e Congestion Window Reduced (CWR) di TCP.

In entrambi i casi (IP e TCP) il mittente reagisce a livello di trasporto attivando il controllo reattivo con Fast Recovery (come se fossero arrivati 3 ACK duplicati).

Nota: ECN è implementato in Linux 2.4+ e nei router Cisco dalla versione 12.2(8)

Qualità del Servizio (QoS)

Due processi che utilizzano la rete ricevono un servizio di comunicazione.

La Qualità del Servizio (QoS) fa riferimento all'aderenza del servizio ricevuto in relazione a **4 parametri primari della comunicazione:**

Affidabilità, Ritardo, Jitter e Banda, a cui l'applicazione è più o meno sensibile.

- ▶ Affidabilità : Garanzia della consegna dei dati spediti
- ▶ Ritardo : Tempo necessario per la consegna
- ▶ Jitter : Variabilità del ritardo.
- ▶ Banda : Velocità nella trasmissione dei dati

Gestione della QoS

La gestione della QoS (eventuale) può essere concordata tra utente e fornitore del servizio attraverso un accordo preliminare denominato **Service Level Agreement (SLA)** attraverso il quale il fornitore si impegna garantire una determinato livello di QoS.

La QoS può essere realizzata in due modi:

- ▶ per **Singolo Flusso**. Al momento dell'apertura del canale è possibile applicare specifiche tecniche di QoS quali **il controllo di accesso o la prenotazione di risorse**. Per poter effettuare la prenotazione è necessaria la “**commutazione di pacchetto a circuito virtuale**” in cui i flussi vengono specificati all'interno del pacchetto (**flow label**) e i router vengono individuati e riservati nella fase di attivazione della connessione:
 - In IPv4 è possibile realizzarla mediante isole MPLS.
 - IPv6 prevede l'etichetta di flusso (Flow Label) nell'intestazione, ma al momento non è utilizzata.
- ▶ per **Classi di Servizio** in cui vengono raggruppate le applicazioni con esigenze comuni rispetto ai parametri della comunicazione.
 - I servizi sono offerti da un insieme di Router che costituiscono un dominio amministrativo. L'amministrazione definisce una serie di classi di servizio. I pacchetti del mittente contengono un campo che consente ai Router di classificare il pacchetto e applicare Policy specifiche.

Esempi di Classi di Servizio : la rete ATM

L'ATM Forum ha definito 4 classi di servizio:

Classe A:

- ▶ **CBR** (Constant Bitrate): Velocità costante. Esempio telefonia.

Classe B:

- ▶ **VBR-RT** (Real Time Variable BitRate) where end-to-end delay is critical, such as interactive video conferencing.

Classe C:

- ▶ **VBR-NRT** (non-real time traffic), where delay is not so critical, such as video playback, training tapes and video mail messages.

Classe D: (Best Effort)

- ▶ **ABR** (Average BitRate) e **UBR** (Unspecified BitRate) : Esempio trasferimento File, visione di un film via Internet, ...

Classi di Servizio: Servizi Differenziati (DiffServ)

I Servizi Differenziati (DS) sono stati introdotti in Internet nel 1998 con l' [RFC 2474](#) per il supporto delle classi di servizio in IPv4 e IPv6.

Nelle **reti IPv4** viene utilizzato il campo **Type of Service (6 bit)** che con l'RFC 2474 diventa **Differentiated Service (DS) field** per la codifica delle Classi di Servizio.

In **IPv6** le classi possono essere codificate nel campo **Traffic Class**.

Le classi di servizio DiffServ più comuni sono:

- ▶ **Default Forwarding.** Best effort
- ▶ **Expedited forwarding (EF)** dedicated to low-loss, low-latency traffic, suitable for voice. Typical networks will limit EF traffic to no more than 30%.
- ▶ **Voice Admit (VA).** has identical characteristics to the EF , but Voice Admit traffic is also admitted by the network using a Call Admission Control (CAC) procedure.
- ▶ **Assured Forwarding (AF).** AF allows the operator to provide assurance of delivery as long as the traffic does not exceed some subscribed rate. Traffic that exceeds the subscription rate faces a higher probability of being dropped if congestion occurs.

Riferimenti: https://en.wikipedia.org/wiki/Differentiated_services

Implementazione della QoS: Controllo del traffico

Le principali tecniche per l'implementazione della QoS basata sulle Classi sono relative alla gestione e il controllo del traffico sulle code di spedizione (di host e router).

Soluzioni principali:

- **Code a priorità**: vengono definite diverse classi di priorità e create una coda per classe. I pacchetti in arrivo vengono classificati ed inseriti in una di queste classi. Le code ad alta priorità vengono servite prima; se non ci sono pacchetti in coda si passa alla coda con priorità inferiore.

- **code pesate**: Ad ogni classe viene associato un peso; il numero di pacchetti inoltrati è proporzionale al peso della coda.

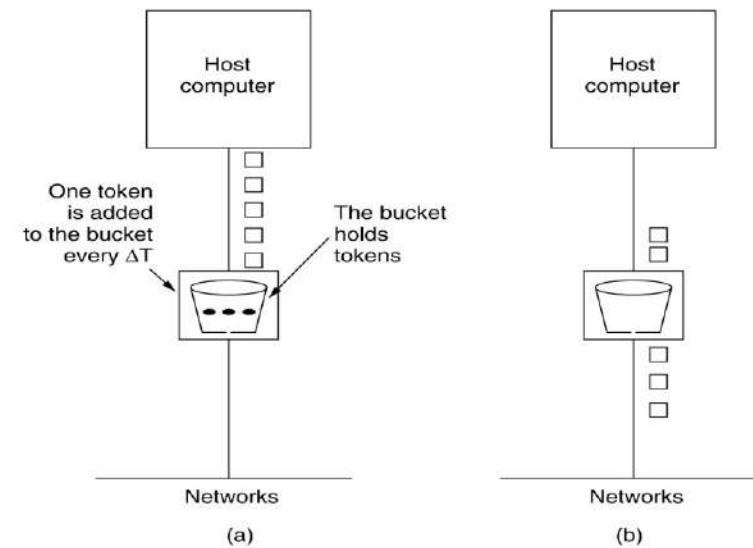
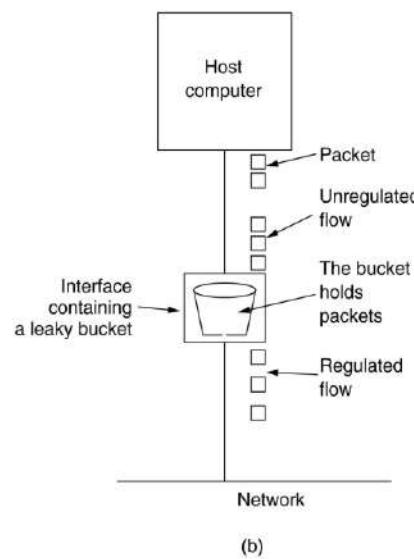
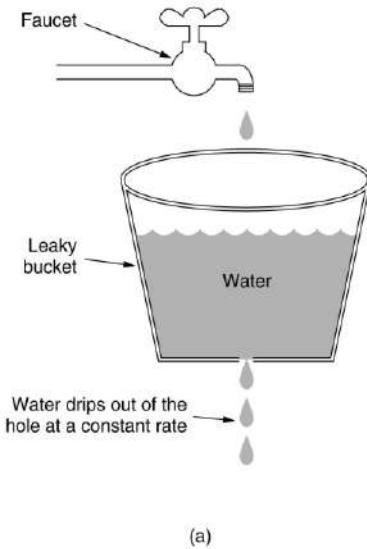
Vantaggio: le code con meno peso vengono comunque servite.

- **Code a velocità limitata**: Si utilizzano quando si vuole limitare la velocità massima, ad esempio quando al mittente è stato offerto un determinato servizio.

Leaky Bucket e Token Bucket sono algoritmi utilizzati per la limitazione della velocità.

Leaky Bucket e Token Bucket

- ▶ **Leaky Bucket (imbuto)** : I dati da spedire possono arrivare a qualsiasi velocità, ma vengono accodati e rispediti ad un tasso costante e limitato.
- ▶ **Token Bucket**: E' più flessibile grazie ai Token. Un Token rappresenta il diritto a spedire a spedire un pacchetto. I Token vengono forniti al trasmettitore a intervalli regolari di tempo.





UNIVERSITÀ
DI PARMA

DIPARTIMENTO DI SCIENZE MATEMATICHE, FISICHE ED INFORMATICHE
Corso di Laurea in Informatica

II Livello Applicativo – Parte A

Applicativi UDP: TFTP e DNS

RETI DI CALCOLATORI - a.a. 2023/2024

Roberto Alfieri

Livello Applicativo: sommario

PARTE A

- ▶ Applicativi UDP: TFTP e DNS

PARTE B

- ▶ I servizi di posta elettronica: SMTP, POP e IMAP.

PARTE C

- ▶ Il World Wide Web

PARTE D

- ▶ Multimedia

TFTP e FTP

TFTP è la versione semplificata (Trivial) di FTP (File Transfer Protocol)

<http://openskill.info/topic.php?ID=87>

Per noi è interessante perché è un esempio di come viene gestito a livello applicativo il controllo del flusso.

Il protocollo FTP (RFC 959) è stato sviluppato per trasferimento affidabile ed efficiente dei dati, per questo motivo si basa TCP. Il server FTP offre anche un servizio di autenticazione per l'accesso al file-system, la gestione delle directory (navigazione, creazione, cancellazione) e dei file.

Altra caratteristica peculiare è quella di usare due porte: la TCP/21 (comandi) e la TCP/20 (dati). Le modalità di funzionamento sono due: attiva e passiva.

Nella modalità attiva il client apre il canale comandi verso il server (porta 21 del server), mentre per la trasmissione dati il client svolge la funzione di server, ovvero rimane in ascolto sulla porta > 1024 mentre il server si comporta da client utilizzando la porta 20.

In genere le politiche di sicurezza impediscono l'accesso alle porte dei client bloccando questa modalità.

Nella modalità passiva il server indica al client la porta > 1024 da utilizzare per il trasferimento dei dati.

TFTP non supporta l'autenticazione e utilizza UDP, con il server in ascolto sulla porta 69. Questo significa che la gestione del flusso (numerazione dei pacchetti, Ack, gestione degli errori) viene realizzata a livello applicativo, all'interno di TFTP.

Il protocollo TFTP

Ogni trasferimento inizia con una richiesta di read (comando get) o write (comando put).

GET: Il server risponde con un file frammentato in datagrammi numerati.

Ogni datagramma deve essere riscontrato.

Il block-size di default è di 512 byte

Un pacchetto di dimensione inferiore rappresenta l'ultimo pacchetto trasmesso

Opcode (16 bit)

1 RRQ (Read Request)

2 WRQ (Write Request)

3 DATA

4 ACK

5 ERROR

2 bytes	string	1 byte	string	1 byte
Opcode	Filename	0	Mode	0

RRQ/WRQ Packet

2 bytes	2 bytes	up to 512 bytes of data
Opcode	Block#	Data

DATA Packet

2 bytes	2 bytes
Opcode	Block#

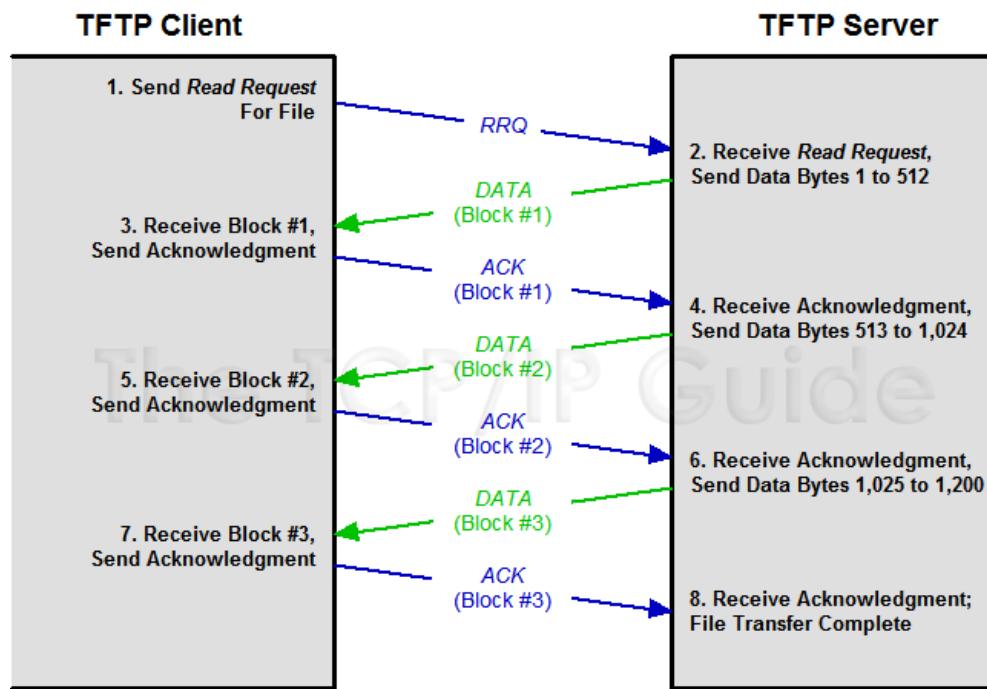
ACK Packet

2 bytes	2 bytes	string	1 byte
Opcode	Block#	ErrMsg	0

ERROR Packet

Le fasi di una sessione TFTP

- Il client contatta il server inviando un pacchetto di tipo RRQ o WRQ
- Il server risponde inviando/ricevendo pacchetti DATA di 512 byte.
Per ogni DATA inviato/ricevuto viene inviato/ricevuto un ACK (o un ERROR)
- I pacchetti vengono trasferiti finché la loro lunghezza non è inferiore a 512 byte;
- Termine della connessione;



Esempi:

```
> tftp -v <nome server> -c put /etc/hosts hosts  
> tftp -v <nome server> -c get hosts
```

DNS: Domain Name System

DNS è il protocollo applicativo più importante tra quelli che si appoggiano su UDP.

Scopo del sistema DNS:

- ▶ gestire uno spazio univoco dei nomi per i nodi della rete
- ▶ Fornire agli utenti di IP un servizio per la traduzione nome-numero e numero-nome

Lo spazio dei nomi è strutturato in modo gerarchico, come il file-system, ma la radice è a destra. *Esempio: didattica-linux.unipr.it.*

Il primo elemento è il nome locale del nodo, mentre gli elementi successivi (domini) rappresentano il percorso nella gerarchia e sono separati da punto ('.').

Un nome completo termina sempre con un punto, che rappresenta la radice della gerarchia.

Il dominio più a destra (**it** nell'esempio) è detto Top Level Domain (TLD).

I TLD sono gestiti dall'organismo internazionale [ICANN](#) (attraverso la sua emanazione [IANA](#)) che li assegna alle organizzazioni che ne fanno richiesta, mentre i livelli successivi sono gestiti in modo autonomo dalle organizzazioni assegnatarie.

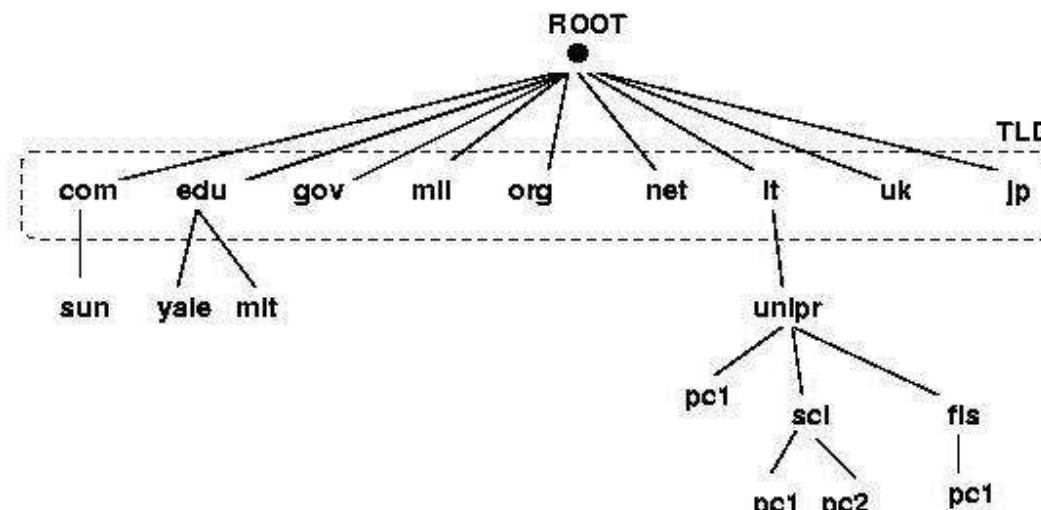
DNS: i Top Level Domain

Inizialmente Internet era composta unicamente da nodi americani, per cui esistono alcuni TLD che rispecchiano la strutturazione Statunitense originale (1985): **com (commerciali), edu (istituzioni educative), gov (governo federale US), mil (forze armate US), net (provider di rete), org (organizzazioni no-profit)**.

Successivamente, con la diffusione di Internet, sono nati i TLD geografici nazionali: **it, es, fr, de, gr ca, at, au, be, nl, pt, ch, ecc**

A partire dal 2000 ICANN ha approvato diversi nuovi nomi TLD generici quali: **biz (business), info (informazioni), name (nome di persona), pro (professionisti), coop (cooperative), museum (musei), travel (viaggi), aero (aerotrasporti)**

Attualmente sono attivi seguenti TLD <http://www.iana.org/domains/root/db/>



Risoluzione Inversa

La risoluzione inversa consiste nella risoluzione del nome a partire dall'indirizzo IP. Viene usata ad esempio per produrre un output leggibile nei file di log, oppure per controlli di autenticazione (e.g. richiedere che il client sia registrato). Il nome dei domini di Reverse è composto dai numeri della rete (in ordine rovesciato), seguiti dalla stringa “in-addr.arpa” (TLD per la risoluzione inversa). Il rovesciamento del numero consente di ricercare i numeri nello stesso albero dei nomi, utilizzando lo stesso procedimento di parsing da destra verso sinistra.

Ad esempio:

10.48.78.160.in-addr.arpa

è il nome di

caio.cce.unipr.it

nel ramo della risoluzione inversa.

provare il comando:

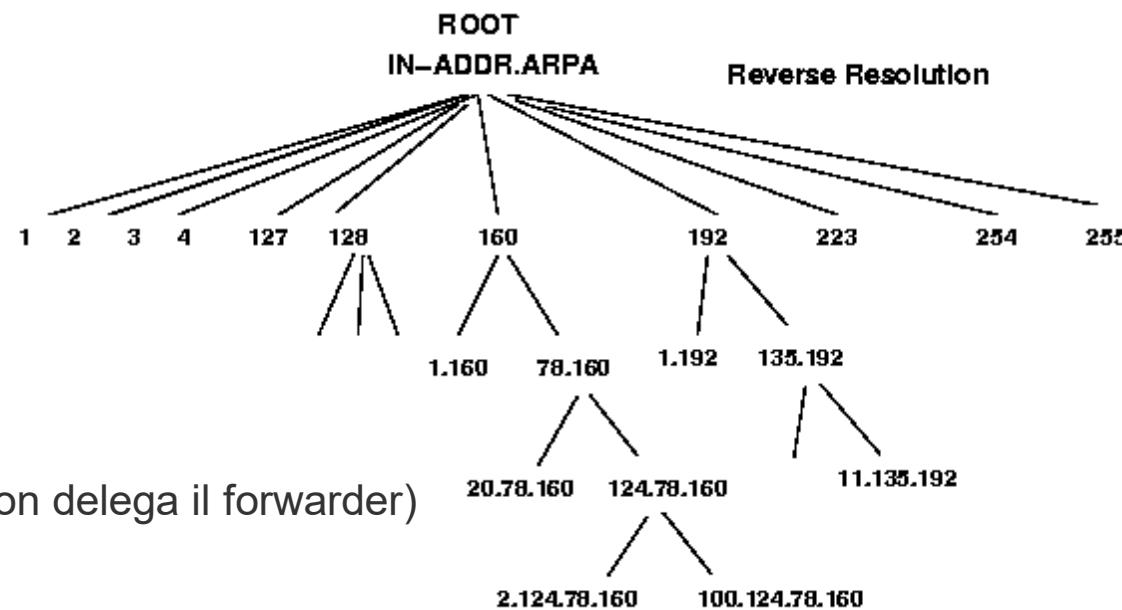
dig PTR 10.48.78.160.in-addr.arpa

dig -x 160.78.48.10

dig -x 160.78.48.10 +trace

-x → semplifica il reverse search

+trace → dig esegue query iterative (non delega il forwarder)



DNS client

Un DNS client contiene un componente software, detto Resolver, che ha il compito di risolvere la richiesta. Il Resolver deve essere configurato con l'indicazione di almeno un **server DNS forwarder** a cui rivolgersi per le risoluzioni.

Il server DNS forwarder recupera l'informazione (interagendo eventualmente con gli altri server DNS) e la comunica al Resolver.

Nei sistemi Linux questa configurazione viene stabilita nel file `/etc/resolv.conf`.

Esempio:

`nameserver 160.78.48.10`

la configurazione del DNS può essere impostata dinamicamente dal protocollo DHCP assieme alle altre informazioni di configurazione (indirizzo IP, Gateway, netmask, ..)

Vista la criticità del servizio DNS un client dichiara tipicamente almeno 2 DNS Forwarder per ridondanza.

In un programma applicativo il ruolo di Resolver è svolto dalla funzione **gethostbyname()**.

Esistono anche specifici client a linea di comando come **nslookup** e **dig**

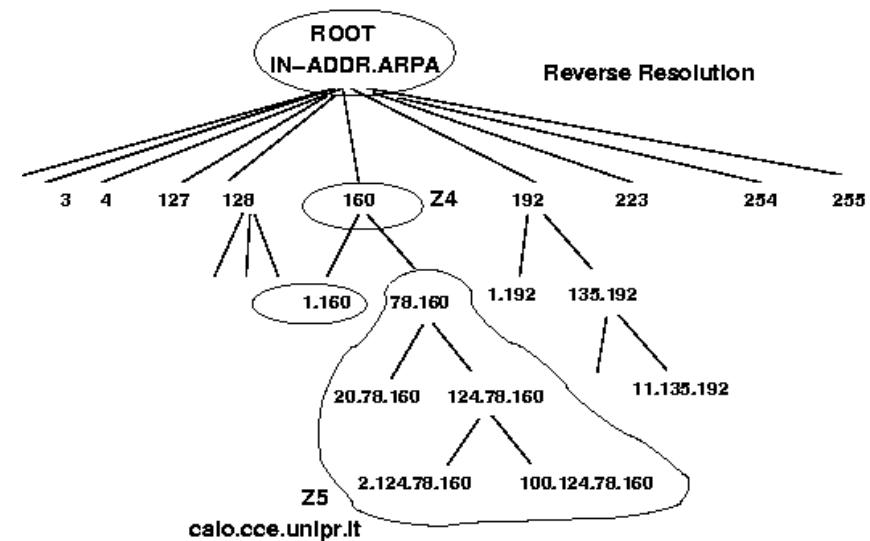
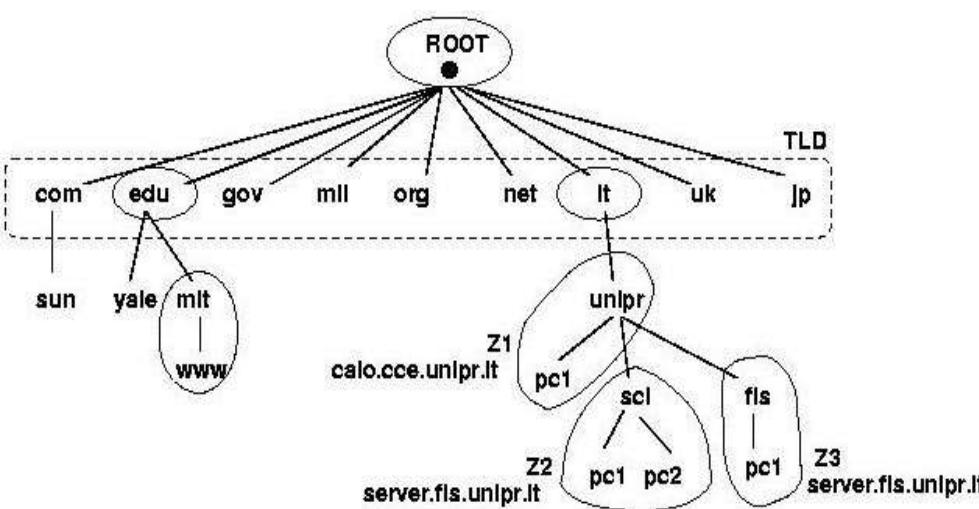
Le Zone del DNS

I TLD possono creare sottodomini di livello 2, i quali a loro volta possono gestire domini di livello 3 e così via. *Ad esempio il dominio a.b.com. può creare c.a.b.com*

Lo spazio dei nomi è gestito in modo distribuito suddividendolo in “Zone”. Ogni Zona include una porzione dell'albero che gestisce in modo autonomo con un DNS server Primario e uno o più DNS server Secondari che ne replicano i dati (per sicurezza e prestazioni).

Una Zona si forma attraverso la delega che il server della Zona superiore assegna mediante il record NS

Ogni nuovo host viene inserito tipicamente sia nella zona per la risoluzione diretta che nella zona per la risoluzione inversa.



Il servizio DNS: il server

Il **servizio DNS** è definito dall'RFC1034 e RFC1035

BIND è il nome del demone DNS più comunemente usato sui sistemi Unix/Linux.

Un server può essere configurato per diversi tipi di funzionamento:

- ▶ **Server Autoritativo di Zona Primario o Secondario**

L'amministratore di Zona aggiorna i dati sul Primario.

I server Secondari si sincronizzano con il primario replicando tutta la Zona (“Zone Transfer”) attraverso un Data-Pull sulla porta 53/TCP.

Risponde alle query che riceve (53/UDP).

Un server Autoritativo è generalmente anche Forwarder NS.

- ▶ **Server Forwarder (o Caching Name Server):**

Riceve query dai client (53/UDP)

Ottiene la risposta interrogando i Server Autoritativi

Mantiene copia locale in Cache delle risposte ottenute

Invia la risposta al client

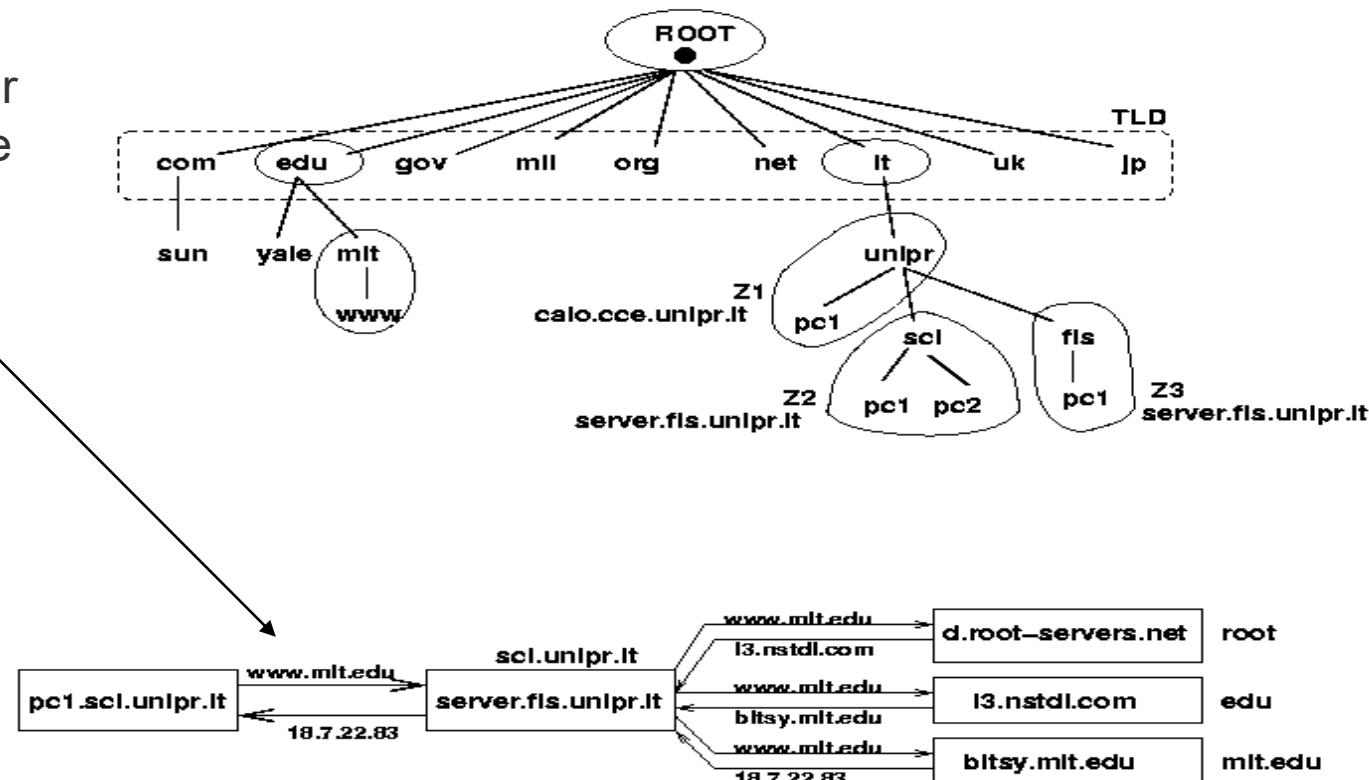
La risoluzione Ricorsiva e Iterativa

Se il server che riceve a richiesta è autoritativo per il dato richiesto risponde direttamente, altrimenti occorre attraversare l'albero passando attraverso i server autoritativi coinvolti. L'attraversamento può essere ricorsivo o iterativo.

Modalità Ricorsiva: Se il server interrogato non è autoritativo per il dato richiesto, passa la richiesta al server successivo e così via in modo ricorsivo.

Modalità Iterativa: Il server restituisce al client l'indirizzo del server successivo. In questo modo è il DNS locale che contatta direttamente i server coinvolti.

Generalmente un server ammette query ricorsive solo per i client locali.



I Root Server

L'albero DNS possiede 13 **Root Server** che contengono le informazioni riguardo i domini di primo livello. Questi server vengono contattati ogni volta che un client chiede informazioni relative ad altri TLD, quindi ogni DNS Forwarder deve possedere una lista aggiornata dei Root Server (file named.ca di Bind).
I Root NS sono iterativi: se non hanno la risposta forniscono l'indirizzo del server del TLD da contattare.

A.ROOT-SERVERS.NET.	3600000 IN	A	198.41.0.4
B.ROOT-SERVERS.NET.	3600000 IN	A	192.228.79.201
C.ROOT-SERVERS.NET.	3600000 IN	A	192.33.4.12
D.ROOT-SERVERS.NET.	3600000 IN	A	128.8.10.90
E.ROOT-SERVERS.NET.	3600000 IN	A	192.203.230.10
F.ROOT-SERVERS.NET.	3600000 IN	A	192.5.5.241
G.ROOT-SERVERS.NET.	3600000 IN	A	192.112.36.4
H.ROOT-SERVERS.NET.	3600000 IN	A	128.63.2.53
I.ROOT-SERVERS.NET.	3600000 IN	A	192.36.148.17
J.ROOT-SERVERS.NET.	3600000 IN	A	192.58.128.30
K.ROOT-SERVERS.NET.	3600000 IN	A	193.0.14.129
L.ROOT-SERVERS.NET.	3600000 IN	A	199.7.83.42
M.ROOT-SERVERS.NET.	3600000 IN	A	202.12.27.33



Resource Record

Le informazioni relative alla zona vengono memorizzate come **Resource Record (RR)**.

Il formato generico di un RR dispone dei seguenti campi:

- **Nome**: Il nome di dominio a cui questo RR si riferisce.
- **TTL**: Tempo di vita del RR nella cache dei server DNS prima di essere scartato.
- **Classe**: Identifica la famiglia di protocollo. **IN** che indica il sistema Internet.
- **Tipo**: Il tipo di RR. I tipi principali sono:
 - ✓ **A**: Il più usato. Indica l'indirizzo IPv4 per il nome specificato.
 - ✓ **AAAA**: Indica l'indirizzo IPv6 (eventuale) associato al nome.
 - ✓ **CNAME**: Record Canonical Name, usato per indicare un nome di alias.
 - ✓ **MX**: Mail eXchanger, indica un host che gestisce la posta per il dominio.
 - ✓ **NS**: Un server DNS per il dominio specificato.
 - ✓ **PTR**: Usato nella risoluzione inversa per associare un indirizzo IP al nome.
 - ✓ **SOA**: Start of Authority, un RR che indica il server DNS dove risiedono i dati autoritativi per questo dominio ed alcuni dati amministrativi.

Esempio di Record

```
; Authoritative data for unipr.it. (Zone1)
;nome      classe    tipo     valore          (serial  refresh  retry  expire  min_life)
unipr.it   IN        SOA      caio.cce.unipr.it  manager (20050818 8H 2H 1W 1D)
unipr.it   IN        TXT      "Univ Parma"
unipr.it   IN        MX       1           mail.unipr.it.
unipr.it   IN        MX       2           mail2.unipr.it.
fis        IN        NS       server.fis.unipr.it. ; Zone 2

dns        IN        CNAME    server1.unipr.it.
mail       IN        CNAME    server2.unipr.it.
mail2      IN        CNAME    server1.unipr.it.
pop        IN        CNAME    server2.unipr.it.
www        IN        CNAME    server1.unipr.it.

server1    IN        A        160.78.124.1
            IN        HINFO    Server1 Linux

server2    IN        A        160.78.124.2
            IN        HINFO    Server2 Linux

pc1        IN        A        160.78.124.101
            IN        HINFO    Client1 Win2000

pc2        IN        A        160.78.124.102
            IN        HINFO    Client2 Win2000
```

DNS e la Posta elettronica

Per ogni dominio che è in grado di ricevere posta il DNS fornisce una lista di server SMTP a cui inviare il messaggio.

In questo modo, se il mail server principale del destinatario non è operativo, è possibile inviare il messaggio verso un server di backup in grado di gestire la posta del dominio.

Queste informazioni sono contenute nei record MX (Mail eXchanger).

unipr.it	IN	MX	1	icaro.cce.unipr.it.
unipr.it	IN	MX	2	ipruniv.cce.unipr.it.

Il campo numerico indica la priorità (il numero più basso ha maggiore priorità)

Provare il comando:

➤ dig -t mx unipr.it +short

0 unipr-it.mail.protection.outlook.com.

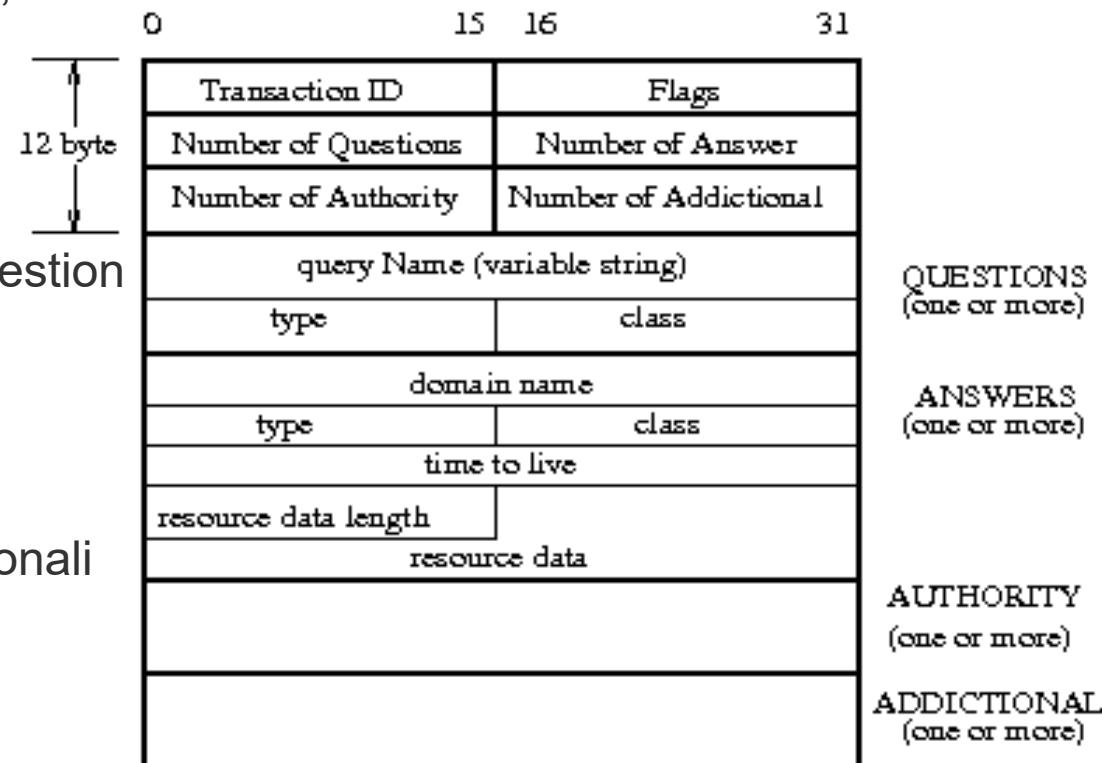
Il formato del pacchetto DNS

Tutti i pacchetti DNS hanno lo stesso formato di figura, composto da:

Transaction ID: serve per associare domanda a risposta

16 Flags tra cui: QR (domanda (0) / risposta (1)) - OPCODE (4 bits, tipo di query)
RD (Recursion Desired) - RA (Recursion Available) - RCODE (4 bits, esito della risposta)

4 sezioni: QUESTIONS, ANSWERS, AUTHORITY e ADDITIONALS



DNS dinamico (dDNS)

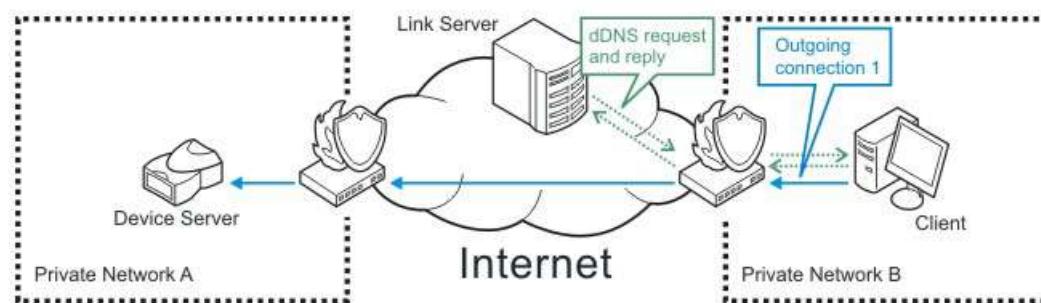
Il DNS Dinamico è una tecnologia che permette ad un nome DNS in Internet di essere sempre associato all'indirizzo IP di uno stesso host, anche se l'indirizzo cambia nel tempo (tipicamente computer portatili).

Il servizio è costituito da una popolazione di client dinamici (host con indirizzo IP dinamico che vogliono che il loro IP attuale sia registrato nel DNS), da uno o più server DNS dinamici e da un protocollo di comunicazione tra le due parti.

[nsupdate](#) è una utility disponibile ai client per l'aggiornamento del DNS.

L'aggiornamento dinamico del DNS può essere fatto direttamente dal server dhcp:

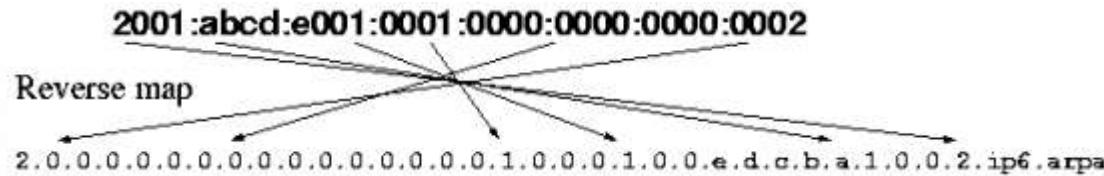
<http://semicomplete.com/articles/dynamic-dns-with-dhcp>



DNS e IPv6

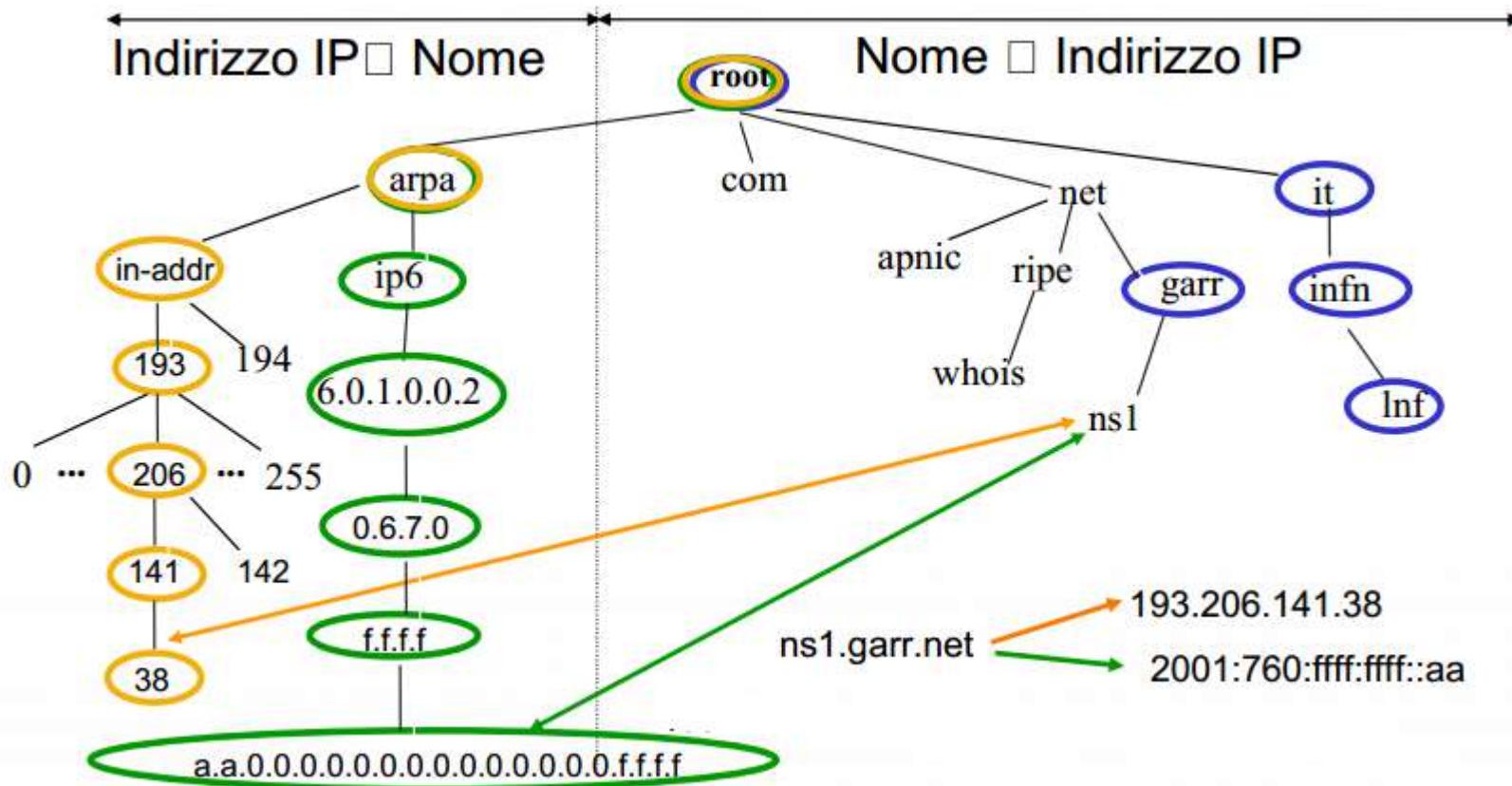
L'utilizzo di IPv6 non modifica i meccanismi di base del DNS, ma è solo stato necessario introdurre alcuni nuovi elementi:

- Un nuovo Resource Record AAAA
 - Un nuovo dominio per la risoluzione inversa : ip6.arpa (RFC 3152)



Riferimenti: Tutorial del GARR http://www.garr.it/eventiGARR/ws9/pdf/Gallo_Valli_pres.pdf

Il name space IPv4 e IPv6



Riferimenti: Tutorial del GARR: http://www.garr.it/eventiGARR/ws9/pdf/Gallo_Valli_pres.pdf

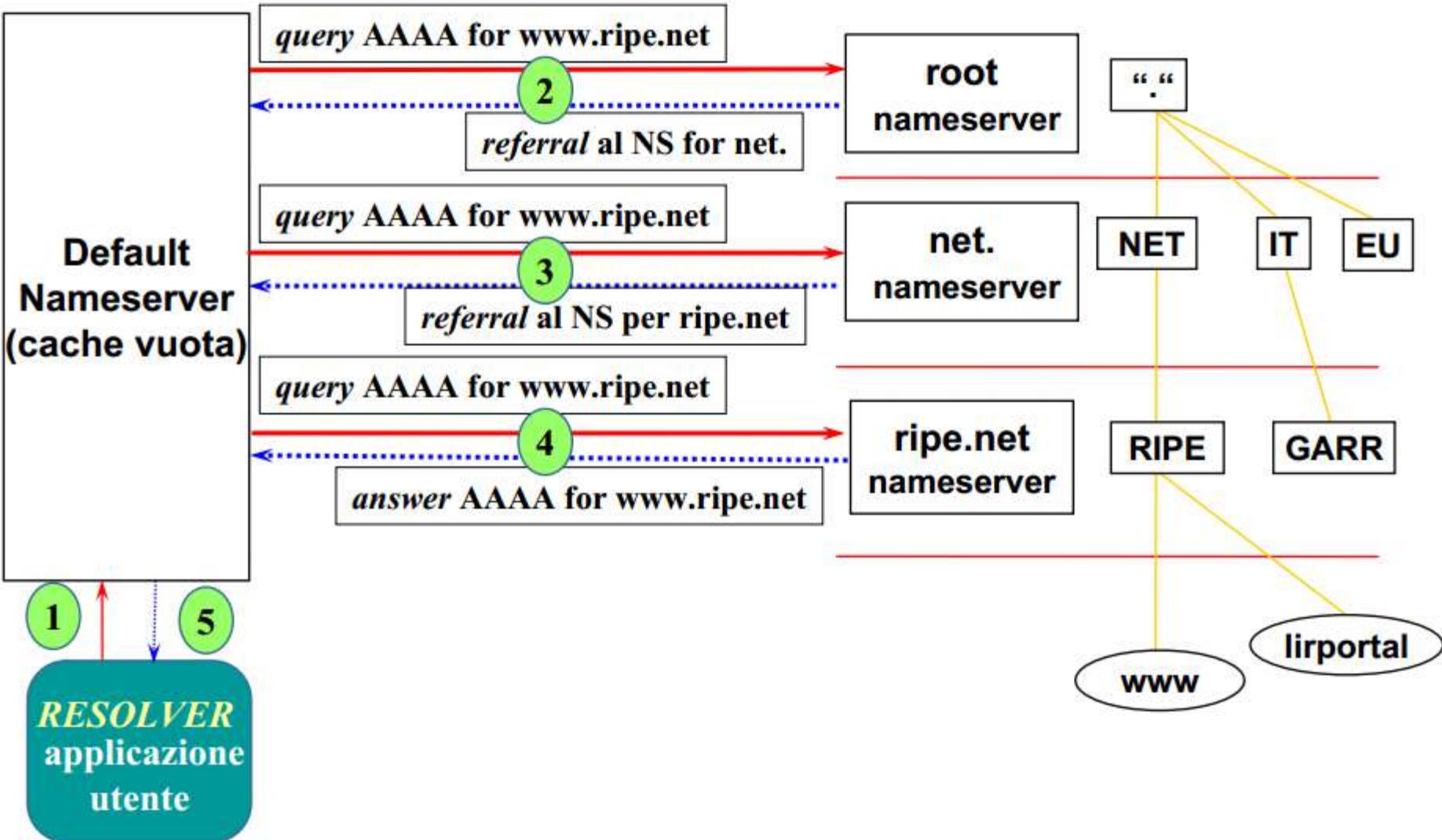
IPv6 e i root NS

Lettera	Vecchio nome	Indirizzo IPv6	Indirizzo IPv4	Operatore	Luogo geografico
A	ns.internic.net	2001:503:ba3e::2:30	198.41.0.4	VeriSign	Dulles, Virginia, USA
B	ns1.isi.edu	2001:500:84::b	192.228.79.201	USC Information Sciences Institute	Marina del Rey, California, USA
C	c.psi.net	2001:500:2::c	192.33.4.12	Cogent	distribuito in anycast
D	terp.umd.edu	2001:500:2d::d	199.7.91.13	University of Maryland	College Park, Maryland, USA
E	ns.nasa.gov	2001:500:a8::e	192.203.230.10	NASA	Mountain View, California, USA
F	ns.isc.org	2001:500:2f::f	192.5.5.241	ISC	distribuito in anycast
G	ns.nic.ddn.mil	2001:500:12::d0d	192.112.36.4	NIC del DoD USA	Vienna, Virginia, USA
H	aos.arl.army.mil	2001:500:1::53	128.63.2.53	U.S. Army Research Lab	Poligono di Aberdeen, Maryland, USA
I	nic.nordu.net	2001:7fe::53	192.36.148.17	Autonomica Archiviato il 1° maggio 2001 in Internet Archive.	distribuito in anycast
J	-	2001:503:c27::2:30	192.58.128.30	VeriSign	distribuito in anycast
K	-	2001:7fd::1	193.0.14.129	RIPE NCC	distribuito in anycast
L	-	2001:500:9f::42	198.32.64.12	ICANN	Los Angeles, California, USA
M	-	2001:dc3::35	202.12.27.33	Progetto WIDE	distribuito in anycast

Al momento non è possibile aggiungere altri nomi di server, a causa di un problema di ottimizzazione del protocollo: un pacchetto UDP deve poter contenere tutti i nomi dei server e con un quattordicesimo nome si supererebbe la dimensione massima del pacchetto.

Riferimenti: Wikipedia https://it.wikipedia.org/wiki/Root_nameserver

Processo iterativo per risoluzione dei nomi



Riferimenti: Tutorial del GARR http://www.garr.it/eventiGARR/ws9/pdf/Gallo_Valli_pres.pdf

Sicurezza del DNS

Le specifiche originali del DNS (RFC 882, 1034 e 1035) non prevedono autenticazione, integrità dei dati o cifratura e espongono il servizio a violazioni della sicurezza. Il principale problema di sicurezza riguarda la possibilità di dirottare la richiesta di traduzione del nome di un server verso un IP fraudolento ([DNS spoofing](#)).

Una tecnica molto diffusa è il [DNS cache poisoning](#) (avvelenamento della cache).

L'attaccante gestisce un server DNS autoritativo per un dominio fake. Quando viene interrogato risponde includendo informazioni false (con un TTL molto grande) riguardo un dominio target, che vengono memorizzate in cache ed utilizzate quando successivamente vengono richieste traduzioni sul target, dirottando la richiesta verso un IP fraudolento.

La soluzione principale è stata l'introduzione del DNSSEC (RFC 9364).

DNS Security Extensions (DNSSEC)

I Domain Name System Security Extensions (DNSSEC) sono una serie di specifiche dell'IETF ([RFC 9364](#)) per garantire la sicurezza e affidabilità delle informazioni fornite dai sistemi DNS.

Servizi:

- Autenticazione: garanzia sull'origine dei dati DNS
- Integrità dei dati ricevuti (non la riservatezza)

Funzionamento:

Ogni server DNSSEC possiede una coppia di chiavi crittografiche, una pubblica e una privata. La chiave privata viene utilizzata per firmare ogni Resource Record (RRset) generando un nuovo Record type, il RRsig. Il server pubblica anche la chiave pubblica introducendo il nuovo record type DNSkey.

Il client, utilizzando la chiave pubblica DNSkey del server può verificare l'autenticità della firma RRsig.

Riferimenti:

<https://www.cloudflare.com/dns/dnssec/how-dnssec-works/>





UNIVERSITÀ
DI PARMA

DIPARTIMENTO DI SCIENZE MATEMATICHE, FISICHE ED INFORMATICHE
Corso di Laurea in Informatica

Il Livello Applicativo – Parte B

La posta elettronica

RETI DI CALCOLATORI - a.a. 2023/2024

Roberto Alfieri

Livello Applicativo: sommario

PARTE A

- ▶ Applicativi UDP: TFTP e DNS

PARTE B

- ▶ I servizi di posta elettronica: SMTP, POP e IMAP.

PARTE C

- ▶ Il World Wide Web

PARTE D

- ▶ Multimedia

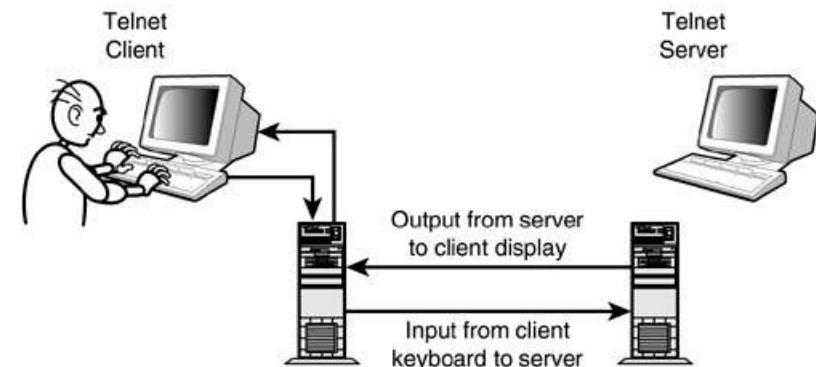
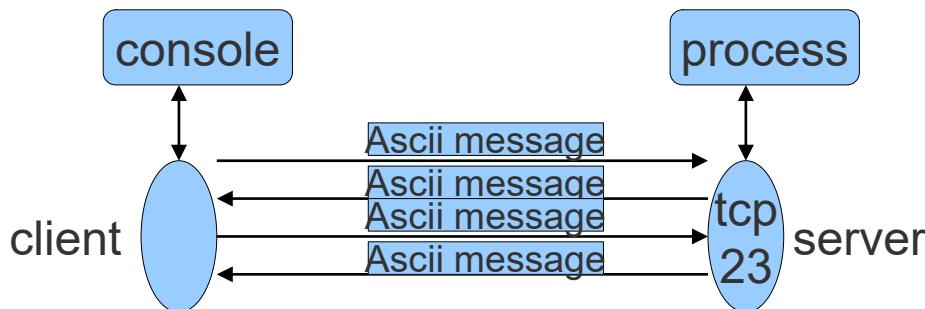
Il protocollo Telnet: emulazione di terminale remoto

Il telnet è un protocollo storico di TCP/IP (RFC 854) creato per l'accesso remoto alla console testuale di un host, basandosi su una connessione TCP tra client e server.

La porta assegnata da IANA è la 23/TCP. Operazioni:

- **Client.** Il client legge lo stream dallo standard input (tastiera) e lo gira al server. Lo stream proveniente dal server viene inviato allo standard output (video).

- **Server.** Il server legge le linee provenienti dal client, le interpreta come comandi di console e invia al client l'output del comando. Il dialogo inizia con la richiesta delle credenziali (username/password).



Telnet è un protocollo testuale (richieste e risposte sono spedite in ASCII, compresa la password) ha quindi un livello di sicurezza molto basso ed è stato sostituito da protocolli più sicuri (SSH).

Viene comunque ancora utilizzato per casi speciali (console di apparati di rete) o come semplice strumento di analisi di altri protocolli testuali (SMTP, POP3, IMAP, HTTP, ...)

La Posta Elettronica

Definita nel 1982 con gli RFC821, RFC822 ed estesa con RFC2821 e RFC2822
Ogni utente possiede una MailBox in cui gli altri utenti possono liberamente inserire messaggi.

Nel 2015 lo spam rappresentava il 55% del flusso mondiale di mail
(<https://securelist.com/kaspersky-security-bulletin-spam-and-phishing-in-2015/73591/>)

Un indirizzo di posta è costituito da due identificativi: nome del server che gestisce la mailbox e identificativo dell'utente: <utente>@<server>

Dove <server> è l'indirizzo IP o un suo identificativo nel DNS.

Il record MX del DNS consente di creare caselle di posta in un dominio, a cui possiamo associare uno o più server di posta per la sua gestione.

Ad esempio il seguente record DNS (ottenuto con dig -t MX unipr.it):
unipr.it. 86400 IN MX 0 unipr-it.mail.protection.outlook.com.

Consente di creare indirizzi di posta del tipo <utente>@unipr.it
Le cui caselle verranno gestite dal server unipr-it.mail.protection.outlook.com.

La Posta Elettronica : l'architettura

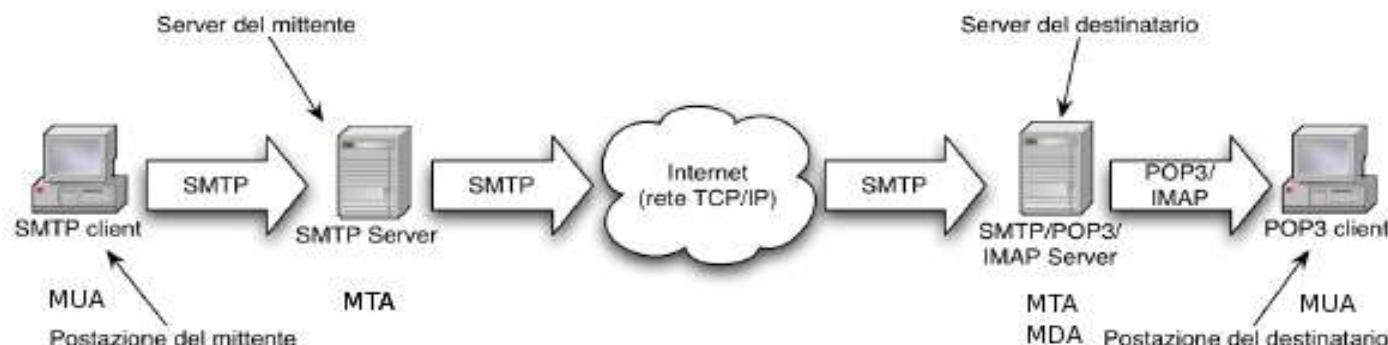
Ogni utente ha bisogno di uno MUA - “Message User Agent”, un programma che gli consente di inviare messaggi o leggere i messaggi dalla propria Mailbox.

Il MUA consegna il messaggio ad un MTA - “Message Transfer Agent” che ha il compito di trasportare il messaggio a destinazione. Il messaggio può attraversare diversi MTA prima di arrivare nella MailBox del destinatario.

Il primo MTA è detto anche Submission server perché è utilizzato dai MUA per la sottomissione delle mail e dovrebbe supportare autenticazione (SMTP AUTH).

Sull'ultimo MTA è presente anche l'MDA - “Message Delivery Agent” che si occupa della consegna del messaggio nella MailBox dell'utente.

Se il MUA del destinatario e MDA sono su host diversi occorre un protocollo per la lettura dei messaggi (POP3 o IMAP)



Il formato dei messaggi

Il formato dei messaggi è stato codificato nel 1982 attraverso l'RFC 822, superato nel 2001 dall'RFC 2822 e poi nel 2008 dall'RFC 5322.

I messaggi sono codificati in ASCII standard (7bit) in righe di max.1000 caratteri

Formato:

Intestazione	(Header)
<cr><lf>	(riga vuota)
corpo del messaggio	(Body)

Intestazione: in ogni riga la coppia <campo>:<valore>

Principali campi:

- ▶ To: indirizzi dei destinatari
- ▶ Cc: indirizzi destinatari secondari
- ▶ Bcc: destinatari secondari con indirizzi nascosti
- ▶ From: La persona che ha creato il messaggio (obbligatorio)
- ▶ Received: Riga aggiunta da ogni MTA attraversato
- ▶ Date: data e ora di invio del messaggio
- ▶ Subject: Breve riepilogo del messaggio
- ▶ Message-ID: Identificatore unico del messaggio (generato automaticamente).
- ▶ User-Agent: Client di posta utilizzato dall'utente

Esempio messaggio

MTA
Attraversati

Received: from smtp.unipr.it (sud.cce.unipr.it [160.78.48.162])

by unipr.it (8.12.6/8.12.6) with ESMTP id j4CCXorj032665

for <destinatario@unipr.it>; Thu, 12 May 2005 14:33:50 +0200



Received: from smtp2.mathworks.com (smtp2.mathworks.com [144.212.95.218])

by smtp.unipr.it (8.12.10/8.12.10/SuSE Linux 0.7) with ESMTP id j4CCTeOW001983

for <destinatario@unipr.it>; Thu, 12 May 2005 14:29:45 +0200

Received: from mail-vif.mathworks.com (fred-ce0.mathworks.com [144.212.95.18])

by smtp2.mathworks.com (8.12.11/8.12.11) with ESMTP id j4CCVPR5009740

for <destinatario@unipr.it>; Thu, 12 May 2005 08:31:25 -0400 (EDT)

Received: from telesto.mathworks.com (telesto.mathworks.com [144.212.95.234])

by mail-vif.mathworks.com (8.11.7/8.11.7) with ESMTP id j4CCVOH06340

for <destinatario@unipr.it>; Thu, 12 May 2005 08:31:25 -0400 (EDT)

Message-ID: <74127410.1115901084851.JavaMail.wwwadmin@telesto.mathworks.com>

Date: Thu, 12 May 2005 08:31:24 -0400 (EDT) #Nota: EDT US East Coast

From: sender@mathworks.com

User-Agent: Mozilla Thunderbird 1.0.2 (Windows/20050317)

Reply-To: sender@mathworks.com

To: destinatario@unipr.it

Subject: Greetings

Greetings from Mathworks

bye

MIME

Il formato RFC822 del 1982 era stato pensato per messaggi in cui corpo era esclusivamente testo espressi in ASCII standard.

Questo schema non ammette lettere accentate, alfabeti non latini, messaggi multimediali ecc.

La soluzione è stata proposta da MIME RFC1341, oggi ampiamente utilizzata in Internet.

MIME (Multipurpose Internet Mail Extensions) introduce 5 nuove intestazioni:

- 1) **MIME-version**
- 2) **Content-description**
- 3) **Content-id**
- 4) **Content-transfer-encoding**. Il nome dello schema di codifica utilizzato per trasformare il messaggio in ASCII standard. Principali valori:

Ascii 7bit: nessuna codifica. Linee fino a 1000 caratteri

Ascii 8bit: viola la versione originale del protocollo, ma probabilmente funziona.

Quoted-printable: messaggi ASCII non standard. I caratteri superiori al 127 sono codificati con = seguito dal codice ASCII in esadecimale. (città -> citt=9A)

base64: Per dati binari. Ogni sequenza di 6 bit viene trasformato in un carattere ASCII grazie ad una codifica di 64 simboli (sprecati 2 bit ogni 6, i dati codificati occupano il 35% in più)

Esempio di codifica di una immagine: `openssl base64 -e -in immagine.png -out immagine.b64`

Esempio di decodifica: `openssl base64 -d -in immagine.b64 -out immagine.png`

5) Content-type. Natura del corpo del messaggio

Espresso nella forma type/subtype (esempio Content-Type: text/plain)

Utile per attivare automaticamente il Viewer corretto (esempio video/mpeg)

Tipi e sottotipi sono definiti da IANA - <http://www.iana.org/assignments/media-types/index.html>

Principali Content-types

Type	Subtype	Description
Text	Plain	Unformatted text
	Enriched	Text including simple formatting commands
Image	Gif	Still picture in GIF format
	Jpeg	Still picture in JPEG format
Audio	Basic	Audible sound
Video	Mpeg	Movie in MPEG format
Application	Octet-stream	An uninterpreted byte sequence
	Postscript	A printable document in PostScript
Message	Rfc822	A MIME RFC 822 message
	Partial	Message has been split for transmission
	External-body	Message itself must be fetched over the net
Multipart	Mixed	Independent parts in the specified order
	Alternative	Same message in different formats
	Parallel	Parts must be viewed simultaneously
	Digest	Each part is a complete RFC 822 message

Esempio messaggio con MIME

Received: from ...

Message-ID: ...

Date: ...

From: ...

To: ...

Subject: ..

MIME-Version: 1.0

Content-Type: multipart/mixed; boundary="-----ThIs-RaNdOm-StRiNg-/_=.468328521:"

-----ThIs-RaNdOm-StRiNg-/_=.468328521:

Content-Transfer-Encoding: 7bit

Content-Type: text/plain

Ecco l'allegato

ciao

-----ThIs-RaNdOm-StRiNg-/_=.468328521:

Content-Transfer-Encoding: base64

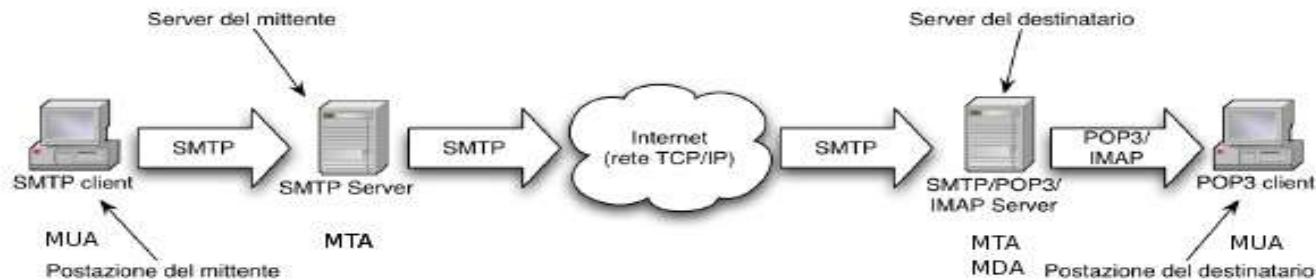
Content-Type: application/msword

BgAAAAAAAGYGAAAAAAAABIAwAAPAEAAAIBAAAAAAAASAMAAAAAAAACAQAAAAAAEgD
AAAAAAA6gkAAAAAAAAAAAAAGYGAaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
AAAAAAAASAMAAAAAAADqCQAAAAAAAGYGAABIwAAZgYAAAAAAAAAAAAA
AAAAAK4JAAAAAAAAGEAAAAAAACAQAAAAAAAASAMAAAAAAADqCQAAAAAAAGYGAABIwAAZgYAAAAAAAAAAAAA
-----ThIs-RaNdOm-StRiNg-/_=.468328521:--

SMTP

SMTP è un protocollo applicativo (25/TCP) che si occupa del trasferimento di un messaggio da MUA a MTA o da MTA a MTA. L'MTA può essere destinatario finale (MTA+MDA) o di trasferimento (Mail Relay, Mail scanner, ...).

Il protocollo è codificato **ASCII Standard** ovvero prevede uno scambio di dati testuali.



Principali **comandi** del client SMTP:

HELO: indirizzi dei destinatari

MAIL FROM: mittente del messaggio

RCPT TO: destinatario del messaggio

DATA: corpo del messaggio

QUIT: fine del messaggio

RSET: reset

HELP: nome del comando

Principali **risposte** del server SMTP:

220: Servizio pronto

250: Comando richiesto completato

251: Utente non locale, il messaggio sarà inoltrato

221: Chiusura canale di trasmissione

421: Servizio non disponibile

500: Errore di sintassi

501: Errore di sintassi nei parametri

554: Transazione fallita

Formato della Mailbox

Esistono 2 formati principali:

Mbox

I messaggi ricevuti sono accodati in un singolo file per ogni utente

Su Unix si trova in /var/spool/mail/utente

Ogni messaggio inizia con una linea

“From sender@domain..”

From sender1@domain1
Messaggio1

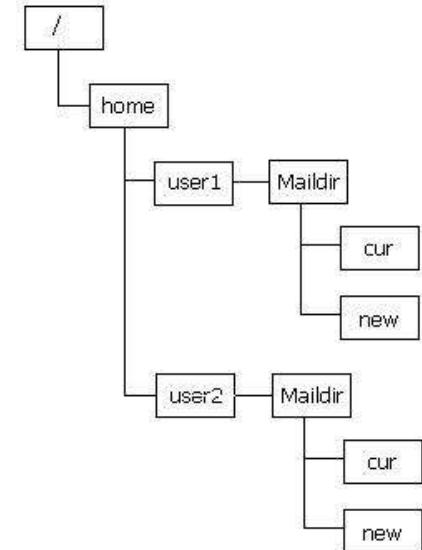
From sender2@domain2
Messaggio2

...

Maildir

Si utilizza una directory per ogni utente

All'interno della directory viene creato un file (di testo) per ogni messaggio ricevuto.



MUA : Mail User Agent

Un MUA è una applicazione che viene usata dall'utente per inviare e ricevere e-mail.

comando «mail»

MUA di base dei sistemi Linux. Non gestisce allegati MIME, POP e IMAP.
E' utile per l'invio automatico di messaggi all'interno di processi.

L'invio è affidato all'MTA su localhost. Esempio di utilizzo:

```
mail -s test user@domain<< EOF  
prova di spedizione  
EOF
```

comando «alpine»

MUA testuale, ma con gestione dello schermo. Software libero per Linux e MacOS.
Gestisce gli allegati MIME, può gestire la Mailbox da remoto con il protocollo IMAP

«Mozilla Thunderbird»

E' un MUA open software con interfaccia grafica disponibile per Windows, Linux e MacOS. Gestisce POP3/ IMAP e un filtro bayesiano anti spam.

Esempio di dialogo SMTP con telnet

telnet localhost 25 ...

Client

```
HELO      client.com  
MAIL FROM: <src@com>  
RCPT TO:   <dest@com>  
DATA
```

```
From: tizio  
To: caio  
Subject: prova
```

```
testo  
testo
```

MESSAGGIO

```
QUIT
```

Server

```
220  srv.com SMTP service ready  
250  client.com OK  
250  sender src@com OK  
250  Recipient dest@com OK
```

```
250  Message Accepted  
221  client.com closing connection
```

Un MTA è un server che accetta nel campo RCPT TO indirizzi diversi dall'host locale.

ESMTP (Extended SMTP)

Visto che con il passare degli anni vengono richieste sempre nuove funzionalità ad SMTP, con l'RFC 1869 del 1995 è stata definita una generale struttura standard di SMTP, denominata ESMTP, in grado di gestire le estensioni presenti e future.

Per utilizzare ESMTP occorre presentarsi con EHLO (anziché HELO). Se EHLO è accettato il server risponde con la lista delle estensioni supportate. Esempio:

```
> EHLO client.com
250-8BITMIME          (8 bit data transmission)
250-SIZE              (Message Size Declaration)
250-DSN               (Delivery Status Notification)
250-AUTH              (SMTP autenticato)
250-STARTTLS          (comunicazione cifrata con StartTLS)
....
```

Le estensioni più interessanti sono AUTH e STARTTLS, che introducono autenticazione e cifratura dei dati in fase di sottomissione del messaggio.

Sicurezza dell' SMTP

Poiché il protocollo SMTP non prevede autenticazione chiunque può contattare un MTA per

- spedire mail verso chiunque (**spam**)
 - Ingannare la vittima per indurlo a rivelare informazioni sensibili o svolgere azioni dannose (**phishing**)
 - furto di identità, ovvero impersonare l'identità di altri (**spoofing**)
 - Sfruttare gli allegati della mail per diffondere malware (virus, trojan, worm, ecc)
- Inoltre, poichè manca anche la riservatezza le mail sono esposte ad **intercettazione**.

Alcune semplici tecniche per mitigare:

- Filtro indirizzi IP sul server SMTP per accettare esclusivamente messaggi in cui il mittente o il destinatario sono locali.
- porte 25 bloccate dal firewall perimetrale per forzare l'utilizzo di MTA istituzionali (ad esempio dotati di antivirus e antispam).
- Verifica della registrazione DNS diretta e inversa del client
- Utilizzo del Resource Record SPF ([Sender Policy Framework](#)) sul DNS per identificare quali sono gli indirizzi IP abilitati a spedire per il nostro dominio.

Sicurezza ESMTP

Sottomissione dei Messaggi con SMTP-AUTH

Extended SMTP è una estensione (RFC 1869) che include nuove funzionalità, tra cui l'autenticazione con SMTP-AUTH per la fase di sottomissione di e-mail attraverso l'introduzione di un MTA dedicato denominato MSA (Message Submission Agent)

Tramite l'estensione SMTP AUTH un MSA può:

- richiedere le credenziali (user/pass o certificati) del client
- cifrare le comunicazioni tra MUA e MSA
- utilizzare una porta di ascolto diversa: 587/tcp

La cifratura SMTP-AUTH riguarda solo la consegna del messaggio all'MSA. Se si vuole una cifratura end-to-end occorre utilizzare una estensione di MIME denominata Secure MIME (S/MIME).

SMTP server

Il server SMTP ha il compito di ricevere e gestire messaggi di posta elettronica.
Opportunamente configurato può svolgere la funzione di

- MSA se riceve messaggi dal MUA offrendo servizi di autenticazione e riservatezza
- MTA se invia il messaggio ricevuto verso altri server SMTP
- MDA se gestisce caselle di posta in cui deposita i messaggi ricevuti

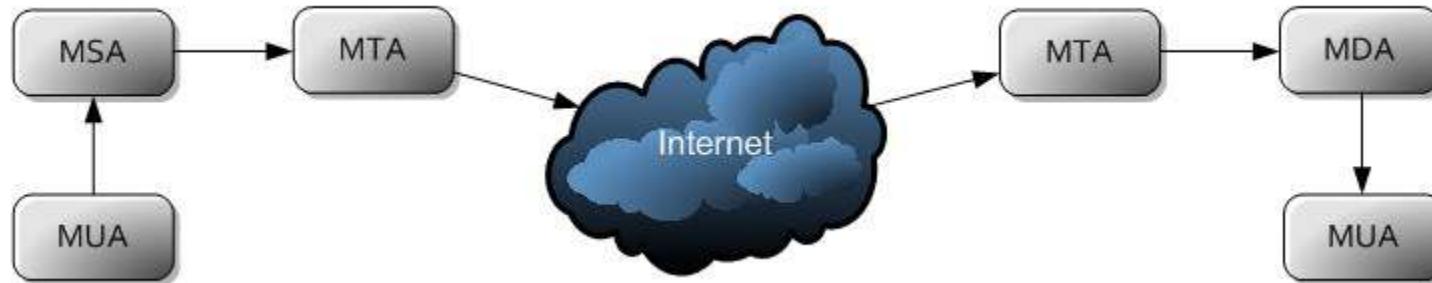
Esistono diverse distribuzioni open source di ESMTP server; le più diffuse sono Sendmail e Postfix.

Lettura dei Messaggi

La consegna del messaggio della mailbox utente è gestita dall'MDA (Message Delivery Agent) che ha anche il compito di consegnare il messaggio al MUA dell'utente previo opportuno meccanismo di autenticazione.

Quando è nato il protocollo SMTP gli utenti lavoravano sulla stessa macchina dove risiedevano le Mailbox, che quindi il MUA poteva accedere direttamente.

Con l'avvento dei PC il MUA si è disaccoppiato dall'MDA ed è nata la necessità di nuovo protocollo di rete per la comunicazione MUA-MDA. Due possibili protocolli: POP3 o IMAP.



POP3

POP3 (Post Office Protocol v.3, RFC 1939) è un protocollo ASCII con autenticazione per il trasferimento dei messaggi dal MailBox allo User Agent utilizzando un servizio TCP sulla porta 110.

Dopo la connessione TCP il protocollo attraversa 3 fasi:

Autenticazione: invio delle credenziali (USER e PASS)

Transazioni: Esecuzione dei comandi (LIST, RETR, DELE, QUIT)

Aggiornamento: Dopo il QUIT il server cancella effettivamente i messaggi eliminati e interrompe la connessione

POP3 è utilizzato tipicamente da Home Users, connessi via modem o ADSL all'ISP, per trasferire (RETR) tutti i nuovi messaggi, che vengono poi eventualmente cancellati dal server (DELE). Il MailBox server funziona così da area di transito per i messaggi, che vengono gestiti sull'Hard Disk dell'utente.

Riferimenti: http://openskill.info/release/guida_ai_protocolli_internet/i_protocolli_pop3_e_imap.htm

Poiché tutte le comunicazioni, incluse le credenziali di autenticazione, avvengono in chiaro, a POP3 è stato affiancato **il protocollo sicuro POP3S** in cui la comunicazione è cifrata grazie all'utilizzo del layer SSL/TLS. POP3S utilizza una porta diversa, la 995/TCP.

Esempio di dialogo POP3 con telnet

telnet localhost 110 ...

Client

USER nomeutente

PASS password

STAT

LIST

TOP 1

TOP 2

RETR 2

DELE 2

QUIT

Server

+OK POP3 ready

+OK

+OK

+OK 2 1000 #(2 mess – 1000 bytes)

+OK

1 500

2 500

.

mostra intestazione messaggio 1

mostra intestazione messaggio 2

recupera messaggio 2

.....

.....

.

+OK Marked to be deleted

+OK Logging Out, message deleted

IMAP

IMAP (Internet Message Access Protocol, RFC 2060), è un protocollo alternativo a POP3 per consentire all'user agent la gestione dei messaggi ricevuti, utilizzando il servizio TCP sulla porta 143.

A differenza di POP3 presume che i messaggi debbano rimanere sul server. Per questo fornisce la possibilità di gestire cartelle di posta sul server in cui archiviare i messaggi ricevuti.

IMAP è adatto per utenti che accedono alla posta utilizzando diversi MUA (casa, lavoro, portatile,...).

Riferimenti: http://openskill.info/release/guida_ai_protocolli_internet/i_protocolli_pop3_e_imap.htm

Poiché tutte le comunicazioni, incluse le credenziali di autenticazione, avvengono in chiaro, a IMAP è stato affiancato il **protocollo sicuro IMAPS** in cui la comunicazione è cifrata grazie all'utilizzo del layer SSL/TLS. IMAPS utilizza una porta diversa, la 993/TCP.

Esempio di dialogo IMAP con telnet

telnet localhost 143 ...

Client

a login <username> <password>
a list "/*" /* * *

Server

OK Dovecot ready
OK Logged in

LIST "/" saved-messages
LIST "/" sent-mail-feb-2018
LIST "/" sent-mail
LIST "/" INBOX
OK List completed.

a examine inbox

OK [PERMANENTFLAGS ()] Read-only mailbox.
6 EXISTS
0 RECENT
OK [UNSEEN 2] First unseen.
OK [UIDVALIDITY 1520015846] UIDs valid
OK [UIDNEXT 7] Predicted next UID
OK [READ-ONLY] Examine completed (0.004 secs).

a logout

BYE Logging out
OK Logout completed.
Connection closed by foreign host.



UNIVERSITÀ
DI PARMA

DIPARTIMENTO DI SCIENZE MATEMATICHE, FISICHE ED INFORMATICHE
Corso di Laurea in Informatica

Il Livello Applicativo – Parte C

WWW

RETI DI CALCOLATORI - a.a. 2023/2024

Roberto Alfieri

Livello Applicativo: sommario

PARTE A

- ▶ Applicativi UDP: TFTP e DNS

PARTE B

- ▶ I servizi di posta elettronica: SMTP, POP e IMAP.

PARTE C

- ▶ Il World Wide Web

PARTE D

- ▶ Multimedia

World Wide Web

WWW (World Wide Web) è una architettura client/server per la consultazione di documenti multimediali e ipertestuali distribuiti in rete.

L'architettura è nata nel 1989 al CERN di Ginevra e dal 1994 il suo sviluppo è gestito dal consorzio **W3C** (accordo CERN-MIT).

HTML (HyperText Markup Language) è il formato con cui vengono descritti gli ipertesti.

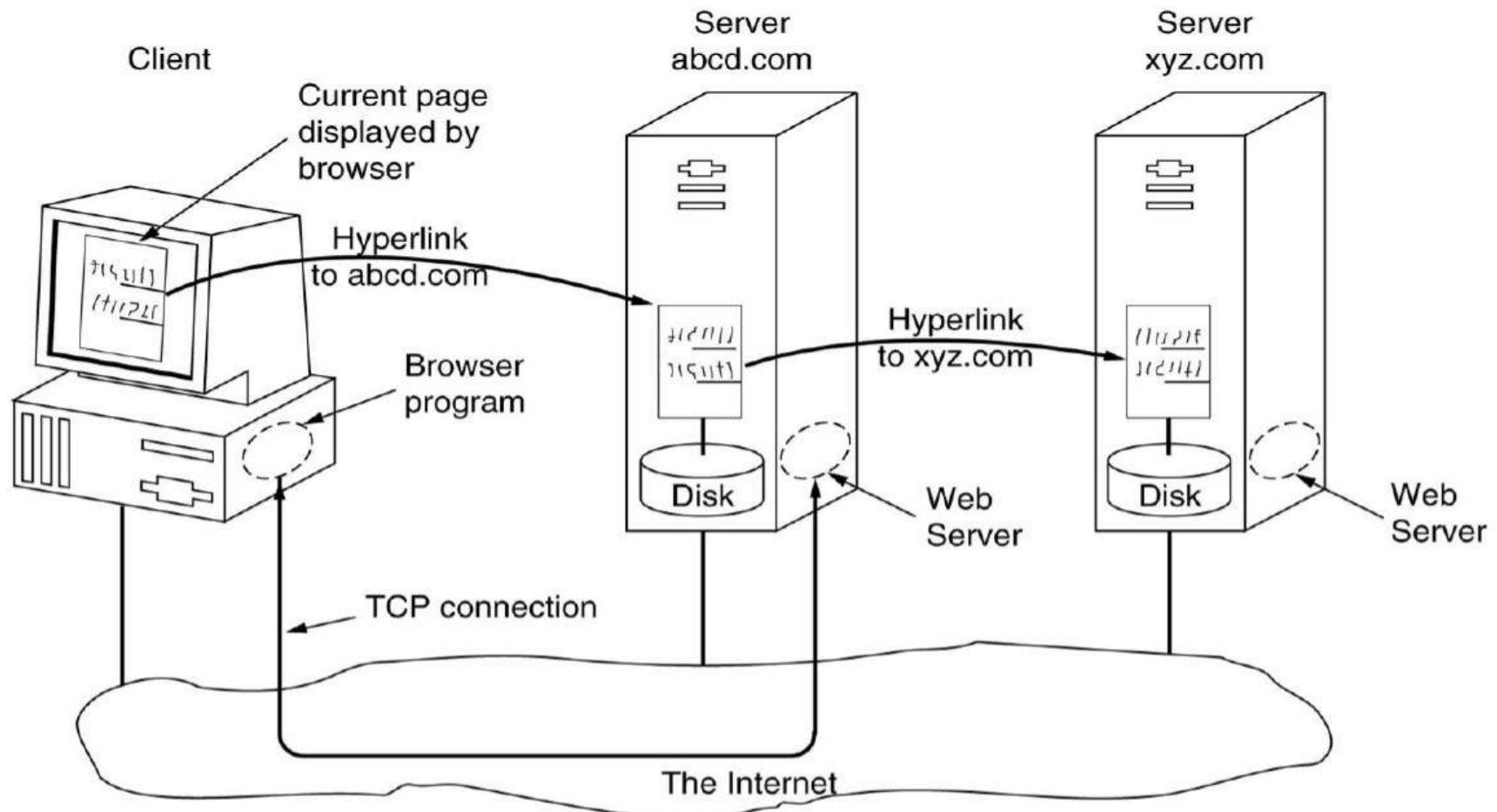
HTTP (HyperText Transfer Protocol) è il protocollo principale per la comunicazione tra **client** e **server**, anche se l'architettura WWW consente l'utilizzo di protocolli diversi.

Ogni documento WWW o singolo oggetto multimediale (file audio, immagine, ..) è identificato mediante un indirizzo univoco in Internet (**URL**) e può contenere riferimenti ipertestuali ad altri documenti.

I documenti possono essere statici (già presenti sul server al momento della richiesta) o dinamici (generati al momento della richiesta dal server o dal client)

L'utente accede ai documenti fornendo l'URL al programma client (Web Browser).

Architettura WWW



URL

Gli URL (Uniform Resource Locator) sono identificatori univoci di documenti WWW.
Sono composti da 4 parti schema://NomeServer:port/NomeLocale

- **schema** è il protocollo per raggiungere il documento
- **NomeServer** è il nome DNS del server che contiene il documento
- **Port** è la porta di ascolto del server
- **NomeLocale** è l'identificatore del documento sul server

Ad esempio: <http://www.company.com:81/a/b/c.html>

http è il protocollo più utilizzato per l'accesso a documenti WWW.

Altri schema diffusi sono:

- ✓ https è la versione cifrata del protocollo http
- ✓ ftp (vedi ad esempio qui: <https://www.gnu.org/prep/ftp.html>)
- ✓ file (esempio <file:///C:/> identifica i file in C:)

Riferimenti:

https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/Identifying_resources_on_the_Web

Tipi Mime

Oltre ai documenti ipertestuali HTML l'architettura WWW supporta un numero sempre crescente di altri formati, denominati MIME-Types

Elenco di Mime-Type comuni:

https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/MIME_types/Common_types

Per ogni Mime-Type il Browser avrà associata la modalità di gestione

Per alcuni formati il Browser conterrà l'interprete necessario per visualizzarli, per altri formati si appoggerà a componenti software esterne, i Plug-in o gli helper:

- ✓ - Un plug-in è un modulo esterno che il browser installa come estensione di se stesso
- ✓ - L'Helper (o applicazione di supporto) è un programma autonomo eseguito come processo separato a cui il browser passa il file da visualizzare.

I Web Client

Prevalentemente **browser grafici** (Firefox, Chrome, IExplorer, ecc)

Dispongono di una Cache su disco per i documenti visitati di recente

Sequenza di operazioni:

- ▶ Input dell'URL
- ▶ Verifica se il documento è presente in Cache
- ▶ Risoluzione del nome DNS nell'indirizzo IP
- ▶ Apertura della connessione TCP
- ▶ Invio della richiesta al server mediante il protocollo indicato nello schema
- ▶ Download del documento
- ▶ Parsing del documento
- ▶ eventuale richiesta di documenti collegati
- ▶ Visualizzazione del documento direttamente o mediante viewer esterni
- ▶ Rilascio della connessione TCP (se non vi sono altre richieste entro breve termine)

Un client Web può essere utilizzato per il **download di documenti**.

Esempi in ambiente Linux o MacOsX: wget e curl

I Web Server

E' un programma che si occupa di fornire, su richiesta del client browser, una pagina WWW.

The Apache Software Foundation è il nome di un gruppo di lavoro
(<http://www.apache.org/>) che sta portando avanti diversi progetti Open Source tra cui due
tra i più diffusi Server Web: **Apache** e **Tomcat**.

Altri server meno diffusi: **NGINX**

Sequenza di operazioni base del server:

- ▶ Accetta la connessione TCP da un client
- ▶ Determina dall'URL il percorso del documento richiesto
- ▶ Accede al documento su disco
- ▶ Invio al client di intestazione (Mime-Type, ..) e contenuto del documento
- ▶ Rilascio della connessione TCP

Web Caching

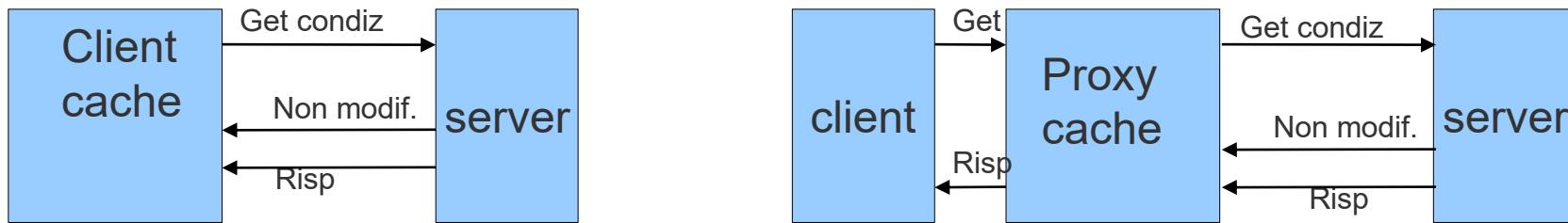
Fare Caching significa memorizzare temporaneamente le pagine in un punto più vicino al client per una **visualizzazione più rapida** e per **ridurre il carico del server**.

I **Browser** operano automaticamente caching sul disco locale dei documenti visitati.

Se il documento richiesto è presente sul disco locale viene fatto un GET condizionale (if-modified-since) in cui si chiede se esiste una versione più recente del documento.

Esempio:

If-Modified-Since: Tue, 17 Mar 2015 07:00:00 GMT



Una LAN può organizzare un servizio di caching disponibile per tutti gli utenti della rete.
Per utilizzare il servizio l'utente deve attivarlo esplicitamente.

Questo servizio di cache viene denominato **Proxy** poiché l'accesso al server che contiene il documento richiesto viene realizzato da Proxy/Cache server.

Il **Reverse Proxy** è un tipo di Proxy che recupera i contenuti per conto di un client da uno o più server. Questi contenuti sono poi trasferiti al client come se provenissero dallo stesso Reverse Proxy, che quindi appare al client come un Web server.

Cache Expiration

Alcuni documenti HTML possono contenere un'intestazione “Expires” che indica il tempo di validità del documento e verrà utilizzata per decidere se utilizzare la copia o recuperare il documento dal server.

```
<head>
<META HTTP-EQUIV="expires" CONTENT="Mon, 24 Mar 2008 08:21:57 GMT">
</head>
```

Altri documenti usano l'intestazione “no-cache” per impedire che il documento venga inserito nella cache (tipicamente pagine dinamiche).

```
<head>
<META HTTP-EQUIV="Pragma" CONTENT="no-cache">
</head>
```

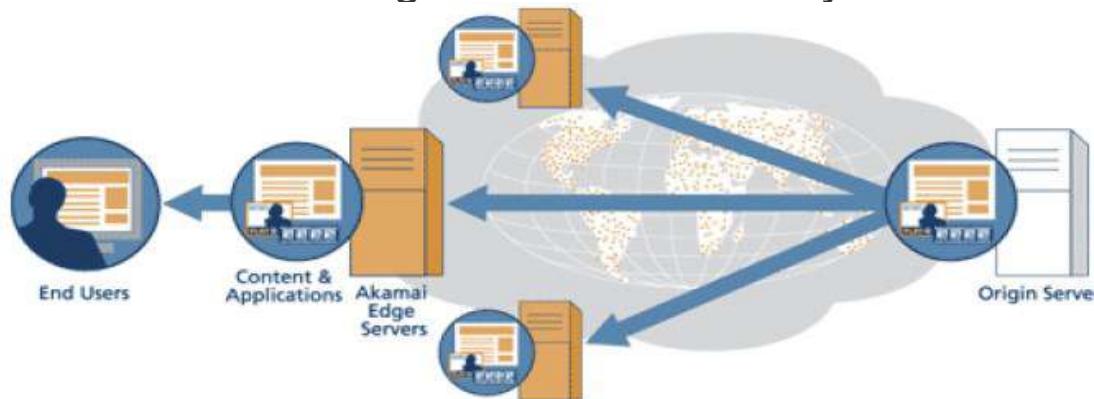
Content Delivery Network (CDN)

Per contenuti che devono essere distribuiti su scala globale (downloads, streaming, ..) esistono 2 approcci: Web caching e CDN.

Con il Web caching è il client a attivare le copie mentre con CDN è il provider che distribuisce copie dei contenuti in un insieme di nodi in differenti posizioni e redirige i client in modo che usino un nodo a lui vicino.

Possibili architetture per realizzare una rete CDN:

- **DNS redirection:** Il name server della CDN è gestito dalla CDN stessa. Il client ottiene dal DNS l'indirizzo Unicast della copia più vicina a lui.
- **Routing Anycast:** i mirror server hanno gli stessi indirizzi Anycast e il compito di trovare il server più vicino



Una delle reti CDN più note è Akamai, con oltre 29000 server sparsi su circa 70 nazioni.

Riferimenti: <https://www.akamai.com/it/our-thinking/cdn/what-is-a-cdn>

HTML

La maggior parte dei documenti WWW sono scritti in HTML.

HTML è un linguaggio di markup, ovvero contiene direttive di formattazione.

I contenuti sono testo formattato (font, colore, liste, tabelle, ecc) e eventuali riferimenti a oggetti esterni (immagini, video, suoni, hyper-link ad altri documenti)

Le direttive per la formattazione sono dette TAG e sono racchiuse tra parentesi angolari (esempio **** testo evidenziato ****)

Il documento HTML è delimitato dal TAG **<html>..</html>** e comprende una intestazione **<head> .. </head>** e un corpo **<body> .. </body>**

Esempio:

```
<html>
<head>
<title> Le informazioni in Head non appaiono nel documento </title>
</head>
<body>
<b> Nel Body viene scritto il testo formattato del documento </b> <p>
<a href="http://www.unipr.it"> Questo e' un collegamento ipertestuale </a>
<p>
Le immagini vengono incluse nel documento nel seguente modo: <p>

</body>
</html>
```

Standard HTML

HTML continua ad evolversi.

HTML 2.0 è il primo vero standard di HTML rilasciato nel 1995 da **W3C**

HTML 3.2, rilasciato nel Gennaio 1997, è stato adottato in diversi browser.

HTML 4.01, rilasciato nel 1999 da W3C, è un'altra release con molte implementazioni

HTML 5, rilasciato da W3C nel 2014.

I Browser e Server Web hanno generalmente implementato le varie release HTML in modo più o meno rigoroso, sorvolando spesso su errori di formattazione e in alcuni casi, aggiungendo TAG non standard, funzionanti solo su alcune piattaforme.

XHTML è un linguaggio di markup che associa alcune proprietà dell'XML nell'HTML.

Un file XHTML è un pagina HTML scritta in conformità con lo standard XML.

Style Sheet

HTML è un linguaggio di Markup che mescola il contenuto alla formattazione. Infatti alcuni TAG descrivono il contenuto del documento, indipendentemente dalla sua rappresentazione finale (esempio `` oppure ``), mentre altri TAG descrivono il modo in cui il documento dovrà apparire al lettore (esempio ``)

Con la crescente complessità e varietà di utilizzo dei documenti Web è sorta la necessità di separare i 2 aspetti del documento. Per questo motivo sono stati introdotti i “Fogli di Stile” (Style Sheet) che sono file associati ad un documento in cui vengono confinate le informazioni di formattazione.

CSS (Cascading Style Sheet) sono i fogli di stile supportati da HTML introdotti nel 1996 da W3C.

Esempio: mystyle.css

```
.piccolo {font-style:normal; font-size:12px; }
```

Esempio: mydoc.html

```
<html>
<head> <link rel="stylesheet" type="text/css" href="mystyle.css"> </head>
<body> <div class="piccolo"> ciao </div> </body>
</html>
```

Il protocollo HTTP

E' un protocollo testuale di tipo request/response che utilizza il servizio 80/TCP.

Riferimenti: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>

Prima versione V0.9 rilasciata dal HTTP Working Group nel 1991 (obsoleto)

Versini successive: HTTP/1.0 rilasciato nel 1996 (obsoleto),

HTTP/1.1 del 1997 , HTTP/2 del 2015 e HTTP/3 del 2022

Deve trasportare un messaggio di richiesta dal client al server ed un messaggio di risposta dal server al client con il documento richiesto. Ogni messaggio è formato da una intestazione (Header) ed un corpo (Body) separati da riga vuota (CR LF).

<METHOD> <URI> HTTP/1.0

<Header>: <Value>

<Header>: <Value>

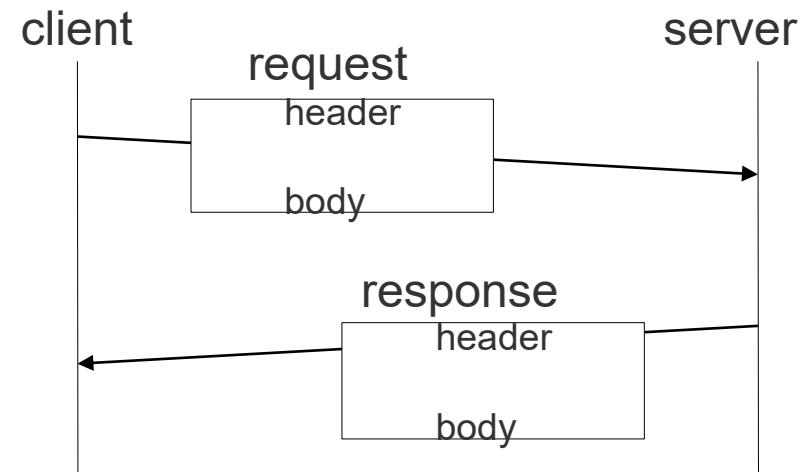
<BODY>

HTTP/1.0 <STATUS_CODE><REASON>

<Header>: <Value>

<Header>: <Value>

<BODY>



Metodi di richieste HTTP

GET : richiede tutte le informazioni disponibili per un determinato URL.
Il body del messaggio non è utilizzato.

```
GET /index.html HTTP/1.0
```

HEAD: Richiede solo l'header, senza la risorsa (il file HTML, l'immagine, ecc.).
Usato soprattutto per diagnostica.

```
HEAD /index.html HTTP/1.0
```

POST: è stato concepito in origine per inviare al server molte informazioni (nel Body della richiesta) , senza un limite sulla quantità di dati da trasmettere e sul tipo, ed in modo non visibile da URL.

```
POST /prog.cgi HTTP/1.0
Content-length: 10

01234566789
```

Metodi di richieste HTTP

OPTIONS: Richiede l'elenco dei metodi permessi dal server

```
OPTIONS * HTTP/1.0
[riga vuota]
...
Allow: GET,HEAD,POST,OPTIONS,TRACE
```

TRACE: Traccia una richiesta, visualizzando come viene trattata dal server.

DELETE: Cancella una risorsa (file) sul server. L'utente con cui gira il web server deve poter avere permessi in scrittura sul file indicato e il server deve essere configurato per poterlo fare.

PUT: Upload di un file sul server con il nome indicato e i contenuti specificati nel body.

Principali intestazioni HTTP

Header	Type	Contents
User-Agent	Request	Information about the browser and its platform
Accept	Request	The type of pages the client can handle
Accept-Charset	Request	The character sets that are acceptable to the client
Accept-Encoding	Request	The page encodings the client can handle
Accept-Language	Request	The natural languages the client can handle
Host	Request	The server's DNS name
Authorization	Request	A list of the client's credentials
Cookie	Request	Sends a previously set cookie back to the server
Date	Both	Date and time the message was sent
Upgrade	Both	The protocol the sender wants to switch to
Server	Response	Information about the server
Content-Encoding	Response	How the content is encoded (e.g., gzip)
Content-Language	Response	The natural language used in the page
Content-Length	Response	The page's length in bytes
Content-Type	Response	The page's MIME type
Last-Modified	Response	Time and date the page was last changed
Location	Response	A command to the client to send its request elsewhere
Accept-Ranges	Response	The server will accept byte range requests
Set-Cookie	Response	The server wants the client to save a cookie
If-modified-since	Request	Allows a 304 Not Modified to be returned if content is unchanged.

http://en.wikipedia.org/wiki/List_of_HTTP_header_fields

Risposte del Server

La prima riga della risposta del server contiene un codice che classifica la risposta:

Code	Meaning	Examples
1xx	Information	100 = server agrees to handle client's request
2xx	Success	200 = request succeeded; 204 = no content present
3xx	Redirection	301 = page moved; 304 = cached page still valid
4xx	Client error	403 = forbidden page; 404 = page not found
5xx	Server error	500 = internal server error; 503 = try again later

Esempio di richiesta/risposta con telnet

Richiesta del Client:

```
telnet localhost 80
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^].
GET /index.html HTTP/1.0
[riga vuota]
```

Risposta del server:

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Thu, 24 Nov 2022 11:34:55 GMT
Content-Type: text/html
Content-Length: 612
Last-Modified: Wed, 23 Nov 2022 08:24:08 GMT
Connection: close
ETag: "637dd8a8-264"
Accept-Ranges: bytes

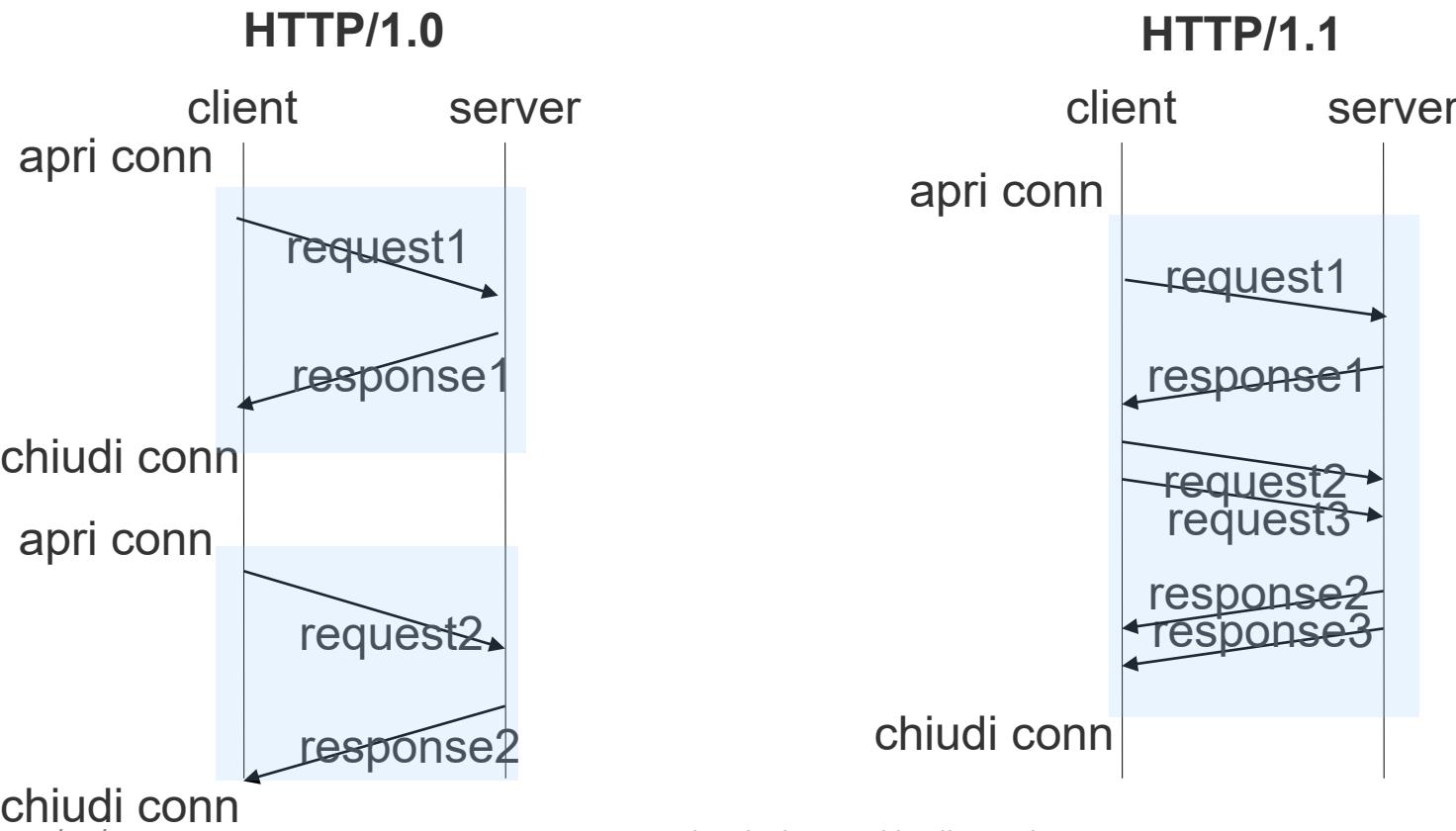
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully
working. Further configuration is required.</p>
```

HTTP 1.1: connessioni persistenti e parallele

HTTP 1.0 richiede la disconnessione TCP tra 2 richieste successive.

HTTP 1.1 supporta connessioni persistenti (riutilizzo di una conn. per diverse richieste)

HTTP 1.1 supporta anche richieste parallele



HTTP 1.1: Chunked Transfer Encoding

In HTTP 1.0 chi spedisce un dato include nell'intestazione la dimensione in byte del dato stesso, mediante l'intestazione **“Content-length: nn”**.

Chunked Transfer Encoding è una modalità di trasferimento introdotta in HTTP 1.1 in cui i dati vengono inviati in una serie di “chunk”.

Viene utilizzata per la trasmissione di dati generati dinamicamente, di cui non conosciamo la lunghezza prima di iniziare la trasmissione, come ad esempio gli eventi in streaming. Si utilizza **“Transfer-Encoding: chunked”** invece di “Content-length: nn”

La dimensione di ogni chunk viene inviata appena prima del chunk stesso in modo che il ricevente possa capire quando ha finito di ricevere il chunk.

Il trasferimento termina con un chunk finale pari a 0.

Riferimento:

http://en.wikipedia.org/wiki/Chunked_transfer_encoding

```
HTTP/1.1 200 OK
Date: Mon, 22 Mar 2004 11:15:03 GMT
Content-Type: text/html
Transfer-Encoding: chunked
```

```
29
<html><body><p>The file you requested is
5
3,400
23
bytes long and was last modified:
1d
Sat, 20 Mar 2004 21:12:00 GMT
13
.</p></body></html>
0
```

Mancanza di stato e Cookie

Il Web è privo di stato: se un Browser richiede più documenti da un server ogni richiesta è indipendente; il server non ricorda i contatti precedenti.

In alcuni casi però sarebbe utile avere “memoria”. Ad esempio se l’accesso ai documenti richiede autenticazione , autorizzazione, ecc.

I Cookie sono stati introdotti da Netscape e formalizzati in RFC 2109 per risolvere il problema.

I Cookie sono generati dal server e scaricati assieme al documento.

Il browser li memorizza in una opportuna directory, ma volendo li può disabilitare.

Informazioni contenute in un Cookie:

Domain	Path	Content	Expires	Secure
toms-casino.com	/	CustomerID=497793521	15-10-02 17:00	Yes
joes-store.com	/	Cart=1-00501;1-07031;2-13721	11-10-02 14:22	No
aportal.com	/	Prefs=Stk:SUNW+ORCL;Spt:Jets	31-12-10 23:59	No
sneaky.com	/	UserID=3627239101	31-12-12 23:59	No

Contenuto dei Cookie

Contenuto (Nome/valore) è una variabile ed un campo obbligatorio.

Expire (Scadenza) è un attributo opzionale che permette di stabilire la data di scadenza del cookie. Può essere espressa come data, come numero massimo di giorni oppure come Now (adesso) (implica che il cookie viene eliminato subito dal computer dell'utente in quanto scade nel momento in cui viene creato) o Never (mai) (implica che il cookie non è soggetto a scadenza e questi sono denominati persistenti).

Dominio e Percorso definiscono l'ambito di visibilità del cookie, indicano al browser che il cookie può essere inviato al server solo per il dominio e il percorso indicati. Se non specificati, come predefiniti prendono il valore del dominio e del percorso che li ha inizialmente richiesti.

Secure (Sicuro) indica se il cookie debba essere trasmesso criptato con HTTPS.

Come funzionano i Cookie

La prima volta che viene richiesto un URL il server invia i cookie al client inserendoli nell'intestazione, assieme al documento. Ad esempio:

```
HTTP/1.0 ....
```

```
Set-Cookie: tuocodice=1234567; expires=Tue, 18-Mar-08 18:43:09 GMT
```

```
Set-Cookie: tuonome=Mario; expires=Tue, 18-Mar-08 18:43:09 GMT
```

Il client memorizza i cookie ricevuti.

Quando l'utente torna a visitare la pagina il Browser cerca tra i Cookie che ha memorizzato un Cookie (non scaduto) con lo stesso dominio dell'URL.

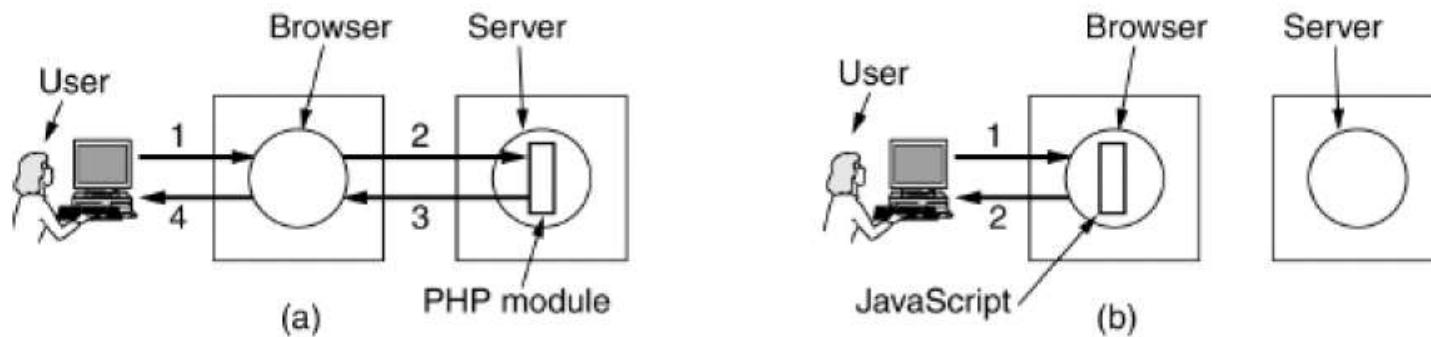
Se esiste viene fatto Upload del Cookie nell'Header assieme alla richiesta:

```
GET .... HTTP/1.0
```

```
Cookie: nome1=valore1 ; nome2=valore2
```

Pagine Web statiche e dinamiche

- ▶ **Le pagine statiche** sono file sul disco del server che vengono spediti al client assieme ad una intestazione (content-type ecc)
- ▶ **Pagine dinamiche:** il documento viene generato in tempo reale, su richiesta. La generazione è eseguita da un programma che può essere eseguito
 - **dal server**
 - via CGI (programmi esterni richiamati dal server)
 - scripting PHP, JSP, ASP (codice incorporato in HTML interpretato dal server)
 - **dal client (pagine attive)**
 - Javascript
 - Java Applet (richiede JVM sul client)
 - ActiveX (tecnologia Microsoft, codice compilato per Intel)



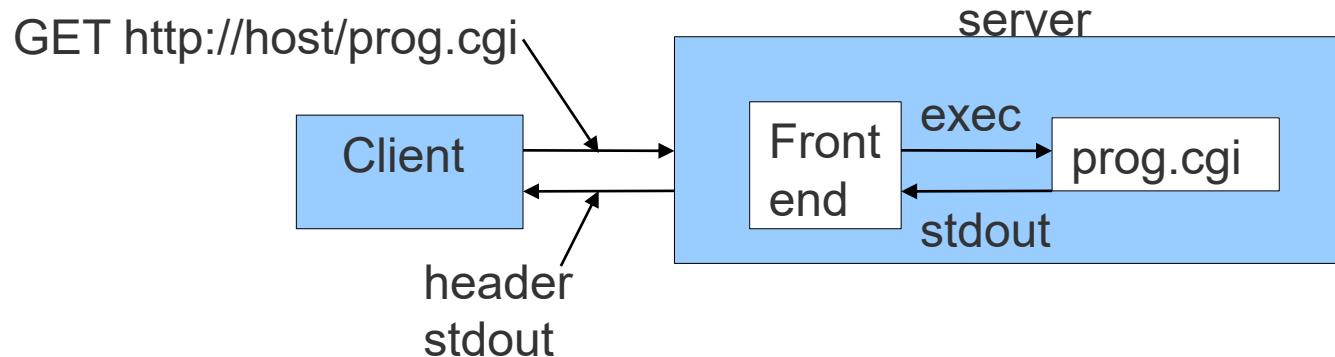
Pagine dinamiche con CGI

Il protocollo **CGI** (Common Gateway Interface, RFC 3875) consente di mettere in esecuzione un programma eseguibile sul server e di redirigere lo standard output del programma verso il client il quale lo interpreterà come una normale risposta http.

Per attivare un programma il client utilizza lo stesso modello URL utilizzato per il riferimento alle pagine statiche, con i metodi GET o POST.

Il server può riconoscere un programma cgi in base alla sua estensione (e.g. .cgi) o alla sua posizione (es /cgi-bin/..)

Normalmente l'output è in formato HTML, ma può assumere anche altre forme (immagini, dati binari, istruzioni particolari per il browser, ..)



Passaggio dei parametri

L'esecuzione di una pagina dinamica deve prevedere la possibilità di passare parametri o dati all'eseguibile.

Questo può avvenire con 2 metodi alternativi: GET e POST.

-**Il metodo GET** codifica i parametri all'interno della stringa URL

Esempio: `http://host/prog.cgi?param`

-**Il metodo POST** utilizza la parte Body della richiesta.

L'HTML fornisce diversi TAG per la codifica di parametri nella forma NOME=valore da passare con GET o POST

Ad esempio:

```
<FORM ACTION="http://host/prog.cgi" METHOD=GET>  
PARAM1: <INPUT TYPE="text" NAME="param1">  
PARAM2: <INPUT TYPE="text" NAME="param2" >  
<INPUT TYPE="submit" VALUE="Invia">  
</FORM>
```

PARAM1: val1	PARAM2: val2	Invia
--------------	--------------	-------

Genera la URL: `http://host/prog.cgi?param1=val1¶m2=val2`

Passaggio di dati con il metodo GET

Con il metodo GET la stringa di query (input) è inserita in coda alla URI del documento preceduta dal carattere “?”

GET http://host/prog.cgi?param1=val1¶m2=val2 HTTP/1.0

header..

header..

<cr><lf>

Il programma riceve la stringa attraverso la variabile di ambiente QUERY_STRING:

QUERY_STRING="param1=val1¶m2=val2"

Passaggio di dati con il metodo Post

I parametri passati con il metodo POST vengono inseriti nel Body della richiesta HTTP e il programma li riceve attraverso lo Standard Input.

Ad esempio, se scegliamo il metodo POST nella FORM dell'esempio precedente:

```
<FORM ACTION="http://host/prog.cgi " METHOD=POST> ...
```

Otteniamo una richiesta HTTP del tipo:

POST http://host/prog.cgi HTTP/1.0

header..

Content-length: 23

<cr><lf>

param1=val1¶m2=val2

Questo metodo ha 2 vantaggi nel passaggio dei parametri rispetto al metodo GET:

- ✓ i parametri non compaiono nella URI (e quindi non vengono tracciati nei file di log)
- ✓ è possibile trasferire non solo parametri, ma anche dati

Pagine dinamiche lato server con PHP

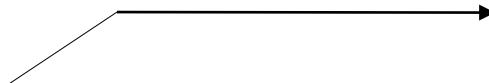
I server web supportano anche la possibilità di incorporare piccoli script all'interno del codice HTML che verrà eseguito al momento della consultazione della pagina.

Questo consente di realizzare documenti in cui solo una parte è dinamica.

Un linguaggio molto utilizzato per questo scopo è il PHP.

Esempio:

```
<html>
<head>
</head>
<body>
<FORM ACTION="http://host/prog.php" METHOD=GET>
PARAM1: <INPUT TYPE="text" NAME="param1">
PARAM2: <INPUT TYPE="text" NAME="param2" >
<INPUT TYPE="submit" VALUE="Invia">
</FORM>
</body>
</html>
```



```
<html>
<body>
<h1>Dati inseriti:</h1>
<? php echo $param1; ?>
<? php echo $param2; ?>
</body>
</html>
```

Pagine dinamiche lato client: javascript

In altre applicazioni è utile che il codice venga eseguito lato client.

Anche in questo caso viene incorporato codice di script nella pagina HTML.

Il linguaggio più popolare lato client è JavaScript. Esempio:

```
<html>
<head>

<script language="javascript" type="text/javascript">
Function response (test_form) {
...
}

</script>

</head>
<body>
<form>
PARAM1: <INPUT TYPE="text" NAME="param1">
PARAM2: <INPUT TYPE="text" NAME="param2" >
<input type="button" value="submit" onclick="response(this.form)">
</form>
</body>
</html>
```

Pagine dinamiche lato client: Applet

Un altro metodo molto diffuso è l'utilizzo di Applet Java.

E' necessario che il browser includa una JVM (quasi tutti i Browser)

Le Applet sono più portabili perché la JVM è la stessa su diverse piattaforme, mentre il supporto JavaScript può differire da un Browser all'altro.

Le Applet possono essere incorporate nelle pagine HTML:<applet> ... </applet>

Esempio:

```
<html>
<body>
<applet code="PrimoApplet.class">
</applet>
</body>
</html>

import java.applet.*;
import java.awt.*;

public class PrimoApplet extends Applet
{
    public void paint (Graphics g)
    {
        g.drawString("Ciao, io sono il primo applet.",0,50);
    }
}
```



UNIVERSITÀ
DI PARMA

DIPARTIMENTO DI SCIENZE MATEMATICHE, FISICHE ED INFORMATICHE
Corso di Laurea in Informatica

Il Livello Applicativo – Parte D

Multimedia

RETI DI CALCOLATORI - a.a. 2023/2024

Roberto Alfieri

Livello Applicativo: sommario

PARTE A

- ▶ Applicativi UDP: TFTP e DNS

PARTE B

- ▶ I servizi di posta elettronica: SMTP, POP e IMAP.

PARTE C

- ▶ Il World Wide Web

PARTE D

- ▶ **Multimedia**

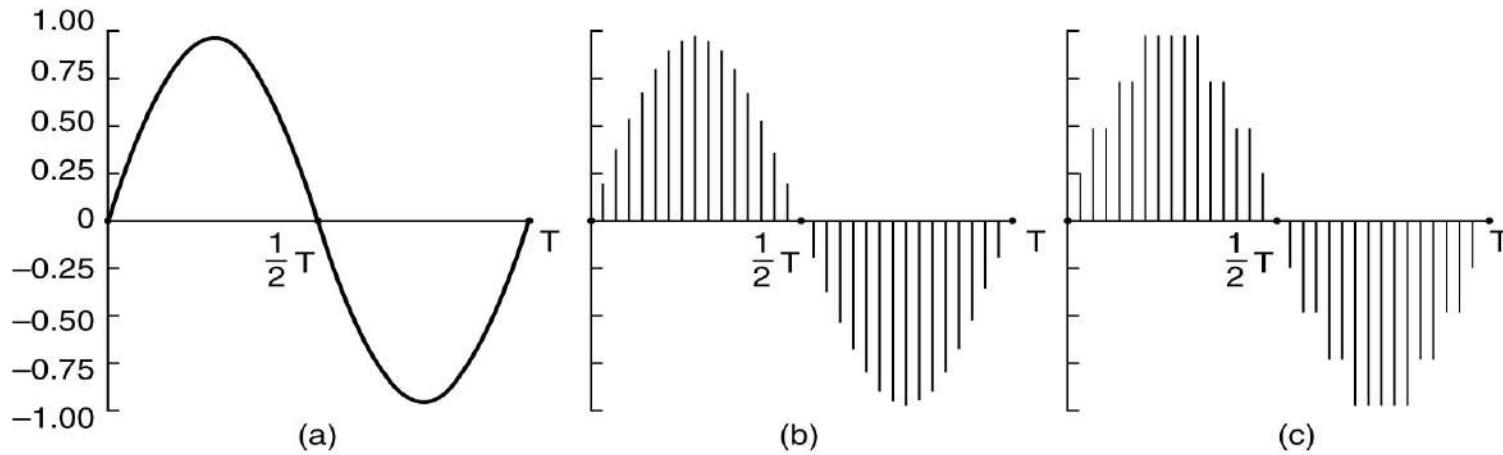
Multimedia

I dati multimediali hanno caratteristiche peculiari rispetto ai dati «classici» che utilizziamo nelle reti di calcolatori e hanno requisiti che richiedono un diverso QoS:

- ✓ I dati sono in origine analogici in 1 (audio) o 2 (video) dimensioni spaziali ed eventualmente una dimensione temporale e devono quindi subire una discretizzazione che introduce inevitabilmente una approssimazione.
- ✓ I dati possono essere pesanti (in particolare i video) e richiedono quasi sempre un processo di compressione/decompressione
- ✓ Sono tollerati (piccoli) errori di trasferimento, ma è richiesta una comunicazione con alto bit-rate e basso jitter.
- ✓ La rete internet, nonostante esistano tecniche per il supporto della QoS (vedi i Servizi Differenziati) attualmente è sostanzialmente best effort.
- ✓ Non esiste quindi un approccio standard per la gestione dei dati multimediali in rete ma vendono utilizzate in molti casi soluzioni proprietarie.

Audio Analogico e Digitale

Onda sonora percepita dall'uomo: onda monodimensionale con frequenze comprese tra 20 e 20.000 Hz (voce sotto i 3KHz).

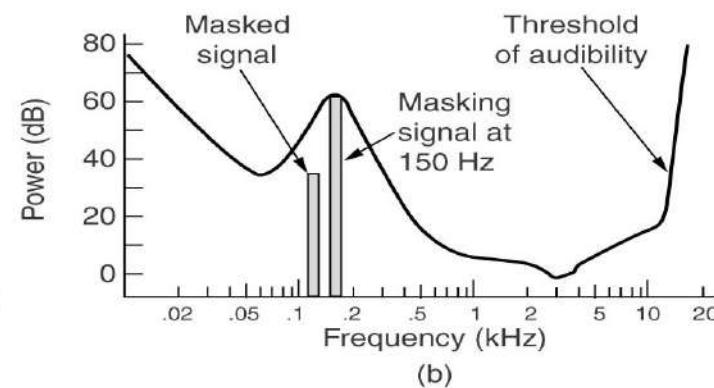
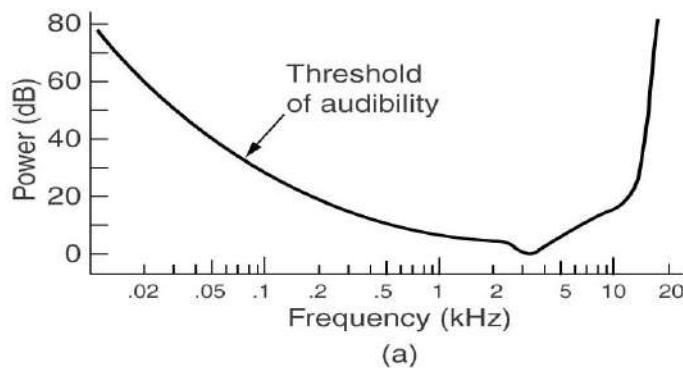


Conversione analogico/digitale:

- ▶ **La frequenza di campionamento è determinata dal teorema di Nyquist:**
- ▶ Freq. di campionamento = 2 * Freq. Max del segnale da digitalizzare
- ▶ **quantizzazione:** numero discreto di valori possibili per le letture nel campionamento (8 bit = 256 valori, 16 bit = 65.536)
- ▶ Il sistema telefonico usa 8 bit per 8.000 volte al secondo (max 4KHz)
- ▶ CD audio: 44.100 campioni al secondo di 16 bit => 1,411Mbps (stereo)

Compressione Audio

- ▶ Per la trasmissione su internet è necessaria forte compressione.
- ▶ La compressione audio si può fare in 2 modi:
 - **Conversione per forma d'onda:** il segnale viene convertito, usando la trasformata di Fourier, nelle sue componenti nel dominio delle frequenze; ogni componente viene codificata con la minima quantità di bit
 - **Codifica percettiva:** sfrutta alcuni limiti del sistema uditivo umano (psicoacustica) per codificare il segnale in modo che sembri lo stesso ad un ascoltatore umano pur essendo diverso dal segnale originario, utilizzando la tecnica del mascheramento.



I formati MP3 e AAC eseguono la trasformata di Fourier, quindi codificano solo le frequenze non mascherabili.

CD audio: si arriva a circa 100 Kbps (1.4Mbps non compresso, Compressione 14:1)

Video digitale

- ▶ **Video digitale:** sequenza di fotogrammi, ognuno dei quali composto da una griglia rettangolare di elementi detti **pixel**
- ▶ La geometria è quella del video analogico con la differenza che le linee di scansione vengono sostituite da righe di pixel discreti
- ▶ Per avere un *movimento fluido* servono **almeno 25 fotogrammi** al secondo
- ▶ Per un flusso video alla risoluzione di 640x480, con 24 bit per pixel e 30 fotogrammi al secondo serve una linea di comunicazione a 200 **Mbps**

- ▶ La **compressione** è l'unica possibilità per riuscire ad inviare filmati video su internet.
- ▶ Servono due algoritmi: uno di **codifica** per la compressione all'origine, uno di **decodifica** per la decompressione alla destinazione
- ▶ Il sistema di codifica/decodifica può essere *irreversibile (lossy)*: a costo di una piccola **perdita di informazioni** si ottiene un fattore di compressione elevato

JPEG

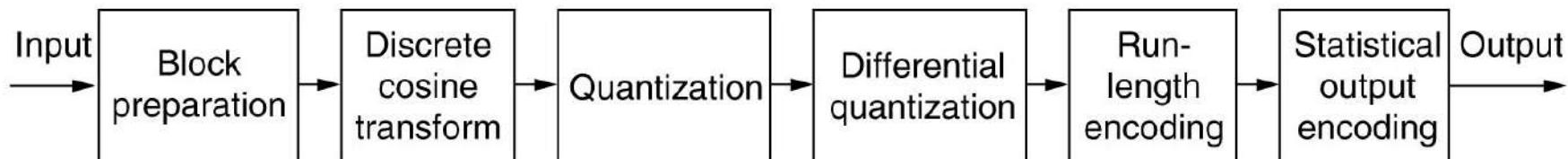
Un video è essenzialmente una sequenza di immagini con l'audio.

un algoritmo per la codifica Video è la codifica in successione di ogni singola immagine, con o senza perdite di informazione

Lo standard JPEG esegue la compressione di immagini statiche a toni continui (come le foto)

Processo di compressione:

- ▶ L'immagine viene suddivisa in blocchi di 8x8 pixel
- ▶ Per ogni blocco si passa dallo spazio tempo allo spazio delle frequenze tramite DCT
- ▶ vengono eliminati i coefficienti DCT meno importanti
- ▶ Si passa ad una rappresentazione differenziale di un blocco rispetto al blocco precedente
- ▶ Linearizzazione dei 64 elementi di un blocco
- ▶ Codifica di Huffman per assegnare codici più brevi ai numeri più frequenti

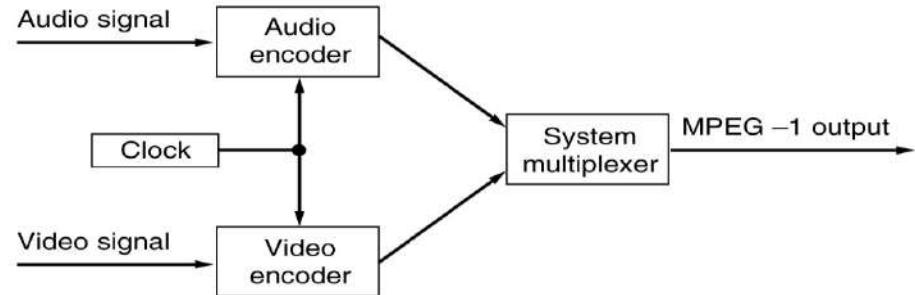


MPEG

E' un gruppo di lavoro (Motion Picture Experts Group) che dal 1993 si occupa di algoritmi e formati per le immagini in movimento.

- ▶ MPEG-1: pubblicato nel 1993, è ancora molto usato. Il suo scopo era quello di produrre un output di qualità simile a quella di un videoregistratore (352x240 per NTSC) usando un bit rate pari a 1,2Mbps con compressione 40:1 (richiederebbe circa 50,7Mbps non compressi)

- ▶ E' composto da tre parti:
 - Audio
 - Video
 - Multiplexer di sistema

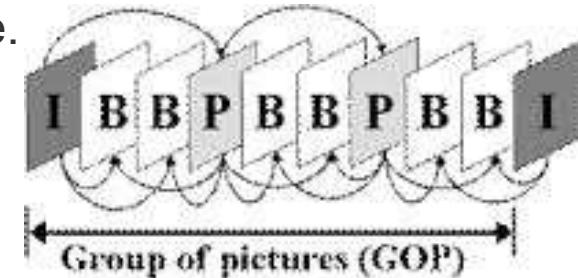


- ▶ Audio e video lavorano indipendentemente e sono sincronizzati usando un segnale comune a 90 KHz che emette il segnale del tempo corrente verso entrambi i codificatori
- ▶ **Ridondanza spaziale:** si usa la codifica JPEG per ogni singolo fotogramma. Utile soprattutto per accedere in modo casuale ad ogni singolo fotogramma
- ▶ **Ridondanza temporale:** si cerca di trarre vantaggio dal fatto che i fotogrammi consecutivi sono quasi identici

MPEG-1 e MPEG-2

MPEG-1 è composto da 3 tipi di fotogrammi:

- ▶ **I-frame** (Intracodificati): immagini statiche codificate in JPEG. Vengono inviate ad intervalli regolari (ad esempio ogni 10 frame).
- ▶ **P-frame** (Predittivi): E' una immagine che dipende dal frame precedente. Viene determinata calcolando le differenze blocco per blocco con l'ultimo fotogramma. Utile se non ci sono troppi cambiamenti rispetto alla precedente.
- ▶ **B-frame** (Bidirezionali): dipende sia dal precedente che dal successivo



MPEG-2: rilasciato nel 1996, è progettato per comprimere video con bit-rate compresi tra 4 e 6 Mbps, per essere inserito in trasmissioni NTSC e PAL per poi arrivare anche a risoluzioni superiori (HDTV)

- ▶ In prima approssimazione MPEG-2 è un superset di MPEG-1
- ▶ Utilizza fotogrammi I,P,B
- ▶ La trasformata DCT utilizza blocchi 10x10 invece di blocchi 8x8
- ▶ Supporta sia immagini progressive che interallacciate
- ▶ Supporta più livelli di risoluzione:

Low (352x240 compatibile MPEG-1), Main (720x480) High-1440 (1440x1152), High (1920x1080)

Dati multimediali in rete

Esistono diversi modalità di utilizzo del multimedia in internet, con diverse requisiti di rete:

Streaming di contenuti registrati (e.g. Youtube, Netflix)

- Numerosi flussi singoli. Riproduzione durante la ricezione.

Realtime streaming (e.g. radio Internet, dirette sportive)

- Ridurre il ritardo per minimizzare lo scostamento temporale rispetto alla trasmissione via etere.
- Centinaia o migliaia di utenti contemporanei

Conferenza in tempo reale (e.g. Skype)

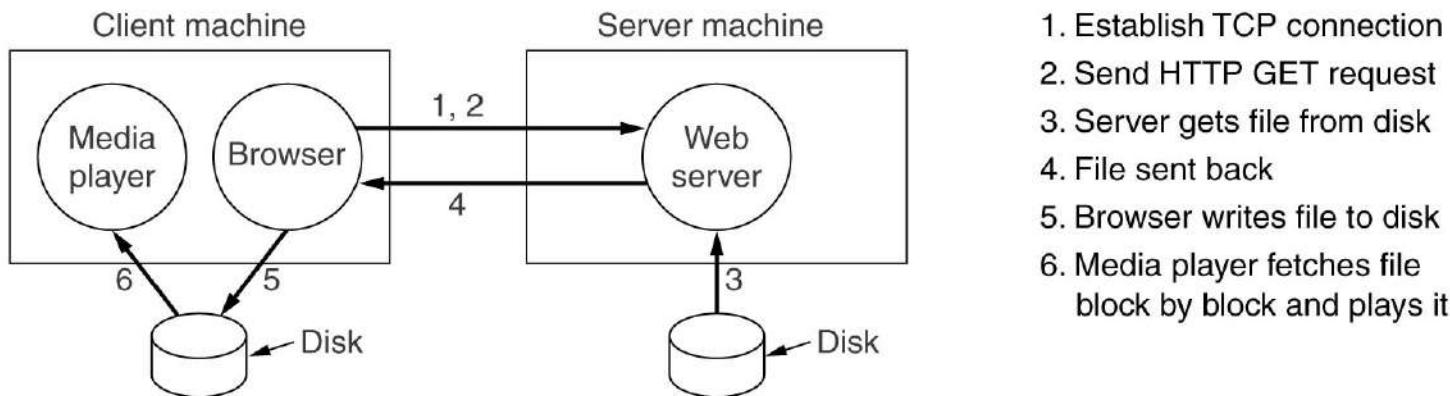
- L'interattività richiede latenze molto contenute.
- Numero di partecipanti tipicamente limitato ma in alcuni casi può essere elevato.

Streaming di contenuti registrati: Download and play

La tecnica più semplice per l'utilizzo di contenuti multimediali registrati su un server remoto è download and play:

il file deve essere scaricato completamente prima di poter essere riprodotto utilizzando il sistema "classico" del MIME

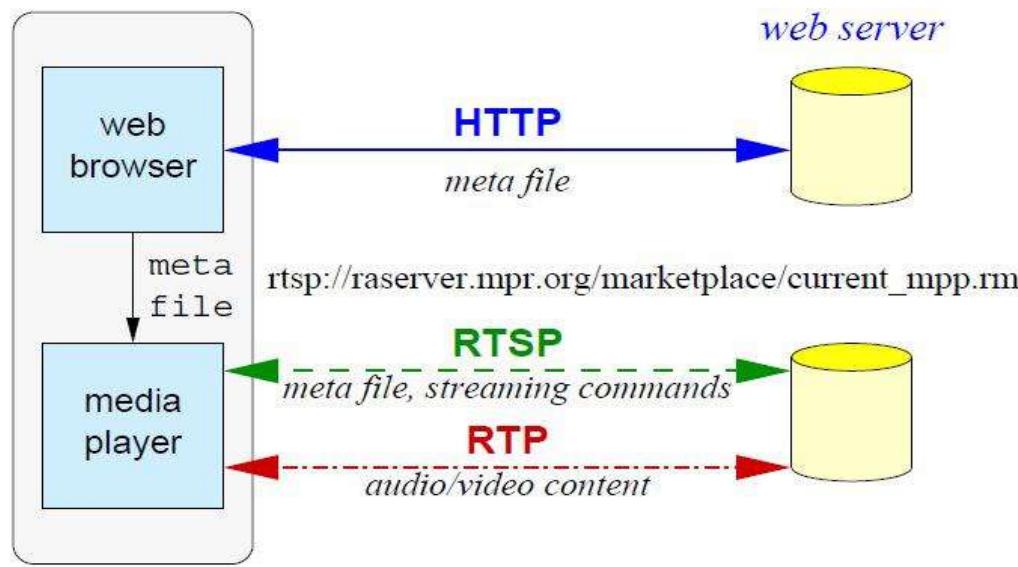
(esempio Content-type: video/mp4) e delle helper applications



Sistema poco efficiente: si deve scaricare il file prima di iniziare l'ascolto.

Streaming di contenuti registrati: Streaming on download

La soluzione generalmente adottata (Audio Streaming) è la seguente : il file collegato al titolo non è quello contenente l'audio ma un **metafile** che rimanda al file vero e proprio da ascoltare (il contenuto potrebbe essere la sola riga rtsp://server/file.mp3). Il browser, una volta passato il **metafile** all'applicazione esterna, non fa più parte del ciclo di comunicazione



L'accesso al file multimediale (e.g. rtsp://server/file.mp3) avviene mediante un protocollo per la gestione dell'interfaccia utente (Play/Record/Pause/ ecc) denominato Real Time Streaming Protocol (RTSP).

Il trasporto del flusso multimediale avviene su un canale separato basato sul protocollo RTP (Real-time Transport Protocol) o HTTP.

Real Time Streaming Protocol

- Il protocollo RTSP supporta un set di comandi che vengono inviati al server mediante un scambio testuale simile all'HTTP

Command	Server action
DESCRIBE	List media parameters
SETUP	Establish a logical channel between the player and the server
PLAY	Start sending data to the client
RECORD	Start accepting data from the client
PAUSE	Temporarily stop sending data
TEARDOWN	Release the logical channel

Vedi un esempio di dialogo: http://en.wikipedia.org/wiki/Real_Time_Streaming_Protocol

*Nota: YouTube usa HTTP/TCP. RTSP è comunque utilizzata per la rete dei cellulari
<http://m.youtube.com> I video sono memorizzati nella CDN di Google.*

Realtime Transport Protocol

RTP (RFC 3350) è un protocollo client/server per il trasporto di flussi anche multipli di dati audio e video. E' basato su UDP e può funzionare sia in unicast che in multicast.

I campi principali dell'intestazione RTP sono:

- Payload type: rende possibile l'identificazione del contenuto
(esempi: 26 → JPEG, 32 → MPEG1, 33 → MPEG2)
- Sequence Number: numerazione progressiva per il riordino
- Timestamp: istante di campionamento del primo byte nel payload.
- Synchronized Source ID: identificatore della sorgente di stream, per distinguere diversi flussi contemporanei.

Con l'aiuto di un piccolo buffer il ricevente può ricostruire il flusso nella sequenza temporale corretta e rifasare eventuali flussi contemporanei.

Payload Type	Sequence Number	Timestamp	Synchronization Source Identifier	Miscellaneous Fields
--------------	-----------------	-----------	-----------------------------------	----------------------

RTP Header

Alcuni fornitori preferiscono implementare protocolli di trasporto brevettati.

Ad esempio i server di Realnetworks utilizzano il protocollo di trasporto [Real Data Transport \(RDT\)](#) di proprietà della RealNetworks stessa.

Realtime streaming

Streaming in tempo reale di eventi (sportivi o musicali) a cui assiste un elevato numero di utenti.

Requisiti:

- **Ridurre il ritardo** per minimizzare lo scostamento temporale rispetto alla trasmissione via etere.
- **Centinaia o migliaia di utenti contemporanei**

Soluzioni:

- Multicast con protocolli RTP/UDP Vedi [FastWeb multicast](#)
Buona scalabilità, ma sia multicasting che la porta RTP possono incontrare problemi nel supporto da parte dei provider.
- Dynamic adaptive streaming over HTTP, in unicast.

Dynamic Adaptive Streaming over HTTP

Dynamic Adaptive Streaming è una tecnica particolare, basata su HTTP, che adegua la qualità dello streaming video alle risorse disponibili nel dispositivo dell'utente, quali le condizioni della rete (bandwidth) o la capacità della CPU, avendo come risultato il miglioramento della QoS dell'utente.

Funziona bene se il numero di utenti è moderato. Necessario avere il server con buona connettività internet, eventualmente in CDN.

Principali tecnologie: MPEG-DASH e HLS

Riferimenti: https://en.wikipedia.org/wiki/Dynamic_Adaptive_Streaming_over_HTTP

<https://www.cloudflare.com/learning/video/what-is-mpeg-dash/>

Conferenze in tempo reale

Comunicazioni multimediali tra più utenti in tempo reale

Requisiti:

- L'interattività richiede latenze molto contenute.
- Numero di partecipanti tipicamente limitato ma in alcuni casi può essere elevato.

Latenza: La rete telefonica considera accettabili latenze in una direzione entro **150 ms**

Componenti principali della latenza:

- Il ritardo di propagazione. *Ad esempio in una fibra ottica per 8000 Km (Seattle-Amsterdam) è di 40ms.*
- Ritardo dovuto al riempimento del pacchetto. *Un pacchetto da 1KB impiega 125ms per riempirsi a 64Kbps. Se si utilizzano pacchetti piccoli (160 bytes) è possibile ridurre la latenza, accettando un degrado di larghezza di banda.*
- Ritardo per compressione/decompressione. Soprattutto per il video.

Buffer: è ancora necessario per evitare riproduzioni a scatti e per il riordino dei pacchetti, ma deve essere piccolo per limitare la latenza.

Protocolli: Occorre un protocollo per il trasporto dei dati, che generalmente è RTP/UDP.

Il protocollo di controllo deve gestire anche la **Segnalazione**, ovvero le problematiche di attivazione e gestione delle chiamate.

Voice over IP (VoIP)

Voice over IP indica una tecnologia che rende possibile effettuare una conversazione, analoga a quella che si potrebbe ottenere con una rete telefonica, sfruttando una connessione Internet.

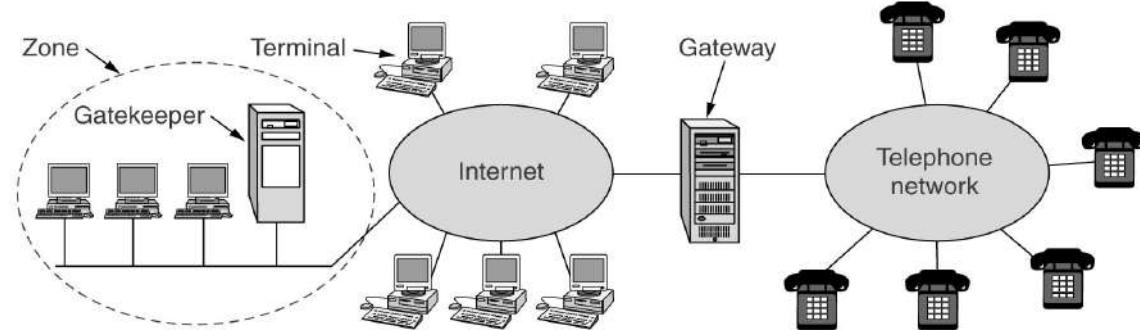
Per il **trasporto dei dati**, nella grande maggioranza delle implementazioni VoIP, viene adottato RTP (Real-time Transport Protocol) su UDP su IP.

Per quanto riguarda la **segnalazione** il processo di standardizzazione non si è ancora concluso. Al momento, sono coinvolti diversi enti internazionali di standardizzazione tra cui l'ITU (International Telecommunications Union) e l'IETF con due proposte alternative: **H.323** di ITU e **SIP** di IETF.

VoIP con H.323

H.323, creato nel 1996 da ITU, rappresenta una panoramica architetturale (dal punto di vista dell'industria telefonica) della telefonia su Internet.

Non emette proprie specifiche ma fa riferimento a diversi protocolli specializzati:
codifica del parlato, impostazione delle chiamate, segnalazione, trasporto di dati, etc.



Speech	Control				
	G.7xx	RTCP	H.225 (RAS)	Q.931 (Call signaling)	H.245 (Call control)
RTP					
UDP				TCP	
IP					
Data link protocol					
Physical layer protocol					

VoIP con SIP

SIP (Session Initiation Protocol) è la risposta di IETF (RFC 3261) ad H.323, considerato un prodotto tipico per telecomunicazioni (grande, complesso e poco flessibile), con un protocollo più semplice e modulare per la telefonia via Internet

Describe come impostare le telefonate via internet, le videoconferenze e altre connessioni multimediali. Gestisce solamente la segnalazione, ovvero l'impostazione, la gestione e la terminazione delle sessioni, mentre per il trasporto si usa RTP.

Riferimenti:

http://www.garr.it/eventiGARR/ws9/pdf/Sommani_pres.pdf

Protocollo SIP

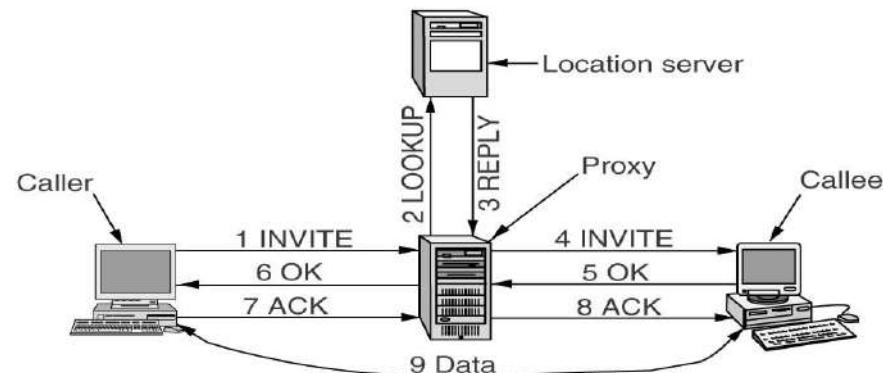
Un SIP-URI (RFC 3261) rappresenta lo schema di indirizzamento SIP per chiamare un altro soggetto attraverso il protocollo SIP. In altre parole, un SIP URI è il recapito telefonico SIP di un utente. Il SIP URI assomiglia ad un indirizzo e-mail scritto nel seguente formato: {sip|sips}:[user-part@[domain-part[:port]]

Esempio: *sip:alfierir@ekiga.net* Queste URI possono contenere indirizzi IPV4, IPV6 o numeri di telefono veri e propri: <sips:+004437612234@sip-proxy.org:5062>

Il protocollo è basato su UDP/5060 con transazioni richiesta/risposta in ASCII (simile ad HTTP). Una transazione inizia con una Request inviata da uno User Agent Client ad un Proxy e termina con una Final Response inviata in senso inverso.

L'RFC 3261 definisce i seguenti metodi:

Method	Description
INVITE	Request initiation of a session
ACK	Confirm that a session has been initiated
BYE	Request termination of a session
OPTIONS	Query a host about its capabilities
CANCEL	Cancel a pending request
REGISTER	Inform a redirection server about the user's current location





UNIVERSITÀ
DI PARMA

DIPARTIMENTO DI SCIENZE MATEMATICHE, FISICHE ED INFORMATICHE
Corso di Laurea in Informatica

Sicurezza delle reti - Parte A

Metodi di attacco e difesa

RETI DI CALCOLATORI - a.a. 2023/2024

Roberto Alfieri

La sicurezza delle reti: sommario

PARTE A

- ▶ I servizi di sicurezza
- ▶ Metodi e strumenti di Attacco
- ▶ Metodi e strumenti di Difesa

PARTE B

- ▶ Crittografia applicata e OpenSSL

PARTE C

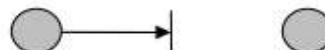
- ▶ Protocolli di autenticazione
- ▶ Ipsec e VPN
- ▶ Sicurezza delle reti WiFi

La sicurezza

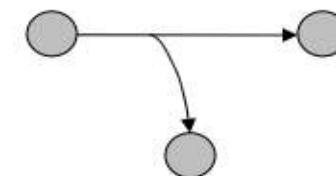
Con il termine sicurezza informatica si intende quel ramo dell'informatica che si occupa dell'analisi delle vulnerabilità, del rischio, delle minacce e della successiva protezione dell'integrità logico-funzionale.

La gran parte delle minacce derivano dalla rete Internet, per cui la sicurezza informatica è un tema importante nello studio delle reti di calcolatori.

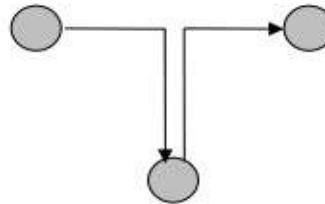
Trasmissione regolare



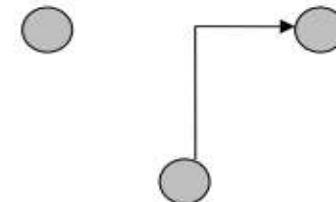
Interruzione



Intercettazione



Modifica



Fabricazione

Politiche di sicurezza

I beni da proteggere (asset) sono principalmente i dati gestiti dalla rete informatica, ma per fare ciò bisogna anche proteggere i computer ed i vari dispositivi presenti nella rete.

La realizzazione di un sistema che garantisca una assoluta protezione da abusi è impossibile, ma è possibile attivare meccanismi di sicurezza tali da limitare e scoraggiare i tentativi.

La politica di sicurezza è ***quindi un compromesso, dettato dalle proprie necessità, tra il costo per attivarla ed il beneficio ottenuto in termini di diminuzione del rischio.***

Security services

Per proteggere un host in rete o una comunicazione occorre stabilire dei servizi di sicurezza che possono essere classificati nel seguente modo:

Autenticazione (Authentication): un servizio che consente di accettare l'identità dichiarata da una entità (origine dei dati o peer in una comunicazione) mediante la verifica di credenziali. Avviene in 2 step: Identificazione e verifica.

Autorizzazione (Authorization, Access Control): Protegge l'accesso ad una risorsa mediante l'applicazione di "Security Policy".

Confidenzialità/Riservatezza (Data Confidentiality): Impedisce l'utilizzo delle informazioni da accessi non autorizzati.

Integrità dei dati (Data Integrity): consente di garantire che i dati acceduti non sono stati modificati.

Integrità dei sistemi (System Integrity): protegge le risorse del sistema contro modifiche, distruzioni accidentali o non autorizzate.

Non ripudio (Non-Repudiation): fornisce protezione contro il ripudio nel coinvolgimento in una comunicazione.

- ▶ non ripudio della sorgente: prova chi è il mittente dei dati in una transazione
- ▶ non ripudio della destinazione: prova che i dati sono arrivati ad uno specifico destinatario

Disponibilità (Availability): Fornisce una protezione per garantire accessibilità di una risorsa di sistema o di rete.

Audit (Accountability, Traceability): Registrazione di eventi di sistema o di rete. Consente di rintracciare, ricostruire (ed eventualmente addebitare) l'utilizzo delle risorse.

Riferimenti: RFC2828 di IETF e la raccomandazione X.800 di ITU

Metodi e strumenti di attacco ai servizi di sicurezza

Esistono diversi metodi e strumenti che un attaccante può utilizzare per violare i servizi di sicurezza a protezione di un host o di una rete.

Gli attacchi possono essere **passivi**, se l'obiettivo è la raccolta di informazioni, in modo lecito o illecito, riguardo un possibile bersaglio, oppure **attivi** quando coinvolgono l'alterazione dei dati sui dispositivi di memorizzazione o nei flussi di comunicazione.

Solitamente un attacco passivo è funzionale ad un successivo attacco attivo.

Attacco passivo

Gli attacchi passivi includono quelle attività che consentono di raccogliere dati e informazioni di un asset bersaglio senza alcuna alterazione sui dati e sui dispositivi. L'obiettivo principale è la raccolta di informazioni sul target utili per un successivo attacco attivo.

Tipologie principali di attacco passivo:

- ▶ Scansione via rete del target e/o della rete a cui appartiene
 - ▶ Strumenti: port scanning (esempio nmap)
- ▶ Intercettazione delle comunicazioni
 - ▶ Strumenti: Sniffing (esempio tcpdump)
- ▶ Raccolta di informazioni Open Source (OSINT)
 - ▶ Strumenti: ricerca di informazioni forum, blog, siti di social network, siti di condivisione di video, wiki, record Whois e DNS di nomi di dominio registrati, metadati e file digitali, risorse web scure, dati di geolocalizzazione, indirizzi IP , i motori di ricerca delle persone e tutto ciò che può essere trovato online.

Queste attività sono il punto di partenza per un attaccante ma sono anche strumenti di difesa che possono consentire i conoscere i propri punti deboli.

Attacco passivo: Sniffing e scanning

Sniffing: Analizzare il traffico di rete utilizzando un analizzatore di protocollo.

E' necessario un accesso privilegiato all'interfaccia di rete.

Possono essere orientati ad analizzare una sequenza di pacchetti

- Esempi di strumenti: **tcpdump e wireshark**

Scanning: testare un intervallo di indirizzi IP e numeri di porta per vedere quali servizi o sistemi sono presenti ed attivi.

- Esempio di strumento : **nmap**

nmap è uno tool open-source per la network exploration e l'auditing.

<https://nmap.org/man/it/>

Opzioni significative: <https://nmap.org/book/port-scanning-options.html>

-sP (ping scan)

-sT (TCP connect() - default)

-sS (TCP syn, richiede i priv. di root)

-A rileva versione

-p seleziona le porte da testare (per default vengono scansionate le 1000 porte più popolari)

Esempi:

nmap 192.168.0.0/24

nmap -A 192.168.0.254 -p 22,8001-8060

Attacco passivo: OSINT

Open Source INTelligence si riferisce alla ricerca di informazioni tratte da fonti liberamente disponibili (non all'open source software).

Ricercare, esaminare, correlare le informazioni pubblicamente disponibili permette di ottenere informazioni sull'organizzazione, sui progetti, sui processi aziendali, sulle persone.

Le informazioni recuperabili tramite OSINT non sono utilizzabili soltanto per fini “negativi”
Possono essere utilmente impiegate per supportare decisioni strategiche, valutare campagne di marketing, verificare il “sentiment” e la reputazione online
Possono essere impiegate per migliorare la sicurezza dell'organizzazione

Modalità di OSINT

- ✓ **Manuale:** la ricerca viene effettuata direttamente dall'operatore
 - Le scelte vengono fatte sul momento. Raccolta e confronto oneroso, difficilmente “scalabile” ed elevato rischio di “perdere” informazioni
- ✓ **Automatica:** la ricerca viene effettuata tramite strumenti parametrizzabili
 - Deve essere effettuata una scrematura a posteriori. Potenzialmente molto efficace, facilmente “scalabile” ed integrabile, richiede enormi investimenti (HW e SW), le tecnologie sono in rapida convergenza. Rischio di falsi positivi dovuti a AI non evoluta come analista

Attacco attivo

Gli attacchi attivi sono quelli che si basano sull'alterazione dei dati oppure dei flussi con cui i dati sono trasmessi in rete.

Tipologie principali di attacco attivo:

- ▶ **Fabbricazione dell'identità** di un'altra entità (Authentication attacks)
 - Spoofing, Man in the Middle
- ▶ **Interruzione/compromissione di servizi** (Availability attacks)
 - Denial of Service (DoS), Distributed DoS (DDoS)
- ▶ Sfruttamento di **Vulnerabilità** nel software installato (System Integrity and Authentication attacks)
 - Applicazione degli Exploit noti
- ▶ Diffusione di Vulnerabilità intenzionali (System Integrity attacks and auth attacks)
 - Virus, Worms, BOT, Trojan
- ▶ Ingegneria Sociale
 - Phishing

Attacco attivo: Spoofing (fabbricazione)

E' un tipo di attacco informatico dove viene impiegata la falsificazione dell'identità (spoof)
Quando la falsificazione non avviene in campo informatico si parla di social engineering

User account spoofing: usare nome utente e password di un altro utente senza averne il diritto.
Può avvenire utilizzando strumenti come sniffer e password crackers

I password cracker possono essere off-line come John the Ripper <http://www.openwall.com/john/> , oppure on-line, come [Hydra](#)

IP Address spoofing: Si basa sul fatto che la maggior parte dei routers all'interno di una rete controllino solo l'indirizzo IP di destinazione e non quello sorgente. Finalità:

- ▶ superare le tecniche difensive basate sull'autenticazione dell'indirizzo
- ▶ Realizzare attacchi DDoS. Vedi ad esempio "[NTP reflection](#)"

MAC Address forging: il MAC address viene modificato impersonando l'indirizzo della vittima.
Diversi sistemi di autenticazione/autorizzazione sono basati su MAC address.

- ▶ Autenticazione verso DHCP server
- ▶ Sessioni attive su Captive Portal ([session Hijacking](#))

ARP Spoofing / Poisoning: Consiste nell'inviare intenzionalmente e in modo forzato risposte ARP contenenti dati inesatti. In questo modo la tabella ARP di un host conterrà dati alterati. [Ettercap](#) è un tool per attacco di tipo **man-in-the-middle**, basato su ARP poisoning.

Attacco attivo: Denial of Service (DoS)

Causano la perdita dell'utilizzo di una risorsa sovraccaricandola, ma non ne permettono l'accesso all'attaccante. Gli attacchi possono essere

- diretti (l'attaccante interagisce direttamente con la vittima)
- indiretti (l'attaccante sfrutta terze parti).

I principali attacchi sono:

- **FLOODING**
 - **Ping floods:** invio di ICMP echo request in numero maggiore a quelli gestibili dal sistema attaccato; l'aggressore invia un grosso flusso di traffico ICMP echo verso una serie di indirizzi di broadcast attribuendosi come indirizzo sorgente quello della vittima.
 - **TCP SYN Floods:** Funziona se un server alloca delle risorse dopo aver ricevuto un SYN, ma prima di aver ricevuto un messaggio ACK (vedi nmap -sS).
 - **NINVIO DI PACCHETTI MALFORMATI**
 - **Ping di grandi dimensioni (ping of death):** può causare buffer overflow con conseguente blocco del servizio o, nei casi più gravi, crash del sistema.
 - **UDP bombs:** costruiti con valori illegali in certi campi. In certi sistemi operativi la ricezione di pacchetti imprevisti può causare crash
- **ATTACCHI DA PIU' HOST: DDOS (Distributed DoS)**

E' una variante di DoS realizzato utilizzando numerose macchine attaccanti che insieme costituiscono una "botnet" controllate da un'una unica entità, il botmaster

- **NTP reflection** (vedi spoofing)

Attacco attivo: sfruttamento delle vulnerabilità

La vulnerabilità può essere intesa come una componente di un sistema, in corrispondenza alla quale le misure di sicurezza sono assenti, ridotte o compromesse, il che rappresenta un punto debole del sistema e consente a un eventuale aggressore di compromettere il livello di sicurezza dell'intero sistema.

La **vulnerabilità** si presenta ovunque ci sia un difetto di progettazione, codifica, installazione e configurazione del software. Esempi di vulnerabilità: <https://csirt.gov.it>

Un **exploit** è un frammento di codice, una sequenza di comandi, o un insieme di dati, che prendono vantaggio da una **vulnerabilità** per acquisire privilegi di accesso, eseguire codice o creare DoS su di una risorsa.

Patch : Quando viene scoperta una vulnerabilità lo sviluppatore rilascia un aggiornamento del software (**patch**), che porta alla risoluzione della vulnerabilità di sicurezza.

Zero-day è il nome delle vulnerabilità di cui ancora non è stata rilasciata una Patch. Zero-day exploit si riferisce a exploit di vulnerabilità zero-day.

Attacco attivo: gli exploit

I più comuni tipi di exploit prendono vantaggio da:

- ▶ **Buffer overflow (Stack o Heap):** si basa sul fatto che un programma potrebbe non controllare in anticipo la lunghezza dei dati in arrivo, ma si limita a scrivere il loro valore in un buffer di lunghezza prestabilita, confidando che l'utente (o il mittente) non immetta più dati di quanti esso ne possa contenere.
- ▶ **Cross site scripring (XSS)** è una vulnerabilità che affligge siti web dinamici che impiegano un insufficiente controllo dell'input nei form.
- ▶ **Code injection:** Questo exploit sfrutta l'inefficienza dei controlli sui dati ricevuti in input ed inserisce codice maligno (ad esempio all'interno di una query SQL, oppure attraverso Remote File Inclusion)

Spesso, quando gli exploit vengono pubblicati, la vulnerabilità viene eliminata attraverso una Patch che produce una nuova versione del software.

Vedi ad esempio: <https://www.cert.garr.it/alert/security-alerts>

Attacco attivo: Malware

Un qualsiasi software creato allo scopo di causare danni a un computer, ai dati degli utenti del computer, o a un sistema informatico su cui viene eseguito.

<http://it.wikipedia.org/wiki/Malware>

- ▶ **Virus** Sono parti di codice che si diffondono copiandosi all'interno di altri programmi, o in una particolare sezione del disco fisso, in modo da essere eseguiti ogni volta che il file infetto viene aperto.
- ▶ **Worm** questi malware non hanno bisogno di infettare altri file per diffondersi, perché modificano il sistema operativo della macchina ospite in modo da essere eseguiti automaticamente e tentare di replicarsi sfruttando per lo più Internet.
- ▶ **Trojan** software che oltre ad avere delle funzionalità "lecite", utili per indurre l'utente ad utilizzarli, contengono istruzioni dannose che vengono eseguite all'insaputa dell'utilizzatore
- ▶ **Spyware** Software che vengono usati per raccogliere informazioni dal sistema su cui sono installati e per trasmetterle ad un destinatario interessato.

Common Weakness Enumeration (CWE)

CWE <https://cwe.mitre.org/> è una lista di debolezze di sicurezza mantenuta da Mitre Corporation <https://www.mitre.org/> (organizzazione no-profit) in cui ogni tipologia di debolezza viene classificata attribuendo un identificativo numerico univoco.

Esempi delle debolezze più rilevanti nel 2022

- CWE-787 : Out-of-bounds Write
- CWE-79 : Cross-site Scripting
- CWE-89 : SQL Injection
- CWE-20 : Improper Input Validation
- CWE-125 : Out-of-bounds Read
- CWE-78 : OS Command Injection
- CWE-416 : Use After Free
- CWE-22 : Path traversal

Common Vulnerabilities ean Exposures (CVE)

Mitre Corporation gestisce anche il dizionario CVE (Common Vulnerabilities and Exposures) in cui sono classificate ed enumerate le singole vulnerabilità e falle di sicurezza note pubblicamente.

Esistono diversi elenchi delle vulnerabilità più sfruttate nel 2022.

Ad esempio:

<https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-215a>

In questo elenco la vulnerabilità più sfruttata nel 2022 è la [CVE-2018-13379](#)

Denominata «Fortinet FortiOS SSL VPN Path Traversal Vulnerability»

Appartenente al CWE-22

(Improper Limitation of a Pathname to a Restricted Directory - 'Path Traversal')

Organizzazioni come GARR-CERT diffondono regolarmente avvisi riguardo nuove vulnerabilità:

<https://www.cert.garr.it/en/alert-en/security-alerts/archive/listing>

CVSS

Il Common Vulnerability Scoring System (CVSS) è una norma tecnica aperta per valutare la gravità delle vulnerabilità di sicurezza di un sistema informatico.

CVSS assegna un punteggio di gravità alle vulnerabilità, consentendo a chi si occupa di rispondere all'emergenza di stabilire la priorità di risposte e risorse in base al livello di minaccia.

I punteggi vengono calcolati con una formula che dipende da diverse metriche che approssimano la facilità e l'impatto di un exploit. Il punteggio è espresso in una scala da 0 a 10, dove 10 indica il livello di vulnerabilità più grave. (Wikipedia)

Esistono diverse versioni di CVSS per calcolare lo score; le versioni più recenti sono CVSS3 e CVSS2.

Diverse organizzazioni calcolano in CVSS per le nuove vulnerabilità, tra cui [NIST](#)

Nell'esempio della vulnerabilità [CVE-2018-13379](#) NIST ha calcolato:

CVSS3: 9.8/10 CVSS2: 9.1/10

OWASP Top10

Un'altra lista molto nota di debolezze è la [Top10](#) che l'Open Web Application Security Project ([OWASP](#)) stila ogni anno.

La lista del 2021 è la seguente:

A01:2021 Broken Access Control

A02:2021 Cryptographic Failures

A03:2021 Injection

A04:2021 Insecure Design

A05:2021 Security Misconfiguration

A06:2021 Vulnerable and Outdated Components

A07:2021 Identification and Authentication Failures

A08:2021 Software and Data Integrity Failures

A09:2021 Security Logging and Monitoring Failures

A10:2021 Server Side Request Forgery (SSRF)

Attacco attivo: Ingegneria sociale e spam

- ▶ Ingegneria sociale. lo studio del comportamento individuale di una persona al fine di carpire informazioni utili. http://it.wikipedia.org/wiki/Social_engineering
 - ▶ Phishing E' un tipo di ingegneria sociale attraverso la quale un aggressore cerca di ingannare la vittima convincendola a fornire informazioni personali sensibili.
- ▶ Spam <http://it.wikipedia.org/wiki/Spam> Lo spamming è l'invio di messaggi indesiderati (generalmente commerciali). Può essere attuato attraverso qualunque sistema di comunicazione, ma il più usato è Internet, attraverso messaggi di posta elettronica

Il termine SPAM deriva da una scena dei [Monty Python](https://www.youtube.com/watch?v=zLih-WQwBSc) <https://www.youtube.com/watch?v=zLih-WQwBSc>

Metodi e strumenti di difesa

Architettura e politiche di sicurezza

- ▶ Perimetro, superficie d'attacco
- ▶ Firewall Packet filter e Firewall Proxy (system integrity), IDS, IPS (availability)
- ▶ Auditing di sicurezza (utenti, sistemi e reti)

Iniziative di comunità

- ▶ Vulnerability Alerts e scan
- ▶ Computer Emergency Response Team (CERT)
- ▶ Normative

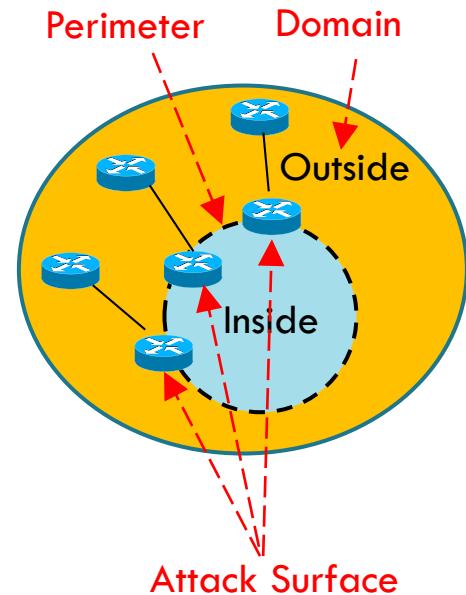
Rinforzo di autenticazione e riservatezza : strumenti crittografici

- ▶ Tools: crittografia simmetrica e asimmetrica, Message Digest, certificati
- ▶ Servizi: autenticazione (authentication, non repudiation), cifratura (data confidentiality), firma digitale (Data Integrity).

Domains, perimeter and attack surface

22

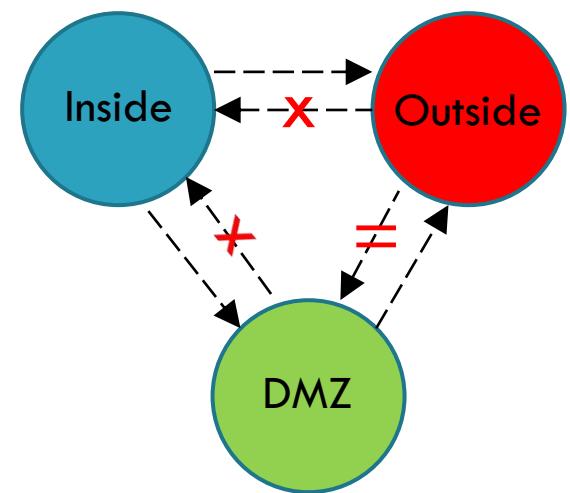
- A **security domain** is a set of entities/resources to be managed as a unique administration area according to a common security policy (security enforcement rules)
- A **security perimeter** is the secured boundary between the external and internal side of a security domain
 - e.g., an internal network and its public facing side, typically the Internet
 - The perimeter can be protected by several security devices
- The **attack surface** of a security domain is the sum of the different points ("attack vectors") where an unauthorized entity ("attacker") can try to enter data to or extract data or do any kind of unauthorized or hostile activity.
 - **Keeping the attack surface as small as possible** is a fundamental basic security measure



Security Domains

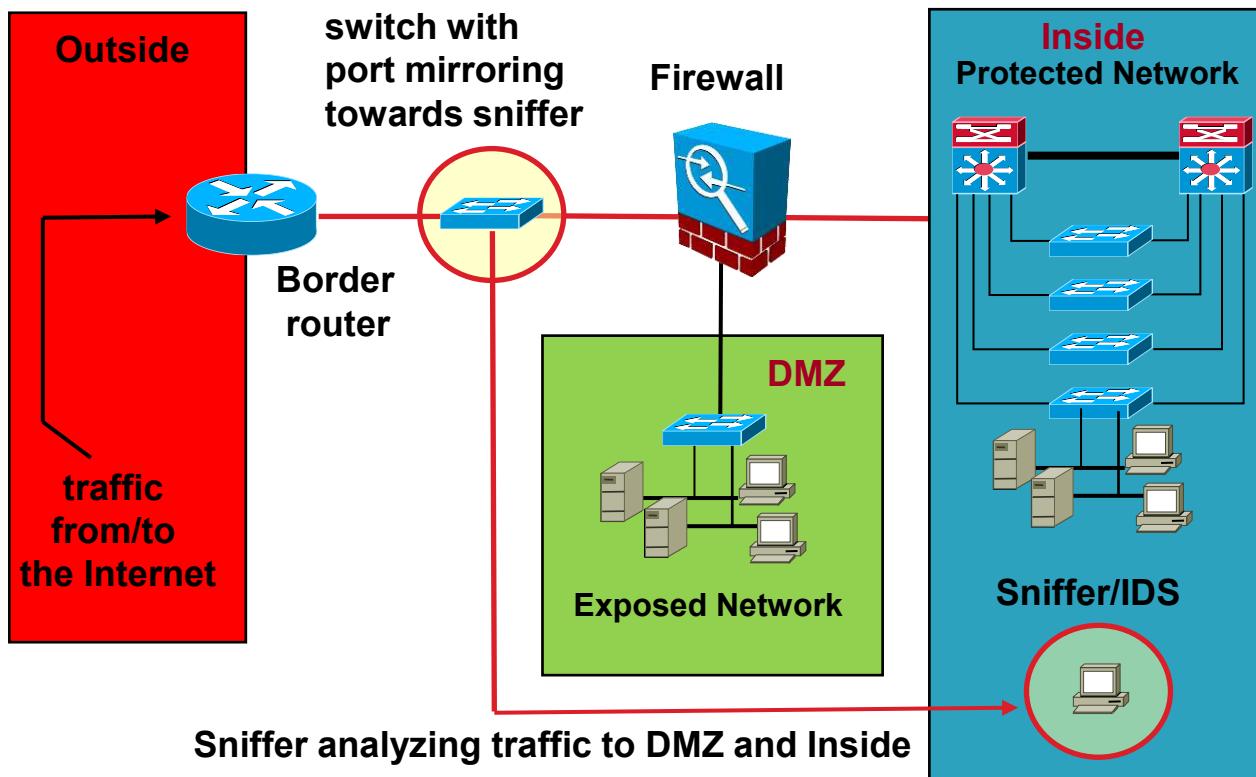
23

- Each security domain is assigned a **degree of trust** or **security level**
- Such degree defines and characterizes its visibility rules (access rights) with respect to the others
 - A domain with a higher degree of trust can have fuller visibility than those with a lower degree
 - Vice versa, visibility is blocked unless specific exceptions (filtering / visibility rules) are defined
 - DMZ and INSIDE have full visibility of OUTSIDE
 - INSIDE has full visibility of DMZ
 - Any other access is not granted



A \dashrightarrow B A is granted access to B
X closed = conditioned

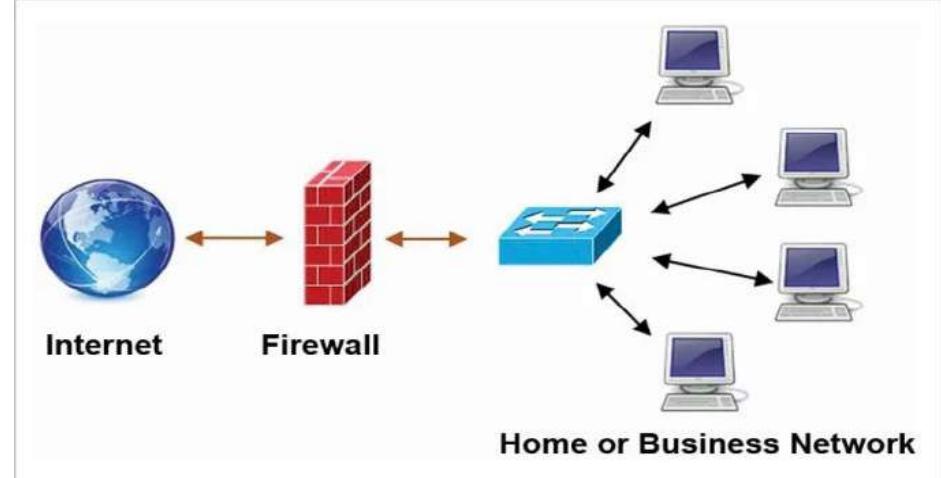
Basic security architecture



- In a common network architecture there are at least three domains:
 - **Outside** (all the world outside - the Internet): trust degree 0
 - **Inside** (the internal organization to be protected and hidden): degree of trust 100
 - **DMZ** (the set of internal machines that expose services outside): degree of trust $0 < x < 100$

Architettura di sicurezza Firewall

Per firewall si intende una entità hardware o software che si pone tra internet e la rete (o l'host) che si vuole proteggere.



Il firewall svolge una funzione di filtro, consentendo il transito solamente alle connessioni ritenute lecite mediante una opportuna “Policy”. Obiettivi principali:

- ▶ Monitorare, limitare, autenticare l'accesso alla rete da proteggere nei confronti di accessi provenienti dall'esterno (Internet).
- ▶ Monitorare, limitare, autenticare l'accesso all'esterno (Internet) da parte dell'utenza interna.

Servizi di sicurezza: system integrity, availability, audit.

I firewall possono essere di 2 tipi:

- **I packet filter:** agiscono ai livelli 3 e 4
- **I proxy:** agiscono a livello applicazione.

Architettura di sicurezza

Firewall a filtro di pacchetti

Questo filtro analizza tutti i pacchetti in transito e applica azioni del tipo permit/deny sulla base di politiche basate sugli indirizzi IP e le porte di provenienza e/o di destinazione.

Obiettivi:

- ▶ Rendere visibili ad internet solamente i servizi di rete destinati ad un accesso pubblico (protezione dei servizi intranet e dei servizi “inconsapevoli”)
- ▶ Bloccare il traffico indesiderato (es: P2P, ...)
- ▶ Strumento per la gestione delle emergenze (bloccare un host ostile o contaminato da virus).

Agisce a livello di pacchetti IP, ma deve leggere anche i primi byte del livello 4 per leggere le porte TCP o UDP.

Può essere realizzato dai Router mediante un modulo aggiuntivo o da HW specifico.

Per Linux esiste il modulo **IPtables** che può essere applicato ad una interfaccia di rete.

Architettura di sicurezza IPtables

Il pacchetto software IPtables consente di applicare ACL per il Packet filtering sulle interfacce di Sistemi Linux. IPtables lavora sulle 3 tabelle filter, nat, mangle sulle quali possiamo creare catene (chains) di regole ACL.

La tabella **Filter** (tabella di default) serve per il packet filter

Le tabelle **NAT** (SNAT e DNAT) servono per le regole di NATting

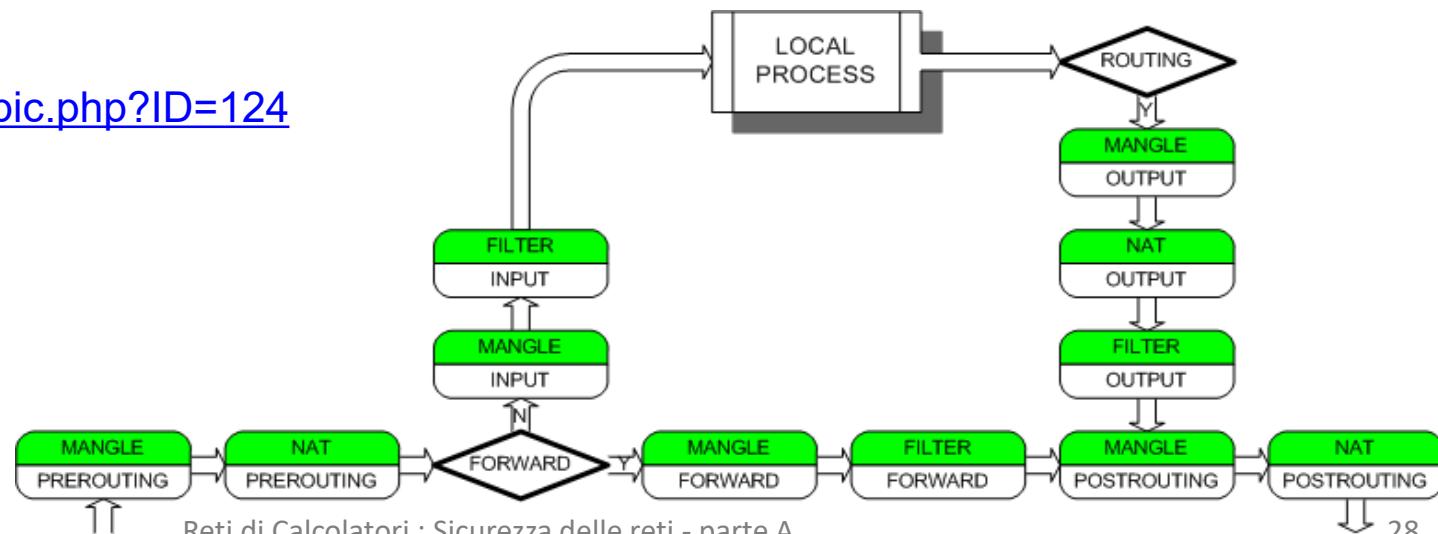
La tabella **Mangle** serve per modificare alcuni parametri nell'header pacchetto

La tabella **Filter** ha 3 catene di Default applicate su una interfaccia di rete:

- ▶ **INPUT** per il processamento dei pacchetti destinati all'host
- ▶ **OUTPUT** per i pacchetti provenienti dall'host
- ▶ **FORWARD** per i pacchetti che devono attraversare il firewall.

Riferimenti:

<http://openskill.info/topic.php?ID=124>



Architettura di sicurezza

Esempi IPtables

```
#accetta i pacchetti entrati di connessioni già stabilite (SYN=0)
```

```
iptables -A INPUT -p ALL -m state --state ESTABLISHED -j ACCEPT
```

```
#accetta i pacchetti entranti dei servizi interni http, 81 (vh1), ssh e 3000 (ntopng)
```

```
iptables -A INPUT -p tcp --dport http -i enp0s3 -j ACCEPT
```

```
iptables -A INPUT -p tcp --dport 81 -i enp0s3 -j ACCEPT
```

```
iptables -A INPUT -p tcp --dport 3000 -i enp0s3 -j ACCEPT
```

```
iptables -A INPUT -p tcp --dport ssh -i enp0s3 -j ACCEPT
```

```
#accetta e registra sul sistema di log i pacchetti entranti verso il servizio telnet
```

```
iptables -A INPUT -p tcp --dport 23 -i enp0s3 -j LOG --log-prefix "TELNET"
```

```
#tutto il resto viene scartato
```

```
iptables --policy INPUT DROP
```

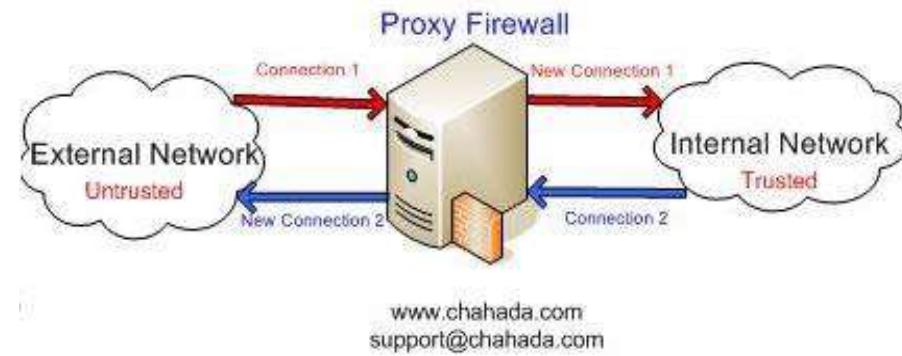
```
iptables -nvL
```

Firewall Basati su Proxy

Il Proxy è un programma applicativo con funzione di tramite tra client e server. In modo implicito o esplicito il client deve rivolgersi al proxy per poter raggiungere il server. Occorre un Proxy specifico per ogni applicazione.

Obiettivi:

- ▶ Mettere in comunicazione client e server che non hanno visibilità diretta (ad esempio se il client è in una rete intranet)
- ▶ Migliorare le prestazioni (es: Web Caching)
- ▶ Servizi di sicurezza
 - Auditing (tracciamento delle attività)
 - Autenticazione e Autorizzazione



Esempi:

- ▶ **Proxy Web** (Vedi Squid <https://it.wikipedia.org/wiki/Squid>)
- ▶ **Proxy SMTP** (MTA con **antiVirus** e **antiSpam** posizionato all'ingresso/uscita della LAN)
- ▶ **socat** (<https://linux.die.net/man/1/socat>) è un tool command-line che consente di connettere due flussi TCP e può quindi essere utilizzato per realizzare un servizio Proxy.

Antivirus

L'Antivirus (AV) è un software atto a prevenire, rilevare ed eventualmente eliminare programmi dannosi. Un AV ha anche una funzione preventiva, impedendo che un virus possa entrare in un sistema ed infettarlo.

Il metodo più utilizzato per individuare virus in un file è attraverso le signatures (firma).

Il programma AV calcola la firma (signature) di un file da analizzare (e.g. Hash MD5 dell'intero file) e lo confronta con le firme di virus noti presenti all'interno di un archivio. Se la firma corrisponde il file è sicuramente un virus.

Esistono anche tecniche euristiche, di solito usate in modo complementare alle firme, che cercano di individuare virus non noti all'AV attraverso la ricerca di pattern sospetti.

Può essere installato

- ▶ sul PC: scan dei dischi dell'host e dei nuovi file salvati
- ▶ sul MailServer: scan delle mail in entrata e in uscita

The best antivirus software for Windows Home User:

<https://www.av-test.org/en/antivirus/home-windows/>

Tecniche Antispam

Link utili: <http://it.wikipedia.org/wiki/Spam>

Black list <https://www.cert.garr.it/it/documentazione/articoli-tecnici/30-blacklist>

Lista di server classificati spammers che viene attivata sul mail server rifiutando mail che provengono da host inclusi in questa lista. L'amministratore del mailserver può costruire manualmente una propria lista o può avvalersi di servizi in Internet che distribuiscono automaticamente le liste.

Gray-List: Si basano sul fatto che i mailer usati dagli spammer generalmente tentano l'invio di una email una sola volta: Il GrayListing consiste nel rigetto della ricezione della mail al primo tentativo, che verrà accettata ad un successivo tentativo, dopo un tempo stabilito (tipicamente 300 sec.)

White List. Liste di mittenti “Fidati” su cui non vengono effettuati controlli antispam. Include gli host accettati da Gray-list e host inseriti manualmente dall'amministratore.

Filtri Bayesiani: Sono filtri che cercano di classificare le mail in arrivo assegnando un punteggio numerico a frasi o modelli che si presentano nel messaggio. Ogni messaggio riceve quindi un punteggio compressivo (tra 0 e 1) che, dopo aver stabilito una soglia, ci consente di classificare il messaggio. Il filtro richiede un addestramento con mail spam e no-spam con cui viene creato un database di riferimento.

Esempi:

- ▶ Spamassassin <http://spamassassin.apache.org/> (liste White e Black, filtri Bayesiani)

IDS (Intrusion Detection System)

IDS è un dispositivo software/Hardware per identificare accessi non autorizzati a host o LAN.

L'IDS generalmente si appoggia su un Data-Base per memorizzare le regole utilizzate per individuare le violazioni di sicurezza.

Gli IDS sono classificabili nel seguente modo:

Host IDS (HIDS): analizzano file di log e file system sull'Host.

TRIPWIRE è un esempio di HIDS. Si basa sulla differenza tra lo stato analizzato ed uno stato iniziale.

Network IDS (NIDS): analizzano il traffico di rete.

SNORT è un esempio di NIDS che può funzionare anche come sniffer / packet logger.

Quando un IDS rileva una intrusione invia una notifica all'amministratore via e-mail o con un messaggio alla console (continuous monitoring e auditing).

Intrusion Prevention System (IPS)

Gli IPS sono un'estensione degli strumenti di IDS: quando rilevano un tentativo di intrusione sono abilitati a bloccare gli accessi considerati pericolosi.

IPS può mandare un allarme (come un IDS), ma anche interagire con un firewall per eliminare pacchetti malevoli, resettare le connessioni e/o bloccare il traffico da un indirizzo IP attaccante.

Strumenti utili: **fail2ban** <http://guide.debianizzati.org/index.php/Fail2ban>

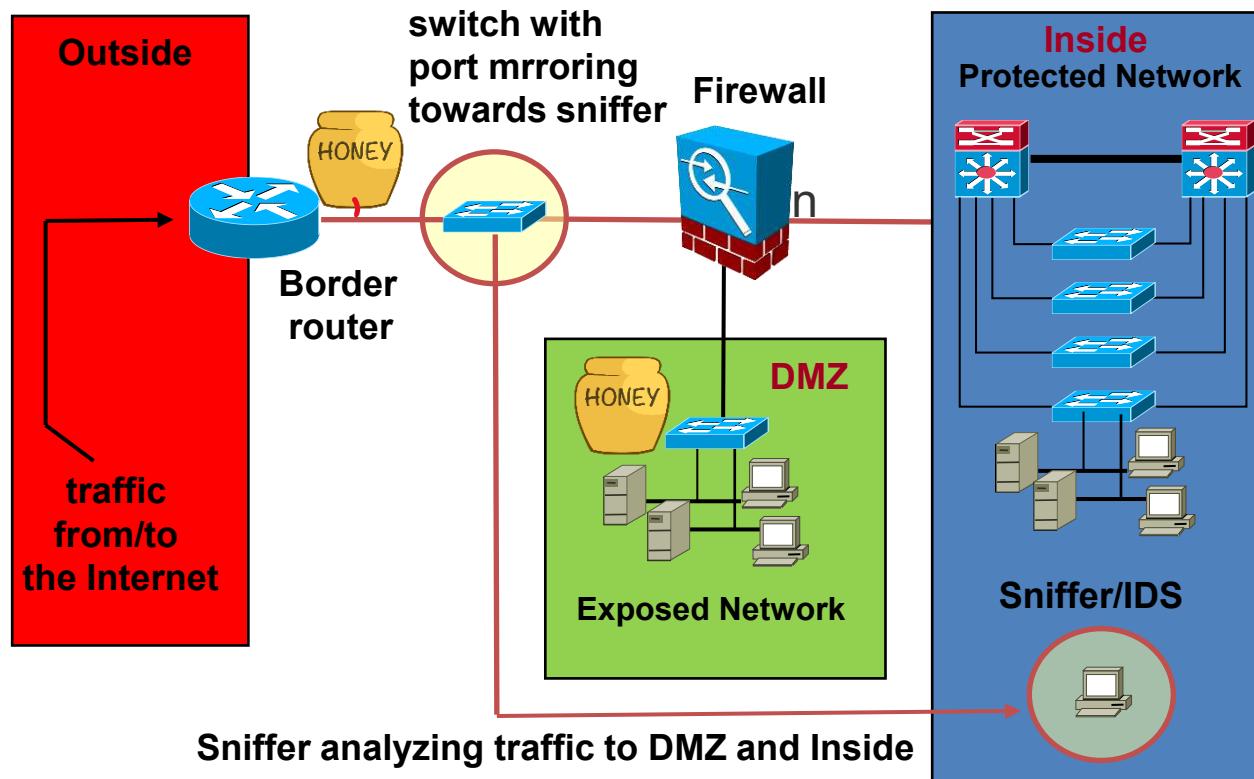
Fail2ban è pensato per prevenire attacchi “brute force” via ssh bloccando temporaneamente gli indirizzi IP che provano a violare la sicurezza di un sistema.

Il programma effettua il parsing di alcuni file di log che contengono informazioni relative ad accessi falliti. Se il numero di accessi falliti supera una certa soglia l'indirizzo IP del client viene bloccato attraverso una regola di iptables.

Architettura di rete Honeypot

Un honeypot è un sistema o componente hardware o software usato come "trappola" o "esca" a fini di protezione contro gli attacchi. Consiste in un computer o un sito che sembra essere parte della rete, ma che in realtà è ben isolato e non ha contenuti sensibili o critici.

Consente di rilevare attività malevole senza coinvolgere la rete interna.



Auditing di sicurezza informatica

Un audit di sicurezza informatica è un'analisi sistematica con l'obiettivo di verificare i controlli di sicurezza, politiche e procedure in atto e convalidare il loro efficace funzionamento in base al rischio.

Gli elementi oggetto di valutazione di un audit di sicurezza informatica sono:

- **personale aziendale**, ovvero le modalità utilizzate dai dipendenti per raccogliere, condividere e archiviare informazioni sensibili.
- I componenti del **sistema informatico** e ambiente in cui è ospitato
- **applicazioni e software** (incluse le patch di sicurezza messe a punto dagli amministratori di sistema)
- **vulnerabilità della rete** interna ed esterna

Passaggi principali dell'auditing:

- Definizione degli asset: elenco dettagliato di risorse, dati sensibili, apparecchiature informatiche e relativa valutazione del rischio informatico
 - Valutazione del personale
 - Monitoraggio dei sistemi e delle reti
 - Identificazione delle vulnerabilità
 - Risposte al rischio, aumento delle protezioni

Auditing del personale

La sicurezza dei sistemi e delle reti dipende anche dal comportamento degli utenti e degli amministratori di sistema. Occorre annotare e registrare quali dipendenti abbiano accesso alle informazioni sensibili e quanti di loro siano preparati in maniera adeguata.

Il rischio principale riguarda il furto delle credenziali per l'accesso ai sistemi aziendali.

La password è una delle forme di identificazione più semplici ed utilizzate ed è quindi uno dei principali bersagli degli attacchi.

I metodi più utilizzati sono:

- 1) Intercettazione. L'utilizzo di un canale non cifrato consente la cattura delle password sulla rete.
- 2) Furto. Alcuni utenti tendono a scriverla su un supporto magnetico per non dimenticarla.
- 3) Tentativi di indovinare la password (password cracker basati su dizionari)
- 4) Phishing.

I risultati dell'audit sul personale vengono utilizzati per programmare un **piano di formazione del personale**

Auditing di sistema

L'audit di sicurezza informatica richiede la definizione dell'elenco dei sistemi, la loro classificazione in base al rischio informatico e l'attribuzione del ruolo di amministratore.

L'amministratore di sistema deve definire un programma di audit attraverso il quale viene definita la gestione e l'analisi degli eventi, con i seguenti obiettivi:

- **(Early) warning:** Individuare rapidamente eventuali attacchi in corso.
- **Trouble-shooting:** mantenere uno storico degli eventi per tracciare le attività.

rsyslog è lo strumento base per l'audit di sistema in ambiente Linux.

Consente ai processi interni di generare eventi classificati in categorie (KERN, USER, MAIL, DAEMON, AUTH, LPR, CRON, LOCAL0-7) e priorità (EMERG, ALERT, CRIT, ERR, WARNING, NOTICE, INFO, DEBUG). Per ogni evento è possibile definire una azione come scrivere su file (tipicamente nella directory /var/log) , inviare mail o attivare script.

Auditing di rete

Audit della rete

Consiste nella raccolta e analisi sistematica del traffico di rete e per la rilevazione in tempo reale di minacce provenienti dalla rete. Strumenti utili sono i **Network Monitor** come [nTop](#), dotato di una console web, vedi figura.

nTop

Flows Hosts Devices Interfaces Search Host

All Hosts

10 Filter Hosts IP Version

IP Address	Location	Flows	Alerts	Name	Seen Since	ASN	Breakdown	Throughput	Traffic
224.0.0.5	Remote Host	7	0	224.0.0.5	2 min, 7 sec		Rcvd	1.07 kbit/s	8.46 KB
192.168.61.1	Remote Host	1	0	192.168.61.1	2 min, 7 sec		Sent	124.77 bit/s	1014 Bytes
192.168.0.254	Local Host	6	0	netlab0	3 days, 5 h, 58 min, 58 sec		Sent Rcvd	156.76 bit/s	22.7 MB
192.168.0.105	Local Host	7	0	netlab5 [GIOVANNI-LAPTOP]	3 days, 5 h, 39 min, 44 sec		Sent Rcvd	156.76 bit/s	25.87 MB
192.168.0.104	Local Host	10	0	netlab4	3 days, 5 h, 39 min, 46 sec		Sent Rcvd	156.76 bit/s	24.8 MB
192.168.0.103	Local Host	11	0	netlab3	3 days, 5 h, 39 min, 49 sec		Sent Rcvd	156.76 bit/s	25.04 MB
192.168.0.102	Local Host	8	0	netlab2	3 days, 5 h, 39 min, 57 sec		Sent Rcvd	156.76 bit/s	26.08 MB
24/11/2023				Reti di Calcolatori : Sicurezza delle reti - parte A					
192.168.0.101		7	0	netlab1	3 days, 5 h, 58 min		Sent Rcvd	156.76 bit/s	25.52 MB

Auditing di rete

Identificazione delle vulnerabilità

Un **vulnerability scanner** è un programma progettato per ricercare e mappare le debolezze di un singolo computer o degli host di una rete.

Identifica le vulnerabilità dovute a software con bugs o non aggiornato, configurazioni errate all'interno di servizi applicativi, web server, firewall router, ecc

Scanner più utilizzati: Nessus e Qualys (commerciale) - openVAS (opensource).

Funzionamento:

- 1 ricerca host attivi
- 2 port scanning per ogni host
- 3 ricerca vulnerabilità tramite test non invasivi
- 4 identificazione vulnerabilità tramite confronto con database
- 5 Generazione di un report

The screenshot shows the Nessus web interface with the following details:

- Host Details:** IP: 192.153.11.16, MAC: 7c:5cf8:cce:9:80, OS: Linux Kernel 2.6, Start: August 26 at 12:22 PM, End: August 26 at 12:36 PM, Elapsed: 13 minutes, KB: Download.
- Vulnerabilities:** A pie chart indicates 25 vulnerabilities: Medium (orange) and Info (blue).
- Scan Results Table:** A table listing 15 vulnerabilities found on the host, categorized by severity (Medium or Info) and plugin family.

Severity	Plugin Name	Plugin Family	Count
MEDIUM	Apache Server ETag Header I...	Web Servers	1
MEDIUM	PHP expose_php Information...	Web Servers	1
INFO	RPC Services Enumeration	Service detection	4
INFO	Nessus SYN scanner	Port scanners	2
INFO	Apache Banner Linux Distrib...	Web Servers	1
INFO	Backported Security Patch D...	General	1
INFO	Common Platform Enumeratio...	General	1
INFO	Device Type	General	1
INFO	Ethernet Card Manufacture...	Misc.	1
INFO	HTTP Methods Allowed (per...	Web Servers	1
INFO	HTTP Server Type and Version	Web Servers	1
INFO	HyperText Transfer Protocol (...)	Web Servers	1

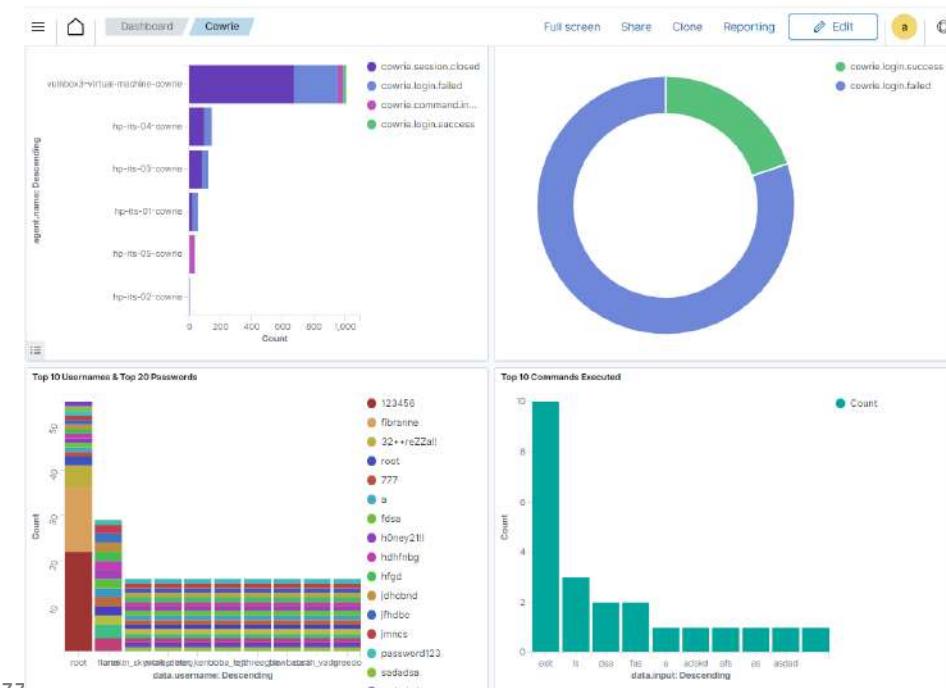
Cyber Threat Intelligence (CTI)

La Threat Intelligence, consiste nell'attività di raccolta di informazioni riguardo potenziali minacce provenienti da varie fonti; i dati vengono memorizzati, correlati e processati da processi automatici di Threat intelligence che possono generare allarmi (alert) o attivare delle azioni (Active Response).

Esempi di sorgenti delle informazioni possono essere EDR (Endpoint Detection and Response) ovvero agenti su dispositivi quali PC, , oppure Honeypot , network monitor e NIDS (Network Intrusion Detection System).

Il collettore e analizzatore dei dati è tipicamente un **SIEM** (Security Information and Event Management).

L'organizzazione delle fonti e la gestione del SIEM può essere operata da un centro operativo denominato **SOC** (Security Operation Center).





UNIVERSITÀ
DI PARMA

DIPARTIMENTO DI SCIENZE MATEMATICHE, FISICHE ED INFORMATICHE
Corso di Laurea in Informatica

Sicurezza delle reti – Parte B

Crittografia applicata e OpenSSL

RETI DI CALCOLATORI - a.a. 2023/2024

Roberto Alfieri

La sicurezza delle reti: sommario

PARTE A

- ▶ I servizi di sicurezza
- ▶ Metodi e strumenti di attacco
- ▶ Metodi e strumenti di difesa

PARTE B

- ▶ Crittografia applicata e OpenSSL

PARTE C

- ▶ Protocolli di autenticazione
- ▶ IPsec
- ▶ VPN
- ▶ Sicurezza delle reti WiFi

Crittografia

La crittografia è lo studio dei metodi per rendere un messaggio “ofuscato” in modo da renderlo comprensibile solo a persone a cui il messaggio è destinato (servizio di sicurezza **Confidenzialità**).

L'algoritmo che esegue la cifratura o decifratura è detto **Cipher**.

La **cifratura E** (Encryption) si applica ad un **messaggio in chiaro P (Plaintext)** per ottenere un **testo cifrato C (Ciphertext)**. $C=E(P)$

La **decifratura D** (decryption) si applica ad un **messaggio cifrato C** per ottenere il **testo in chiaro P**. $P=D(C)$

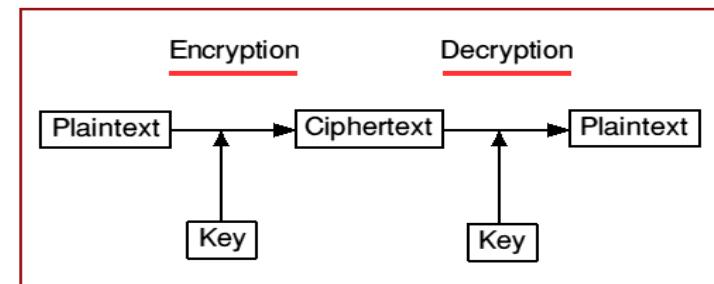
La **crittanalisi** è lo studio dei metodi per ottenere il significato di un testo cifrato violando la tecnica di cifratura del Cipher per verificarne la robustezza.

La robustezza di un Cipher non si basa sulla segretezza degli algoritmi di cifratura, ma sull'utilizzo di **chiavi segrete k** utilizzate nella fase di cifratura e decifratura.

Se gli algoritmi utilizzano la stessa chiave per cifrare e decifrare sono detti a **chiave simmetrica**, altrimenti sono detti a **chiave pubblica**.

A chiave simmetrica $C=E(P,k)$ $P=D(C,k)$

A chiave pubblica $C=E(P,k_1)$ $P=D(C,k_2)$



SSL/TLS

SSL (Secure Socket Layer) protocollo sviluppato originariamente dalla Netscape per fornire autenticazione e privacy in una connessione tra un client e un server mediante l'uso di diverse tecnologie crittografiche.

Le principali tecnologie sono:

- Algoritmi crittografici (a chiave simmetrica o a chiave pubblica)
- Message Digest e firma digitale
- Certificati e Certification Authority
- Security Socket Layer (SSL) e Transport Layer Security (TLS)

Versioni di SSL e TLS:

- ▶ SSL v2.0 (1995): prima versione implementata da Netscape (deprecata dal 2011)
- ▶ SSL v3.0 (1996): revisione della precedente (deprecata dal 2015)
- ▶ TLS 1.0 (1999): RFC 2246. Revisione di SSL 3.0 (deprecata dal 2021)
- ▶ TLS 1.1 (2006): RFC 4346. (deprecata dal 2021)
- ▶ **TLS 1.2 (2008): RFC 5246.**
- ▶ **TLS 1.3 (2018): RFC 8446** (attuale)

Vedi “SSL/TLS Strong Encryption: An Introduction” https://httpd.apache.org/docs/2.4/ssl/ssl_intro.html

Implementazioni del protocollo SSL/TLS

OPENSSL (<http://www.openssl.org/>)

E' una libreria Open Source basata sulla libreria SSLeay sviluppata da Eric Young e Tim Hudson. Fino alla versione openSSL1.1 si applica la licenza originale SSLeay più restrittiva e incompatibile rispetto a GPL.

Dalla versione openSSL3.0 si applica la Apache License v2 (<https://www.openssl.org/source/license>)

Fornisce:

- **Un ToolKit** per l'utilizzo a linea di comando delle API.
- **La Libreria SSL** per la programmazione di canali cifrati con SSL/TLS.
- **La Libreria Crypto** per la programmazione di diversi algoritmi crittografici tra cui la cifratura a chiave simmetrica, a chiave pubblica, certificati e funzioni di Hash.

GnuTLS (<http://www.gnu.org/>)

Creata per consentire alle applicazioni del Progetto GNU di usare una libreria compatibile con la GPL Licenza GPL e LGPL

Fornisce API di programmazione (C/C++ Python PHP) e alcune utilities di supporto.

OpenSSL ToolKit

Il Toolkit consente l'utilizzo a linea di comando di tutte le operazioni crittografiche della libreria.

Sintassi:

```
openssl comando [opzioni dei comandi]
```

Ad esempio per vedere la versione di openssl in uso:

```
openssl version
```

Documentazione:

man <comando>

<https://www.openssl.org/docs/manpages.html>

wiki

https://wiki.openssl.org/index.php/Main_Page

OpenSSL Command-Line HOWTO: <http://www.madboa.com/geek/openssl/>

Principali Comandi:

Cipher simmetrici: enc (base64 bf des des3 idea rc4 rc5)

Crittografia a chiave pubblica: rsa rsautl gendsa genrsa

Certificati X.509: x.509 (gestione dati dei certificati), ca (per utilizzare le funzioni di Ceritification Authority), req (richiesta di certificati), verify (verifica di certificati), crl (Certificate Revocation List), crl2pkcs7 pkcs7 (conversione tra diversi formati di certificati),

Message digest : dgst (md2, md5, rmd160, sha, sha1)

Altro: ciphers -v (elenco ciphers supportati), speed (benchmarking), prime (numeri primi) e rand, password, smime, s_client, s_server

Comandi OpenSSL: base64

Viene utilizzata per codificare una sequenza binaria in una sequenza di caratteri ASCII.

La sequenza binaria è suddivisa in blocchi di 6 bit ciascuno dei quali viene trasformato in un carattere ASCII mediante un alfabeto dei 64 caratteri [a-zA-Z0-9./]

Vedi <https://wiki.openssl.org/index.php/Base64>

```
$ openssl base64 -e -in toencrypt.txt -out toencrypt.b64 (-e default)
$ openssl base64 -d -in toencrypt.b64 -out toencrypt.txt

$echo "encode me" | openssl base64 > encrypted.b64
$openssl base64 -d -in encrypted.b64
```

Comandi OpenSSL: prime e rand

OpenSSL contiene routine per trattare numeri primi e numeri random (poiché questi servono per le tecniche crittografiche).

Vedi: https://wiki.openssl.org/index.php/Random_Numbers

Esempi:

```
# Testare se uno o più numeri sono primi:  
$ openssl prime 119054759245460753  
> 1A6F7AC39A53511 is not prime  
$ for N in $(seq 100 300); do openssl prime $N; done  
  
# Scrive un numero random di 128 byte su stdout codificato in base64  
$ openssl rand -base64 128  
  
# Scrive un numero random di 32 bits (4 bytes) su un file  
$ openssl rand -out random-data.bin 4
```

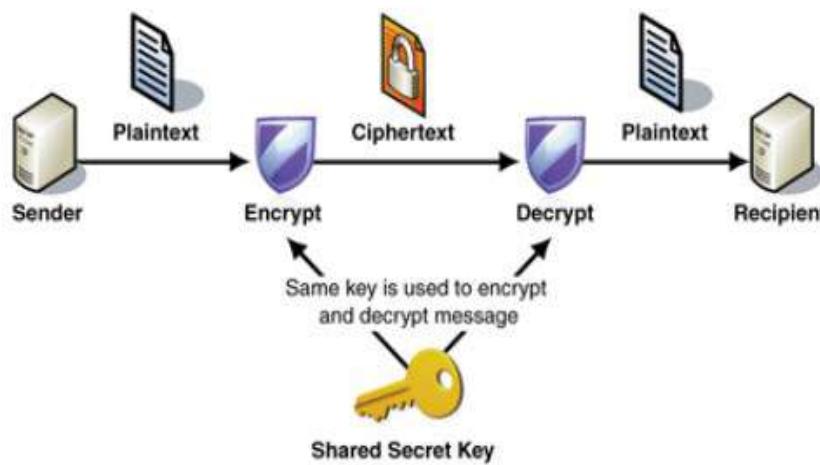
Esempio: programma *in prime.c* che usa le funzioni BN (BigNum) della libreria Crypto per generare numeri primi random da 1024 bit

<http://didattica-linux.unipr.it/~roberto.alfieri@unipr.it/matdid/RETI/security/prime.c>

Crittografia a chiave Simmetrica

E' una tecnica crittografica a chiave condivisa che consente di ottenere la cifratura di un messaggio $P \rightarrow C = E_K(P)$ in modo che la stessa chiave possa essere utilizzata per decifrarlo $C \rightarrow P = D_K(C)$

Gli algoritmi E e D (Cipher) sono noti. La complessità della cifratura è data da K la cui lunghezza determina l'ampiezza dello spazio delle possibili codifiche di P e di conseguenza la robustezza della cifratura.

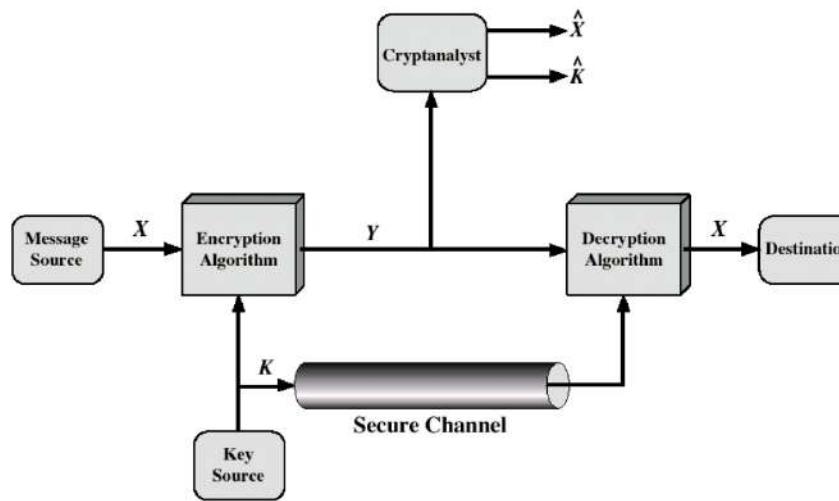


Algoritmi di cifratura molto performanti e semplici da implementare.

Crittografia a chiave Simmetrica: criticità

1) E' necessario un canale (o un metodo) sicuro per **distribuire la chiave tra le due parti.**

Generalmente lo scambio avviene attraverso algoritmi a chiave pubblica, più complessi sia da implementare che da eseguire, ma che permettono questo scambio in modo sicuro.



2) Il cipher può comunque essere compromesso con **un attacco a forza bruta**, ovvero testando tutte le possibili chiavi del cipher.

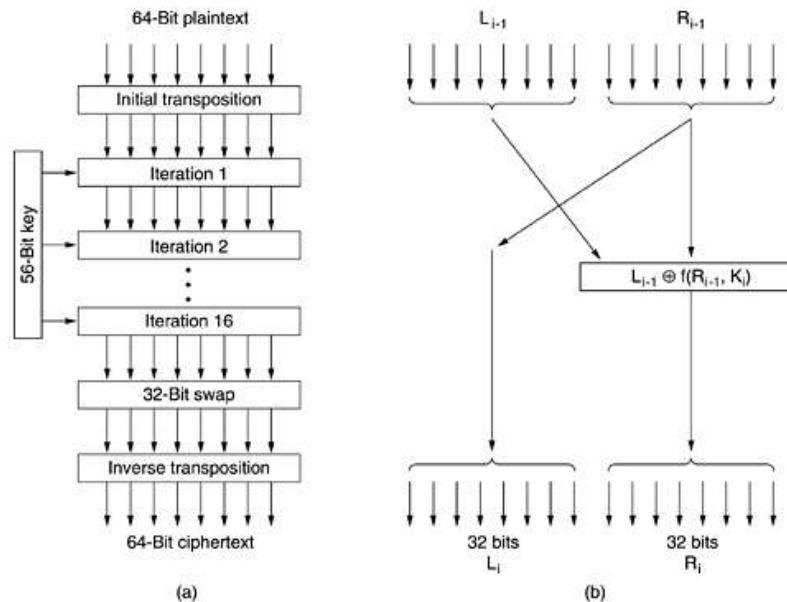
La protezione da questo tipo di attacco si ottiene aumentando la lunghezza del **numero di bit della chiave** e conseguentemente lo spazio delle possibili chiavi da testare, in modo che il costo per forzare il cipher ecceda il valore del dato cifrato.

Cipher Simmetrici

OpenSSL supporta tutti i principali Cipher Simmetrici, tra cui:

DES: (Data Encryption Standard) http://it.wikipedia.org/wiki/Data_Encryption_Standard

sviluppato dall'IBM e adottato dal governo USA nel 1977. Lavora **su blocchi** del messaggio di 64 bit, con chiave di 64 bit di cui 8 sono di parità dispari, pertanto la chiave è di 56 bit.



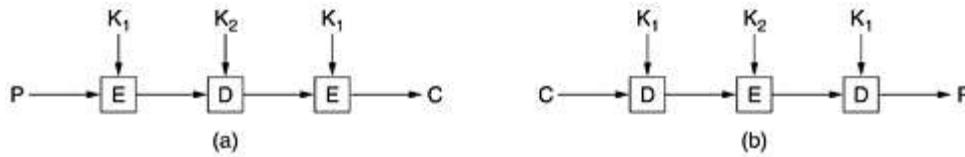
(a) struttura generale

(b) dettaglio di una iterazione

La lunghezza della chiave è troppo breve e l'algoritmo può essere forzato in poche ore.

Cipher Simmetrici

TripleDES: E' l'algoritmo DES applicato 3 volte. Può utilizzare 1, 2 o 3 chiavi DES. Esempio con 2 chiavi EDE (K1 e K2, totale 112 bit):



AES: Nel 1997 il NIST (National Institute of Standard and Technology) promosse un concorso per la realizzazione di un nuovo Cipher Simmetrico che sarebbe diventato uno standard del Governo USA con il nome di AES (Advanced Encryption Standard).

L'algoritmo vincitore fu Rijndael (V. Rijmen e J. Daemen).

L'algoritmo supporta chiavi da 128, 192 o 256 bit su blocchi da 128 bit.

RC4: (Rivest 1987) è un algoritmo utilizzato ampiamente in protocolli quali il WEP (WiFi) e SSL.

Comandi OpenSSL: enc

Questo comando serve per cifrare/decifrare utilizzando uno tra i cipher simmetrici supportati che sono circa 50, tra cui

-des -des3 (3 chiavi EDE) -aes128 -aes192 -aes256 -rc4

Vedi la pagina Wiki: <https://wiki.openssl.org/index.php/Enc>

Esempio: DES3

```
$ openssl enc -e -a -des3 -in myfile           -out myfile.des3
$ openssl enc -d -a -des3 -in myfile.des3 -out myfile
( -a      determina l'output codificato BASE64 )
```

Message Digest

Il Message Digest (MD)

è una sequenza di bit di lunghezza limitata e fissa associata ad un messaggio (P) e ne rappresenta la “firma” (o “impronta”).

Il MD **non è invertibile**, ovvero non è possibile risalire al messaggio originale.

Se il massaggio cambia anche di un solo bit il MD diventa completamente diverso.

Il MD viene calcolato applicando al messaggio una **funzione di Hash**: $MD=H(P)$

L'algoritmo deve essere “**Collision Free**”, ovvero deve evitare (o minimizzare) la possibilità che 2 messaggi generino lo stesso MD. Per questo il MD non può essere troppo breve.

Applicazioni Principali:

- Verificare l'integrità di messaggi o file. Il messaggio P viene spedito assieme al MD; chi li riceve ricalcola i MD e lo compara con quello ricevuto.
- Verifica della password. Viene memorizzato il MD della password in chiaro $MD=H(\text{clear_passw})$. Per verificare la password bisogna confrontare $H(\text{proposed_clear_passw})$ con MD.

Comandi OpenSSL: dgst

Principali Digest:

MD5 è un algoritmo di Hashing a 128 bit realizzato da R. Rivest nel 1991 (RFC1321).

RIPEMD è una famiglia di algoritmi sviluppati come alternativa Europea a MD5

Il più importante e' **ripemd160** <http://en.wikipedia.org/wiki/RIPEMD>

SHA (Sechure Hash Algorithm) indica una famiglia di 5 diverse funzioni sviluppate da NSA come standard Federale del governo USA:

- **sha1** (160 bit)
- sha2 (**sha224** (224 bit) – **sha256** (256bit) – **sha384** (384 bit) - **sha512** (512bit))

Il comando **dgst** serve ad applicare una tra le funzioni di Hash supportate (man dgst):

Esempi:

```
openssl dgst -md5 -hex myfile  (-md5 default, vedi anche md5sum)  
openssl sha1 -hex myfile
```

HMAC

Se 2 parti A e B condividono una chiave simmetrica possono autenticare i messaggi che si scambiano utilizzando lo schema HMAC (Hashed Message Authentication Code) che è una funzione di Hash applicata al Messaggio e alla Chiave condivisa:

Firma= HMAC (K,M)

- ▶ Il mittente che deve inviare M calcola HMAC(K,M) che invia assieme ad M
- ▶ Il destinatario riceve M e HMAC(K,M), quindi verifica la firma ricalcolando l'HMAC

Questo schema è utilizzato in diversi protocolli crittografici, tra cui IPsec.

```
echo "foo" | openssl dgst -sha256 -hmac 123  
#-hmac key : create a hashed MAC using "key".
```

Password

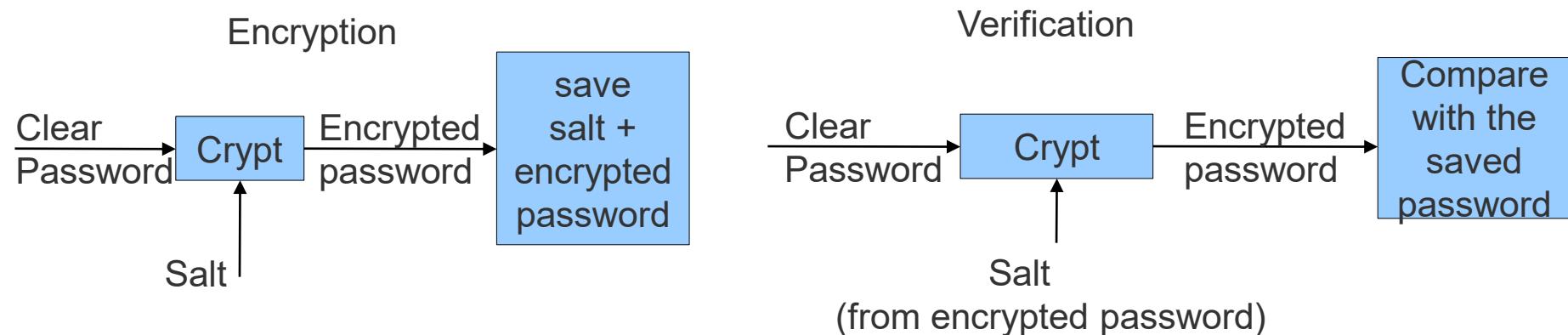
La password è una sequenza alfanumerica che l'utente deve inserire per accedere ad una risorsa protetta. E' quindi un segreto condiviso tra utente e risorsa, che normalmente la risorsa memorizza in formato cifrato.

La cifratura non e' invertibile, quindi per la verifica occorre cifrare la password da verificare e confrontarla con la password cifrata memorizzata dalla risorsa.

La cifratura avviene utilizzando cipher simmetrici, o funzioni di Hash.

Spesso si utilizza una stringa casuale, detta «salt», per complicare gli attacchi al dizionario.

Il comando **passwd** per generare le password Unix/Linux si basa sulla funzione **crypt()**.



Password con crypt()

crypt() in glibc:

Crypt itera 25 volte l'algoritmo DES per cifrare un messaggio costante (tipicamente una sequenza di 0) utilizzando una chiave simmetrica derivata dalla password in chiaro inserita dall'utente + una stringa casuale nota di 12 bit "salt". Il salt viene scritto in chiaro all'inizio della password cifrata con codifica base64 (2 caratteri).

La modifica di DES (25 iterazioni + salt) rende la cifratura non invertibile.

La verifica della password consiste nel confronto tra i messaggi codificati:

crypt viene ripetuto con la password da verificare e il salt prelevato dalla password cifrata.

Il fatto che la cifratura non è invertibile consente di utilizzare gli algoritmi di Hashing come tecniche alternative di codifica, implementate in glibc2.

crypt in glibc2: aggiunge la possibilità di cifrare con MD5.

L'hash MD5 è di 128 bit

L'output è una stringa composta al più da 34 byte di cui:

-la prima parte è il salt: \$1\$<max 8 char>\$

-la seconda parte è una sequenza di 22 caratteri (128 bit di MD5 / 6 di BASE64) contenenti MD5(<salt><clear_password>)

Comandi OpenSSL: passwd

Con openssl è possibile generare le password utilizzando il crypt di glibc (default) oppure un Message Digest:
MD5 con l'opzione -1, SHA-256 opzione -5, SHA-512 opzione -6

Esempi:

```
$ openssl passwd mysecret
```

```
QvpTKPjqpBD9.
```

(i primi 2 caratteri Qv sono il salt generato random e utilizzato assieme a mysecret per derivare la chiave di cifratura)

```
$ openssl passwd -salt Qv mysecret
```

 (riproduce la stessa cifratura)

```
$ openssl passwd -1 mysecret
```

```
$1$NvOCDeMO$5keqaA/5i/07kpEXArm0L/
```

(NvOCDeMO è il salt casuale, le restanti 22 cifre BASE64 sono l'hash MD5 di 128 bit)

Le password di Apache:

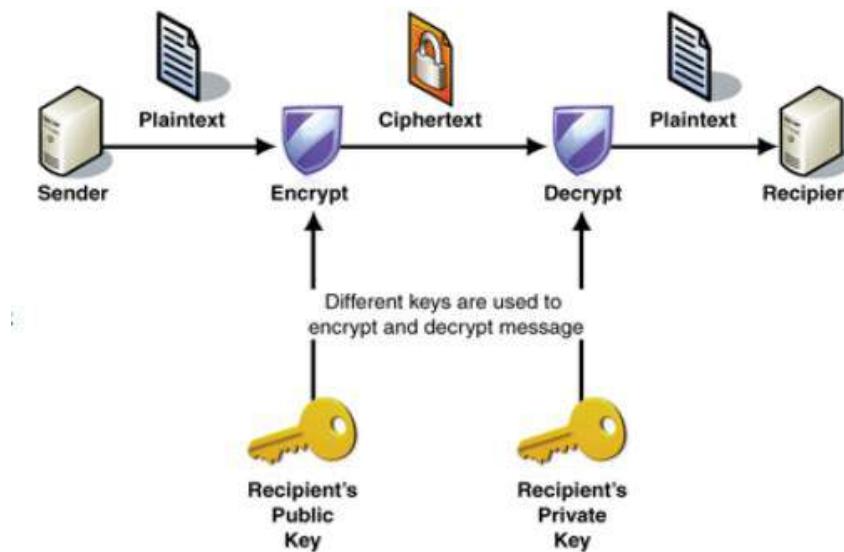
https://httpd.apache.org/docs/2.4/misc/password_encryptions.html

Algoritmi a chiave pubblica

La cifratura a chiave simmetrica ha una grave debolezza nella condivisione della chiave: il trasferimento della chiave la espone ad intercettazione.

Diffie ed Hellmann (Stanford) nel 1976 proposero una tecnica nuova di crittografia, basata sull'aritmetica modulare, in cui vengono utilizzate due chiavi K_e e K_d distinte per la codifica $C=K_e(P)$ la decodifica $P= K_d(C)$.

Assegnando ad una chiave il ruolo di "chiave privata" e all'altra il ruolo di "chiave pubblica" si supera la debolezza della chiave condivisa.



Algoritmi a chiave pubblica

Una importante proprietà di questo algoritmo è: $K_d(K_e(P)) = K_e(K_d(P)) = P$

Questo consente di poter applicare le chiavi in 2 modi diversi ottenendo 2 diverse funzioni:

1) Applicando prima la chiave pubblica si ottiene la **Privacy (crypt/decrypt)**.

A deve inviare un messaggio P riservato a B su di un canale insicuro.

B possiede una coppia di chiavi asimmetriche B_e (pubblica) e B_d (privata).

A cifra P con la chiave pubblica di B: $C=B_e(P)$. Solo B può decifrarlo $P=B_d(C)$

2) Applicando prima la chiave privata si ottiene l'**Autenticazione e Integrità (sign/verify)**

A deve inviare un messaggio P attraverso un canale insicuro. Tutti lo possono leggere, ma chi lo riceve deve essere sicuro che è stato inviato da A.

A possiede una coppia di chiavi asimmetriche A_e (pubblica) e A_d (privata).

A cifra P con la propria chiave privata: $C=A_d(P)$. Chiunque può applicare la chiave pubblica di A: $P=A_e(C)$. La decifratura funziona solo se P è stato cifrato da A.

RSA

Rivest, Shamir e Adleman del MIT implementarono nel 1978 l'algoritmo che ha preso il loro nome (RSA) e che è attualmente il più utilizzato nelle applicazioni crittografiche a chiave pubblica.

- 1) si scelgono 2 numeri primi casuali p e q , sufficientemente grandi.
- 2) si calcolano $m = pq$ (chiamato modulo) $z = (p-1)(q-1)$
- 3) si sceglie un numero $e < (p-1)(q-1)$ coprimo con $(p-1)(q-1)$ #(senza divisori comuni)
- 4) si calcola un numero d tale che $e^d \equiv 1 \pmod{z}$ (il resto della divisione $(e^d)/z$ deve essere 1)

$C = P^e \pmod{m}$ (e, m) è la chiave pubblica

$P = C^d \pmod{m}$ (d, m) è la chiave privata

Le 2 chiavi hanno una parte comune **m (Modulo)** che è tipicamente di 1024 o 2048 bit e una parte specifica **e, d (Esponenti)** di circa 20 bit.

Per violare la chiave privata occorre determinare **d** . Questo può essere fatto solo per “forza bruta” fattorizzando **m** che è il prodotto di 2 numeri primi.

L'operazione richiederebbe un tempo estremamente grande anche sul più veloce dei computer.

La cifratura $C = P^e \pmod{m}$ limita la dimensione massima di P (deve essere $P < m$)

Comandi OpenSSL: RSA

OpenSSL 1.x supporta RSA con i seguenti comandi: genrsa, rsa e rsautl.

Nota: In openSSL 3.x i comandi sono genpkey, pkey e pkeyutl

genrsa (man genrsa) consente di creare una coppia di chiavi RSA:

```
openssl genrsa -out rsa_key.pem 2048  
# aggiungere -des3 per cifrare la chiave privata con un pass-phrase
```

rsa (man rsa) consente di processare le chiavi rsa. Esempi:

```
openssl rsa -in rsa_key.pem -text      (mostra il contenuto)  
openssl rsa -in rsa_key.pem -pubout -out rsa_pub.pem (estrae la chiave  
pubblica)
```

rsautl viene utilizzato per cifrare/decifrare firmare/verificare con chiavi RSA.

Esempio encrypt/decrypt:

```
openssl rsautl -encrypt -pubin -inkey rsa_pub.pem -in text.txt      -out  
    encrypted.txt  
openssl rsautl -decrypt           -inkey rsa_key.pem -in encrypted.txt -out text.txt
```

Esempio sign/verify:

```
openssl rsautl -sign            -inkey rsa_key.pem -in text.txt      -out signed.txt  
openssl rsautl -verify -pubin   -inkey rsa_pub.pem -in signed.txt -out text.txt
```

Comandi OpenSSL: DIGEST firmato con RSA

Le chiavi RSA possono essere utilizzate per firmare il Digest di un messaggio:

Il seguente comando crea il digest di file.txt utilizzando SHA1, quindi lo firma con la chiave privata:

```
openssl dgst -sha1 -sign rsa_key.pem -out file.sha1_sign file.txt
```

Per la verifica occorre il messaggio originale, il digest firmato e la chiave pubblica di chi ha firmato:

```
openssl dgst -sha1 -verify rsa_pub.pem -signature file.sha1_sign file.txt  
> Verified OK
```

Algoritmi a chiave pubblica: Certificazione dell'identità

Se io genero una coppia di chiavi, uso la chiave privata per cifrare un messaggio e pubblico il messaggio cifrato, chiunque può verificare con la mia chiave pubblica che il messaggio l'ho cifrato io, ma questo non garantisce nulla riguardo la mia identità.

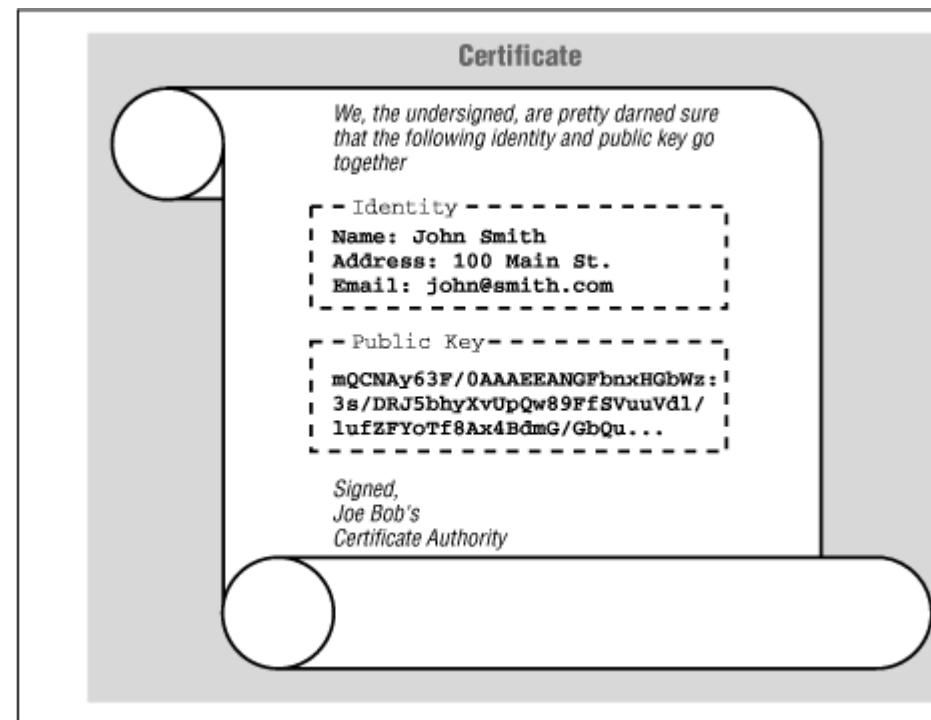
Per associare in modo certo una chiave pubblica a una persona (o host o software) si utilizza il Certificato, ovvero l'insieme della chiave pubblica e dei dati del proprietario firmati (cifrati con la chiave privata) da una **Autorità di Certificazione** che garantisce l'autenticità dei dati contenuti nel **Certificato**.

Applicazioni principali:

Posta elettronica (identità del mittente)

Connessioni Web (identità del server e del client)

Software (identità dello sviluppatore)



Certificati X.509

X.509 è lo standard Internazionale emanato da ITU per il formato dei Certificati.

La versione v.3 di X.509 è utilizzata da SSL/TLS.

Questo standard stabilisce quali informazioni possono comporre un certificato; i principali campi sono:

Version: numero della versione di X.509 (v1, v2 o v3)

Serial Number: numero univoco di emissione da parte della CA

Signature Algorithm: Algoritmo usato per firmare il certificato

Issuer: Distinguished Name DN della CA che ha emesso il certificato

Validity: Inizio e Fine del periodo di Validità.

Subject: Distinguished Name DN del proprietario del Certificato.

Subject Public Key Info : Chiave pubblica (Modulo + Esponente) e algoritmo utilizzato

X509v3 Extensions: Estensioni opzionali (solo v3).

Signature: Firma da parte della CA (MD del Certificato, cifrato con la chiave privata della CA).

Firmata dalla CA (Issuer)							
Version	Serial Number	Signature	Issuer	Validity	Subject	Subject Public Key Info	Extensions

La Certification Authority

La CA è un ente che firma le richieste di certificato da parte di una comunità di utenti/host/software garantendone l'identità.

La CA possiede una propria coppia di chiavi e autofirma la propria richiesta (self-signed).

Per identificare univocamente i certificati esiste un name-space gerarchico di certificati, in cui ogni nodo ha un attributo e un valore.

I principali attributi sono: O (Organization), OU (Org. Unit), C (Country), CN (CommonName) , ecc

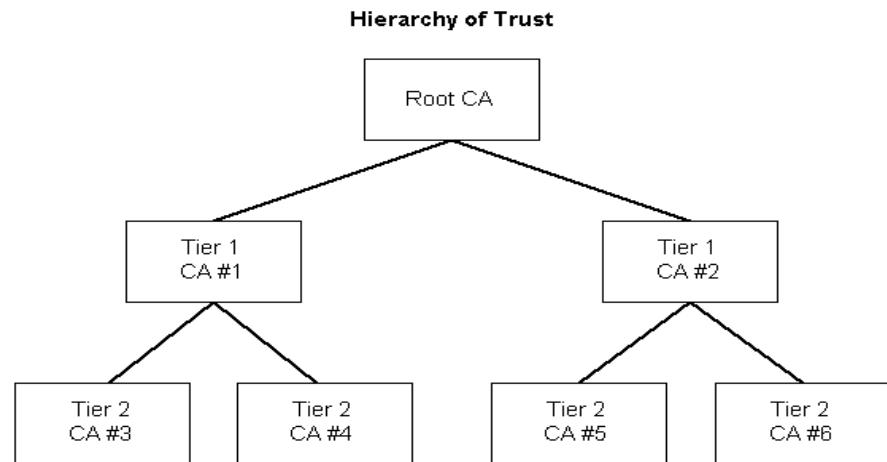
Ogni certificato possiede quindi un FQDN (Subject). Esempio:

C=IT, O=UniprScienze, OU=Staff, CN=roberto alfieri/emailAddress=roberto.alfieri@unipr.it

La CA possiede un BaseDN ed è vincolata ad emettere Certificati all'interno del suo BaseDN.

Una CA può firmare certificati anche per altre CA, poiché il namespace gerarchico consente di organizzare diverse CA all'interno dello stesso albero.

Ogni CA deve gestire la lista dei Certificati Revocati (CRL) che va aggiornata regolarmente.



Istanze di Certification Authority

Alcune CA rilasciano certificati a pagamento e sono già inserite e riconosciute dai più diffusi client Web e SMTP, come ad esempio Comodo, Entrust, GeoTrust, GlobalSign, Cybertrust, Verisign e DigiCert (vedi le impostazioni del Web Browser).

Generalmente una Organizzazione/Ente/Impresa può creare una CA per le certificazioni riconosciute al suo interno

Il [GARR](#) fornisce un Certification Service per tutti gli enti afferenti:
<https://www.servizi.garr.it/cs>

GARR partecipa al [Trusted Certificate Service](#) (TCS) promosso da Géant a favore delle reti della ricerca europee.

Tramite questo servizio offre gratuitamente alla propria comunità certificati digitali x.509 emessi da una delle maggiori Certification Authority commerciali: Sectigo Limited, riconosciuta automaticamente dalla quasi totalità dei browser web esistenti.

Comandi OpenSSL: ca

OpenSSL consente di creare una propria Certification Authority:

Questo comando genera chiave e certificato autofirmato:

```
openssl req -config openssl.cnf -newkey rsa:512 -days 1000 -nodes \
             -keyout cakey.pem -out cacert.pem -x509 -new
```

Con il comando **ca** vengono gestite le operazioni della Certification Authority.
Esempi:

- Firma di una richiesta di Certificato:

```
openssl ca -in req.pem -out newcert.pem
            -config /var/www/html/myCA/openssl.cnf
```

- Generazione della Certificate Revocation List:

```
openssl ca -gencrl -out crl.pem
```

Esempio di Certificato della CA (MyCA)

Certificate:

Data:

Version: 3 (0x2)

Serial Number: 0 (0x0)

Signature Algorithm: sha1WithRSAEncryption

Issuer: C=IT, O=UniprScienze, OU=UniprScienzeCA, CN=UniprScienze/emailAddress=roberto.alfieri@fis.unipr.it

Validity

Not Before: Jun 4 20:21:12 2010 GMT

Not After : Jun 3 20:21:12 2015 GMT

Subject: C=IT, O=UniprScienze, OU=UniprScienzeCA, CN=UniprScienze/emailAddress=roberto.alfieri@fis.unipr.it

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public Key: (512 bit)

Modulus (512 bit):

00:cc:89:ba:73:31:b2:b3:e8:74:d9:30:b8:93:02:

51:b1:12:8f:f5:e7:f9:2f:96:68:15:e6:4a:19:d8:

66:6a:e9:74:66:3e:9f:6f:02:25:ef:3e:5f:09:c3:

63:70:e7:40:63:53:a8:75:3c:b7:a8:cb:68:de:4e:

dd:c2:89:c9:6d

Exponent: 65537 (0x10001)

X509v3 extensions:

X509v3 Subject Key Identifier:

7A:4A:AE:FA:83:9D:C8:DC:E7:74:0A:B9:17:3A:CB:38:3A:31:CA:A7

X509v3 Authority Key Identifier:

keyid:7A:4A:AE:FA:83:9D:C8:DC:E7:74:0A:B9:17:3A:CB:38:3A:31:CA:A7

Comandi OpenSSL: req

Questo comando serve per generare una richiesta di certificato.
La chiave privata può essere fornita (-key) o può essere generata dal comando.
Con l'opzione -nodes non viene cifrata la chiave privata. Questo serve per i certificati host per i quali non è possibile digitare una Passphrase.
Il file di configurazione (openssl.cnf) contiene le informazioni relative al proprietario del certificato, alcune delle quali vengono richieste interattivamente.

Esempi:

Generazione della coppia di chiavi (chiave privata e richiesta di certificato):

```
openssl req -new -nodes -out hostreq.pem -keyout hostkey.pem -config  
openssl.cnf
```

Verifica il contenuto della richiesta:

```
openssl req -in req.pem -text
```

Certificati self-signed

Con l'opzione `-x509` non viene generata una richiesta, ma un certificato self-signed
Se il file di configurazione (`-config`) non viene fornito i dati del certificato vengono
richiesti interattivamente.

Generazione del certificato self-signed per un server:

```
openssl req -new -nodes -out hostcert.pem -keyout hostkey.pem -x509
```

Generating a 1024 bit RSA private key

writing new private key to 'keyfile.pem'

You are about to be asked to enter information that will be incorporated
into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

Country Name (2 letter code) [GB]: IT

State or Province Name (full name) [Berkshire]: Parma

Locality Name (eg, city) [Newbury]: Parma

Organization Name (eg, company) [My Company Ltd]: UNIPR

Organizational Unit Name (eg, section) []: LPR

Common Name (eg, your name or your server's hostname) []: lpr1.fis.unipr.it

Email Address []: roberto.alfieri@unipr.it

Formati dei certificati X.509 : DER e PEM

I certificati X.509 possono essere rappresentati in diversi possibili formati.

I principali sono: DER, PEM e PKCS12.

DER è un formato binario utilizzato in ambiente Windows e Java con estensioni .DER o .CER.

PEM è testuale (base64) ed è utilizzato prevalentemente in ambiente Unix.

Può contenere Certificati, richieste di Certificati, chiavi private.

Gli oggetti contenuti sono delimitati da stringhe del tipo:

```
-----BEGIN CERTIFICATE-----  
-----END CERTIFICATE-----  
-----BEGIN RSA PRIVATE KEY-----  
-----END RSA PRIVATE KEY-----  
-----BEGIN CERTIFICATE REQUEST-----  
-----END CERTIFICATE REQUEST-----
```

Comandi OpenSSL: x509

Visualizzare i campi e cambiare formato (tra PEM e DER) di un certificato:

Esempi:

```
openssl x509 -in cert.pem -noout -text  
openssl x509 -in cert.pem -noout -serial  
openssl x509 -in cert.pem -noout -subject  
openssl x509 -in cert.pem -inform PEM -out cert.der -outform DER
```

<https://www.openssl.org/docs/man1.1.1/man1/x509.html>

Formati dei certificati X.509 : PKCS

PKCS (Public Key Cryptography Standards) è un gruppo di standard creati da "RSA Data Security" con lo scopo di creare formati di interoperabilità.

Sono stati creati 15 standard (da PKCS#1 a PKCS#15) ma i formati prevalentemente utilizzati sono:

PKCS#12: Personal Information Exchange Syntax Standard.

Nato come evoluzione di PFX, definisce un formato per immagazzinare chiave privata e certificato in un file protetto con password (cifratura con chiave simmetrica). L'estensione è p12.

PKCS#7: Cryptographic Message Syntax Information.

E' un formato per rappresentare messaggi cifrati o firmati ed è utilizzato da S/MIME per l'invio di e-mail cifrate e/o firmate. Recepito da IETF (RFC 2986).

PKCS#11: Cryptographic Token Interface

Definisce le API per utilizzare i Token Crittografici, come le SmartCard e le chiavi USB

Comandi OpenSSL: pkcs12

Questo comando serve per gestire i file in formato PKCS12.

<https://www.openssl.org/docs/manmaster/man1/openssl-pkcs12.html>

Esempi:

Genera il file PKCS12 dai file PEM:

```
openssl pkcs12 -export -out cert.p12 -inkey userkey.pem -in usercert.pem
```

Dal formato PKCS12 a PEM (separatamente chiave e certificato):

```
#estrai la chiave da cert.p12 (aggiungere -nodes per i server)
```

```
openssl pkcs12 -nocerts -in cert.p12 -out userkey.pem
```

```
#estrai il certificato da cert.p12
```

```
openssl pkcs12 -clcerts -nokeys -in cert.p12 -out usercert.pem
```

```
#visualizza il certificato in chiaro
```

```
openssl x509 -in usercert.pem -text
```

Dal formato PEM a PKCS12 (con cypher AES256-CBC):

```
openssl pkcs12 -export -inkey userkey.pem -in usercert.pem -out test.p12  
-certpbe AES-256-CBC -keypbe AES-256-CBC
```

Comandi OpenSSL: s/mime

MIME (Multipurpose Internet Mail Extensions)

è una estensione del protocollo di posta Elettronica per poter includere in un unico messaggio più documenti (Allegati), codificati in ASCII Standard (vedi la Posta Elettronica).

MIME introduce un header in cui è possibile inserire campi che descrivono il contenuto, tra cui:

Content-Type : Tipo di dato contenuto (esempio image/JPEG)

Content-transfer-encoding: Codifica del contenuto (esempio BASE64)

S/MIME (Secure/MIME)

include in questo schema la possibilità di

cifrare/decifrare il contenuto, utilizzando un algoritmo a chiave simmetrica.

La chiave simmetrica viene cifrata con la chiave pubblica del destinatario e inviata insieme al messaggio stesso.

(Content-Type del messaggio cifrato: application/x-pkcs7-mime)

firmare/verificare un messaggio allegando la Firma

(Content-Type della firma: application/x-pkcs7-signature)

Consultare le opzioni: man smime

<https://www.openssl.org/docs/manmaster/man1/openssl-smime.html>

Cifrare/Decifrare con s/mime

Per inviare un messaggio cifrato al destinatario la chiave simmetrica viene cifrata con la chiave pubblica del destinatario e inviata assieme al messaggio stesso. Il destinatario legge la chiave simmetrica utilizzando la propria chiave privata, quindi decripta il messaggio con la chiave simmetrica.

Il Mittente codifica il messaggio con seguente comando

```
openssl smime -encrypt -text -in message.txt \
               -out encrypted-message.txt destination-user-certificate.pem
```

viene generato un file Mime con una intestazione del tipo:

MIME-Version: 1.0

Content-Type: application/x-pkcs7-mime; name="smime.p7m"

Content-Transfer-Encoding: base64

Il Destinatario lo decodifica con il seguente comando:

```
openssl smime -decrypt -text -in encrypted-message.txt \
               -out decrypted-message.txt -inkey userkey.pem
```

Firmare/Verificare con s/mime

La firma consiste nel cifrare con la chiave privata del mittente il Digest del Messaggio, che verrà inviato insieme al messaggio stesso.

- **Garanzia dell'identità del Signer:** Consiste nella decifratura della Firma con la chiave pubblica del Signer.
- **Garanzia di Integrità del Messaggio:** Il messaggio è integro se il Digest decifrato nella firma e il Digest ricalcolato sono uguali.

Per firmare un messaggio:

```
openssl smime -sign -text -in message.txt -out signed-message.txt \
               -signer usercert.pem -inkey userkey.pem
```

Il certificato del Signer è incluso nel messaggio.

Con questo comando il Destinatario estrae il Certificato del Signer:

```
openssl smime -pk7out -in signed-message.txt | openssl pkcs7 -print_certs
```

Con questo comando il destinatario verifica il certificato del Signer tra le CA di cui si fida (-CAfile o -CPath) , quindi usa il Certificato per decifrare la firma ed estrarre il MD, infine confronta il MD ricevuto con quello calcolato.

```
openssl smime -verify -text -in signed-message.txt -CAfile CAcert.pem
```

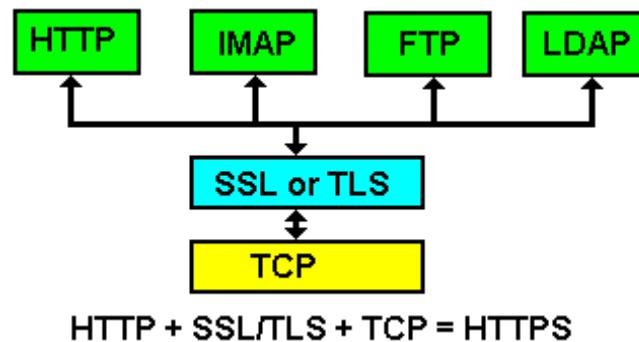
Crittografia nella comunicazione

Gli algoritmi crittografici simmetrici e a chiave pubblica vengono utilizzati per costruire protocolli di rete con l'obiettivo di cifrare la comunicazione, fornendo diversi servizi di sicurezza quali la **Confidenzialità, l'Autenticazione e il Non Ripudio**.

Il principale protocollo di questo tipo è **SSL** (Secure Socket Layer), poi diventato **TLS** (Transport Security Layer).

SSL /TLS è strutturato come un layer che si pone tra il trasporto (TCP) e l'applicazione.

Molte applicazioni di rete (ad esempio HTTP, IMAP, LDAP) sono state adattate per appoggiarsi su SSL (anziché TCP) per usufruire dei servizi di sicurezza del protocollo.



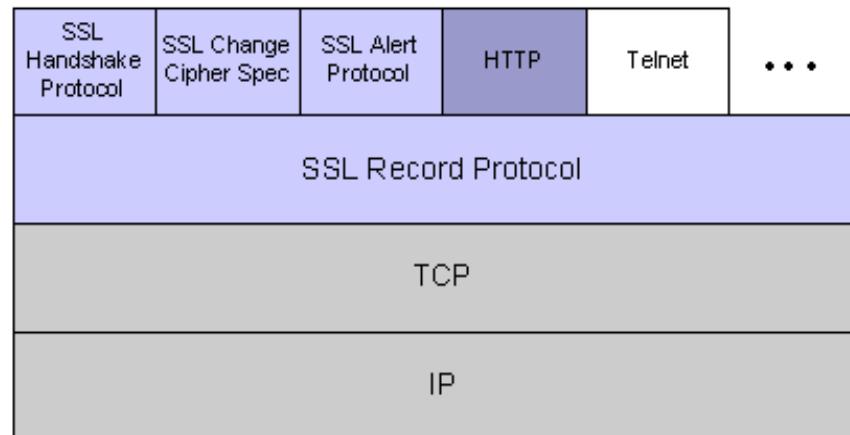
SSL/TLS

Il protocollo SSL mette in sicurezza una connessione Client-Server introducendo un **Layer di Sicurezza al livello di Trasporto ([TLS](#))**.

In questo modo l'applicativo non vede più le API di TCP e deve quindi riscrivere l'interfaccia con il livello di Trasporto utilizzando le API SSL/TLS.

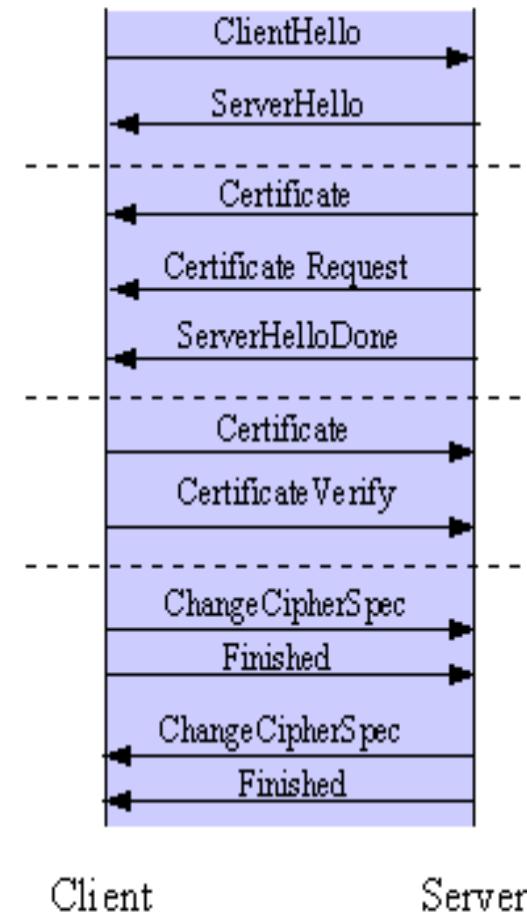
Vengono utilizzati 2 protocolli principali:

- ▶ Il Record Layer Protocol è utilizzato per trasmettere e ricevere dati (cifratura e decifratura) mediante un Cipher Simmetrico.
- ▶ L'Handshake Protocol interviene solo all'inizio (o per ripristinare una sessione interrotta). Negozia i parametri per lo scambio dei dati (scambio dei Certificati e condivisione di una chiave di Sessione per il Cipher Simmetrico).



SSL Handshake Protocol

1. Il client manda SSL version e il Cipher Suite da utilizzare
2. Il server manda il proprio SSL versione e il cipher suite
3. Il server manda il proprio certificato
4. Il server opzionalmente richiede il certificato del client
5. Il client verifica il certificato del server e procede solo se è ok
6. Il client genera il pre-master secret
7. Il client critta la chiave con il cert. del server e la invia
8. Se il sever ha richiesto l'autenticazione del client,
Il client cifra un challenge e lo invia al server
9. Se il server non riesce a decifrare termina la sessione.
10. Client e server usano il pre-master secret per
Generare la chiave di sessione condivisa.
11. Il client manda un messaggio di conclusione dell'handshake
12. Il server manda un messaggio di conclusione dell'handshake



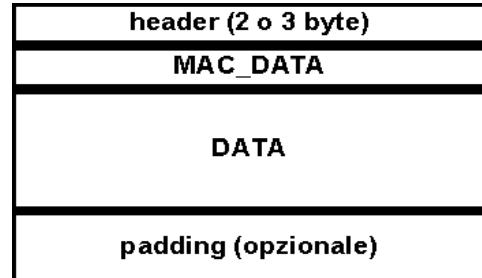
In qualsiasi momento entrambi possono **Renegoziare** la connessione, nel qual caso la procedura è ripetuta.

SSL Record Protocol

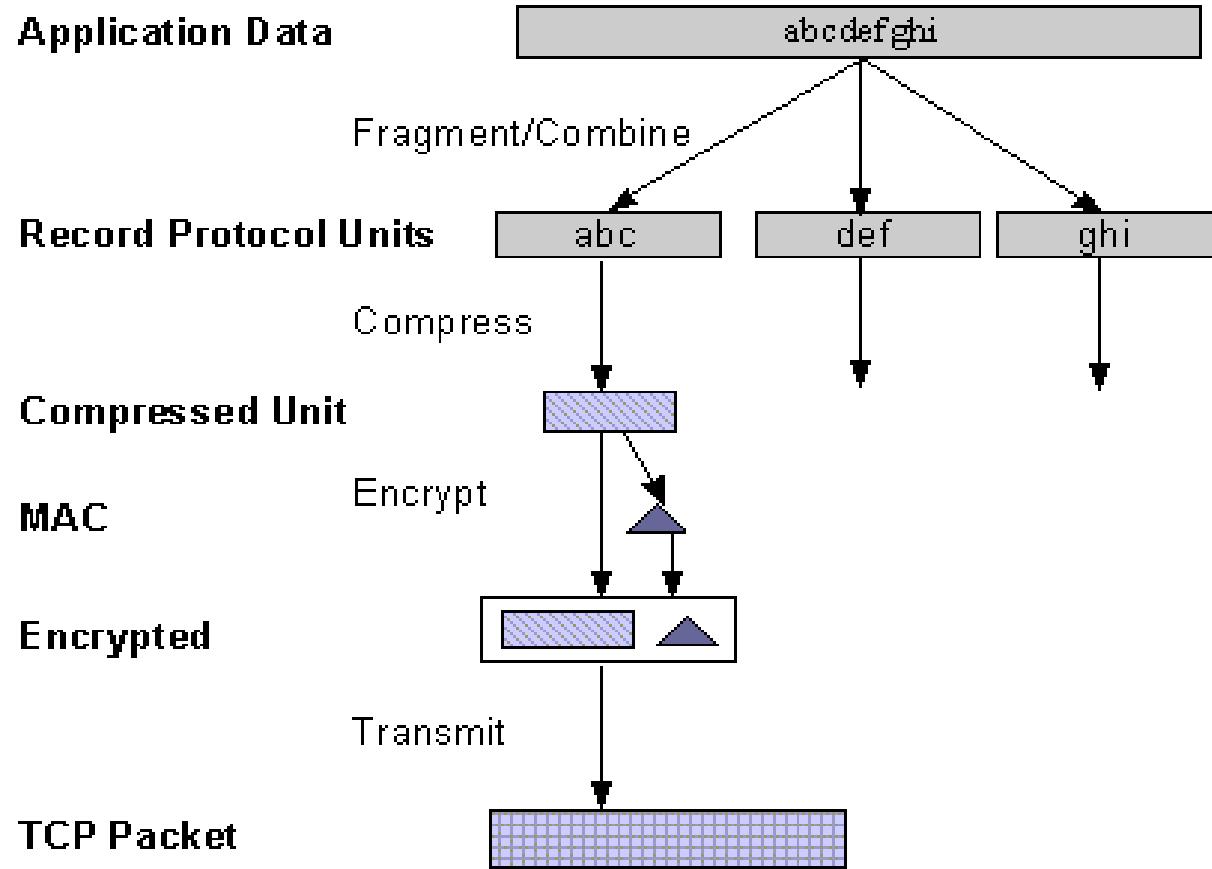
Al termine dell'Handshake Protocol inizia l'SSL Record Protocol:

1. Frammentazione del flusso
2. Compressione dei frammenti
3. Calcolo del Digest
4. Cifratura

Pacchetto SSL



Application Data



SSL Cipher suites

Una sessione SSL/TLS richiede l'utilizzo di un set di strumenti crittografici per

- Scambio chiavi (Kx)
- Autenticazione (Au)
- Cifratura del canale (Enc)
- Digest (Mac)

Un cipher suite è un insieme di strumenti che implementano le funzioni necessarie.

<https://www.ssl.com/it/guida/tls-conformit%C3%A0-agli-standard/>

Il seguente comando elenca le cipher suite supportate in openssl:

```
openssl ciphers -v
```

Applicazioni SSL e StartTLS

Alcune versioni di applicazioni comuni incorporano SSL/TLS attivando connessioni cifrate in ascolto su porte diverse e che possono quindi coesistere con le versioni non cifrate. Le principali applicazioni SSL sono:

- ▶ **https** (HTTP over SSL) 443/tcp
- ▶ **pop3s** (POP3 over SSL) 995/tcp
- ▶ **imaps** (IMAP over SSL) 993/tcp
- ▶ **smt�** (SMTP over SSL) 465/tcp
- ▶ **ldaps** (LDAP over SSL) 636/tcp

STARTTLS è una estensione della comunicazione in chiaro, che offre l'opzione di passare ad una connessione cifrata (SSL o TLS) anziché passare ad una connessione cifrata separata.

IMAP, SMTP , POP e LDAP hanno implementazioni con supporto StartTLS.

Comandi OpenSSL: s_client s_server

s_client (info: `man s_client`) è un client in grado di attivare una connessione con un server ssl/tls. Visualizza i dati del protocollo ssl handshake (protocollo tls utilizzato, cipher, la chiave di sessione, il certificato del server, subject, ecc)

Esempio: `openssl s_client -connect www.unipr.it:443`

Il carattere R forza la rinegoziazione, Q forza la disconnessione

Una volta connessi è possibile digitare manualmente i comandi da inviare al server (ad esempio, se il server è www : "GET / HTTP/1.0")

s_server (info: `man s_server`) svolge le stesse funzioni lato server. Esempio:

```
openssl s_server -accept 443 -www -cert hostcert.pem -key hostkey.pem
```

Con l'opzione www viene emulato un web server che risponde al browser con i dati salienti della connessione SSL.

Programmazione openSSL

La libreria **SSL** fornisce le API necessarie per la programmazione in C di canali cifrati.

<https://developer.ibm.com/tutorials/l-openssl/>

La cifratura avviene grazie all'astrazione di I/O, denominata BIO (Basic I/O abstraction), che consente di creare un canale cifrato o in chiaro

Esempio in C:

TLS_client.c https://wiki.openssl.org/index.php/SSL/TLS_Client

Simple_TLS_Server.c https://wiki.openssl.org/index.php/Simple_TLS_Server

Python consente di creare connessioni SSL/TLS, sia client che server, attraverso diverse librerie che implementano un wrapper di TCP basato su openSSL. Le principali sono:

- **TLS/SSL wrapper for socket objects** <https://docs.python.org/2/library/ssl.html>
- **pyOpenSSL** <https://pyopenssl.org/en/stable/>

Il modulo **httplib** (python 2) supporta sia HTTP, con `httplib.HTTPConnection()`, che HTTPS, grazie al metodo `httplib.HTTPSConnection()`

In Python 3 il modulo `httplib` è stato rinominato [http.client](#)



UNIVERSITÀ
DI PARMA

DIPARTIMENTO DI SCIENZE MATEMATICHE, FISICHE ED INFORMATICHE
Corso di Laurea in Informatica

Sicurezza delle reti – Parte C

Protocolli

RETI DI CALCOLATORI - a.a. 2023/2024

Roberto Alfieri

La sicurezza delle reti: sommario

PARTE A

- ▶ I servizi di sicurezza
- ▶ Metodi e strumenti di attacco
- ▶ Strumenti di Difesa

PARTE B

- ▶ Crittografia applicata e OpenSSL

PARTE C

- ▶ Protocolli di Autenticazione
- ▶ IPsec e VPN

Protocolli di Autenticazione

Riferimenti: http://it.wikiversity.org/wiki/Protocolli_di_autenticazione

Autenticazione (Authentication): un servizio di sicurezza che consente di accettare l'identità dichiarata da una entità mediante la verifica di credenziali.

L'autenticazione può avvenire:

- tra una persona fisica e un host o dispositivo (es. bancomat)
- in una comunicazione di rete (l'origine dei dati o i peer di una comunicazione) mediante un opportuno protocollo

L'autenticazione può essere mutua oppure no, dipende dalle situazioni.

Ad esempio è mutua quando consulto la posta elettronica:

- Il server si deve autenticare con me per dimostrarmi di essere il server che gestisce la mia posta.
- Io devo dimostrare al server che sono il titolare della mailbox.

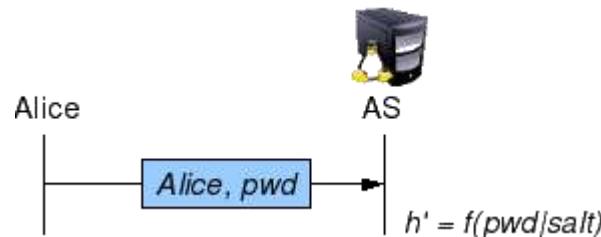
Le tecniche possono basarsi su

- La conoscenza di un segreto (password, PIN, ...)
- tecniche crittografiche
- Caratteristiche biometriche (se l'autenticazione avviene tra una persona ed un host locale): timbro voce, impronta digitale, fondo dell'occhio, ecc

Autenticazione tramite password: PAP

PAP (Password Authentication Protocol)

- presume che il canale sia sicuro (non intercettabile)
- Il client manda al server il proprio Nome e la Password
- Il server cerca in una tabella il nome utente e verifica la correttezza della password applicando una funzione di trasformazione
(che consente di evitare che il server memorizzi la password in chiaro)
- Ancora in uso in PPP anche se deprecato.



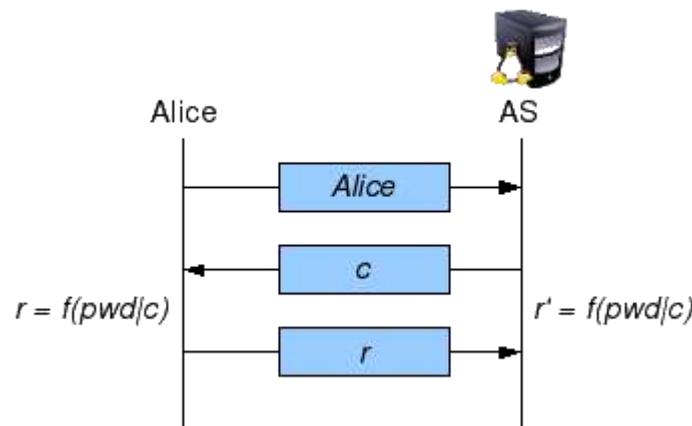
Autenticazione tramite password: CHAP

CHAP (Challenge Handshake Authentication Protocol)

- Usato in PPP
- Windows usa una variante di CHAP detta MS-CHAP
- Il server invia un numero casuale c (challenge) utilizzato dal client come salt
- La funzione di trasformazione $r=f(pwd,c)$ è calcolata sia dal server che dal client
- L'implementazione standard di CHAP usa MD5: $r=MD5(pwd, c)$

Vantaggi: La password non viene scambiata tra client e server

Problemi: Il DB delle password deve essere salvato in chiaro. Attacco al dizionario



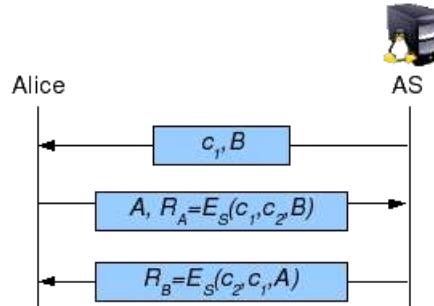
Autenticazione con challenge e chiave simmetrica

Le 2 parti A e B , che condividono una chiave simmetrica S, si inventano ciascuna un numero casuale, detto Challenge (c1 e c2)

Il server invia la propria identità (B) e il proprio Challenge (c1)

Il client risponde inviando la propria identità (A) e la cifratura di c1, c2 e B.

Il server chiude il protocollo inviando la cifratura di c1, c2 e A.



Vantaggi: I messaggi cifrati non sono esposti ad attacco al dizionario.

Problemi:

- ▶ Se abbiamo N nodi ogni nodo deve conoscere N-1 chiavi.
- ▶ La condivisione di una chiave simmetrica si espone ad intercettazione.

Autenticazione con KDC

Il modello del “Centro di Distribuzione delle Chiavi” (KDC, Key Distribution Center) si applica ad una comunità di N entità (persone/host/servizi) che devono autenticarsi reciprocamente.

In questo schema ogni utente ha una singola chiave condivisa con il KDC.

Esempio: A deve comunicare con B

A condivide con KDC la chiave K_a , B condivide con KDC la chiave K_b

A sceglie una chiave di sessione K_s , invia a KDC la chiave e B, in modo cifrato.

KDC decifra K_s e la invia a B

A \rightarrow A, $K_a(B, K_s)$ \rightarrow KDC

KDC \rightarrow $K_b(A, K_s)$ \rightarrow B

Vantaggi: Singola chiave K_a per comunicare con N entità.

Problemi: A deve inserire la chiave K_a per ogni connessione.

Autenticazione Kerberos

Kerberos è un protocollo di Autenticazione (progettato al MIT) che implementa il modello del “Centro di Distribuzione delle Chiavi”.

E' ampiamente diffuso soprattutto negli USA sia su Linux che Windows

Un sistema Kerberos gestisce una comunità di utenti (REALM) in cui ogni utente ha una singola chiave condivisa Ka con il KDC, ma il KDC si compone di 2 server:

AS (Authentication Server) Gestisce il LOGIN

TGS (Ticket Granting Server) Gestisce la sessione

La password di A (Ka) viene usata una sola volta per tutte le autenticazioni della sessione (Single Sign On - SSO) e rimane sul computer del client solo per pochi millisecondi.

La chiave di sessione che A presenta a B serve solo a dimostrare l'identità di A (autenticazione). B deciderà cosa consentire di fare ad A (autorizzazione)

Autenticazione Kerberos

Innanzitutto A chiede all'AS la chiave di sessione Ks (Login sul REALM):

A → A → AS
A ← Ka(Ks), Ktgs(A,Ks) ← AS

Ktgs(A,Ks) contiene A e Ks cifrate con la chiave segreta del TGS

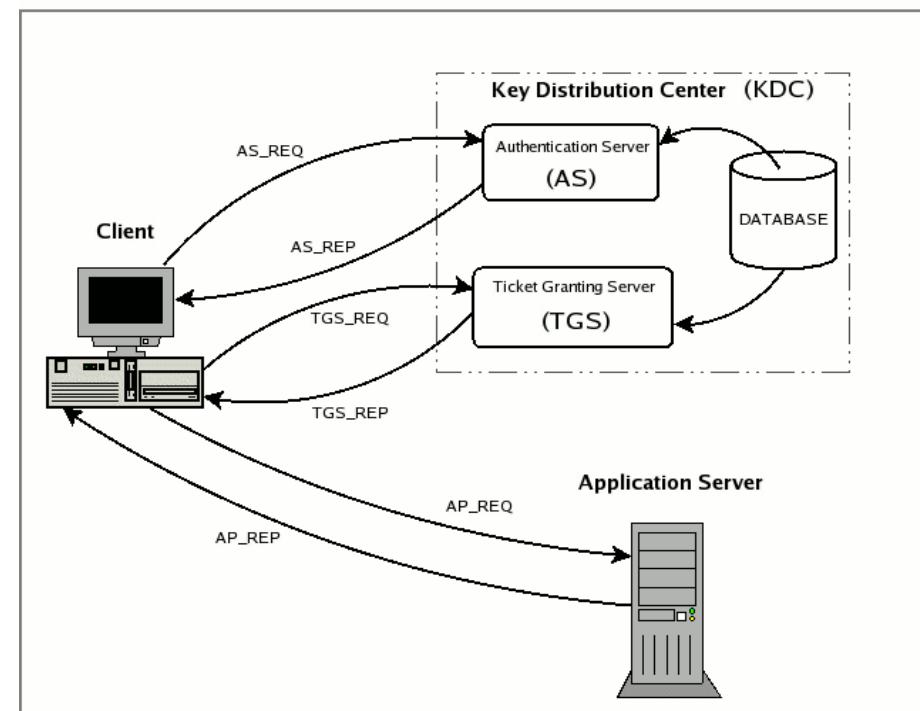
Quando A deve comunicare con B chiede al TGS un Ticket Kab da usare con B:

A → Ktgs(A,Ks), B, Ks(t1) → TGS
A ← Ks(B, Kab), Kb(A,Kab) ← TGS

Quindi A si rivolge a B comunicandogli la chiave di sessione Kab:

A → Kb(A,Kab), Kab(t1) → B
A ← Kab(t2) ← B

I timestamp t1 e t2 impediscono che qualcuno possa intercettare i messaggi e replicarli con un mittente falsificato (spoofed).



Scambio chiavi di Diffie-Hellman

Consente a 2 entità che non hanno avuto contatti in precedenza di stabilire in modo sicuro una chiave simmetrica condivisa.

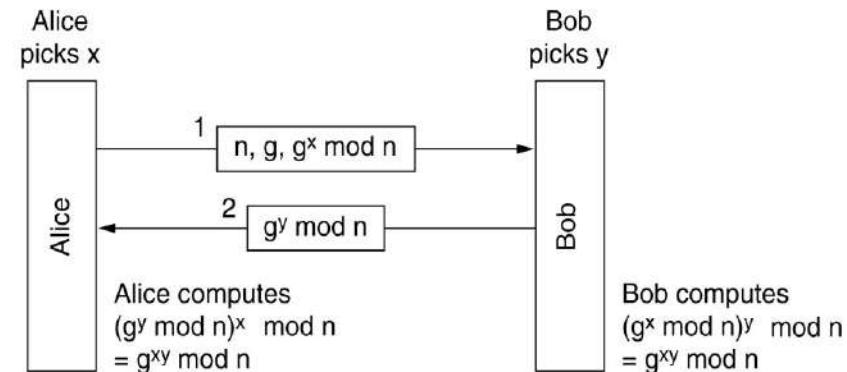
Servizi di sicurezza: **confidenzialità senza autenticazione**.

A e B devono condividere 2 numeri grandi n e g che possono scambiarsi in chiaro.
(n e (n-1)/2 sono primi, $g=f(n)$ opportunamente calcolato)

1) A sceglie X grande che mantiene segreto
quindi invia $A \rightarrow n, g, g^x \text{ mod } n \rightarrow B$

2) B sceglie Y grande che mantiene segreto
quindi invia $A \leftarrow g^y \text{ mod } n \leftarrow B$

3) B calcola $(g^x \text{ mod } n)^y \text{ mod } n = g^{xy} \text{ mod } n$
A calcola $(g^y \text{ mod } n)^x \text{ mod } n = g^{xy} \text{ mod } n$



$g^{xy} \text{ mod } n$

è la chiave condivisa di sessione

Autenticazione con PKI

L'utilizzo di una PKI (Public Key Infrastructure) ha il vantaggio di non richiedere preventivamente chiavi condivise.

I nodi A e B hanno una coppia di chiavi A-> (Ea,Da) B-> (Eb,Db).

Le chiavi Ea ed Eb sono pubbliche.

A invia la propria Identità e un Challenge Ra a B, cifrati con la chiave pubblica di B.

A -> Eb(A,Ra) -> B

B decifra il messaggio, sceglie una chiave di sessione Ks e la invia ad A:

A <- Ea (Ra, Rb, Ks) <- B

A risponde con il Challenge di B cifrato con la chiave di sessione Ks:

A -> Ks(Rb) -> B

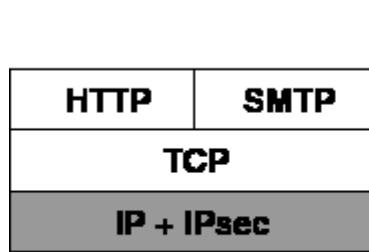
Servizi di sicurezza:

- ▶ **la chiave di sessione Ks è condivisa (confidenzialità)**
- ▶ **gli host hanno verificato l'identità reciproca (autenticazione).**

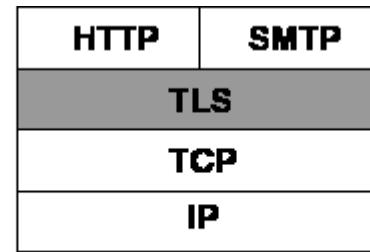
Protocolli per la riservatezza

La cifratura di una comunicazione può avvenire a diversi livelli:

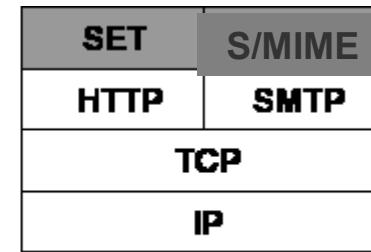
- Alcune applicazioni cifrate si appoggiano sull'applicazione in chiaro. Il payload viene cifrato e quindi veicolato da applicativo non cifrato (vedi ad es. S/MIME su SMTP)
- Il protocollo **SSL/TLS** fornisce un Layer intermedio tra TCP e applicazione che consente di cifrare le applicazioni. Questo richiede la riscrittura delle applicazioni che devono interfacciarsi al layer SSL anziché TCP.



(a) **livello rete**



(b) **livello sessione**



(c) **livello applicazione**

- IPsec è un Layer di cifratura che viene posizionato a livello rete, rendendo la cifratura trasparente al livello delle applicazioni, che non devono essere modificate

IPsec è integrato in IPv6 (Extension Header 50 e 51), mentre è opzionale in IP4.
Attualmente l'uso predominante di IPsec è la creazione di Reti Virtuali Private (VPN)

Protocolli IPsec

Una “connessione” IPsec chiamata SA (Security Association), è una connessione Simplex e ha un Identificatore di sicurezza associato.

Per una connessione Duplex è necessario attivare un SA per ciascuna direzione.

Ogni pacchetto Ipsec include nell'intestazione un indice (Security Parameter Index – SPI) che consente al ricevente individuare la SA e quindi di reperire la chiave di decifratura.

IPsec, analogamente a SSL, è formato da

- ▶ Un protocollo per lo scambio delle chiavi necessarie per la cifratura del canale:
 - **IKE** (Internet Key Exchange)
- ▶ Due protocolli alternativi per la cifratura dei dati sul canale:
 - **AH** (Authentication Header). Gestisce integrità, ma non confidenzialità.
 - **ESP** (Encapsulating Security Payload). Anche confidenzialità (cifratura payload)

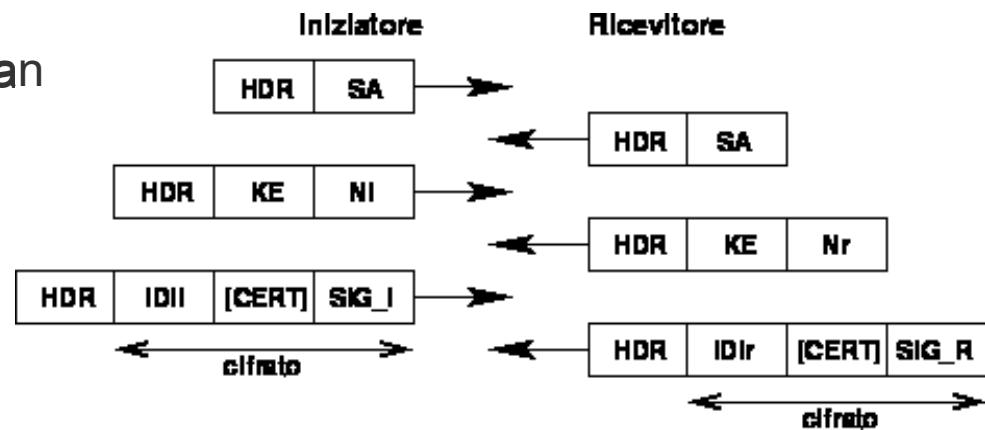
IKE (Internet Key Exchange)

IKE è utilizzato per stabilire una SA.

E' a livello applicazione e usa UDP come trasporto sulla porta 500.

L'obiettivo è stabilire una Shared Session Secret da cui poi derivare la chiave per cifrare la SA.

Viene utilizzato l'algoritmo di Diffie-Hellman



Ogni host IPsec gestisce un Security Association Database che include l'elenco delle SA attive.

Ogni elemento del DB indicizzato dal Security Parameter Index (SPI) include:

- ▶ l'indirizzo di destinazione
- ▶ servizi di sicurezza (AH, ESP)
- ▶ Algoritmi simmetrici usati per cifrare i dati (3DES, AES, ..) e le chiavi associate
- ▶ altri parametri quali l'IPsec lifetime

Transport mode e Tunnel mode

Sia AH che ESP possono funzionare il modalità Transport o Tunnel.

Transport mode

- ▶ connessione host-to-host
- ▶ usato dagli end-point non dai gateway
- ▶ viene cifrato solo il payload dei datagrammi IP, non l'header
- ▶ computazionalmente leggero
- ▶ ogni host deve avere tutto il software necessario ad implementare IPsec
- ▶ si aggiunge solo l'header IPsec; mittente e destinazione si vedono

Tunnel mode

- ▶ connessione Gateway-to-Gateway
- ▶ viene cifrato tutto il pacchetto IP originale
- ▶ utilizzato per realizzare le VPN
- ▶ computazionalmente oneroso
- ▶ solo i Gateway devono avere il software IPsec
- ▶ si hanno punti di centralizzazione quindi single point of failure
- ▶ si ha un doppio incapsulamento, aggiungendo l'header del gateway e l'header IPsec

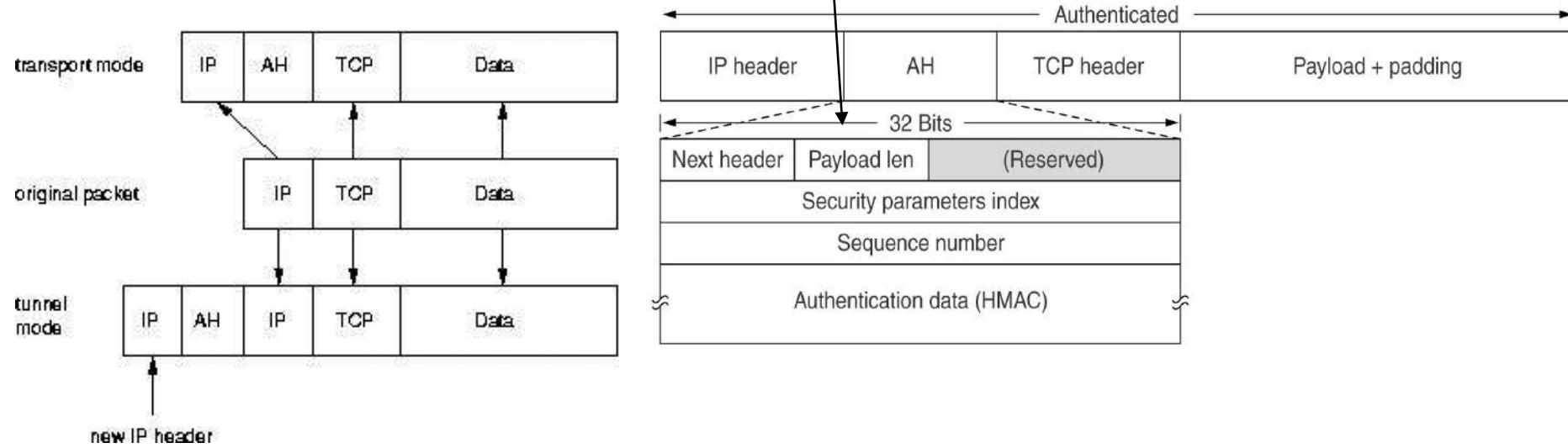
AH (Authentication Header)

AH gestisce integrità del pacchetto, ma non la confidenzialità: non ha la cifratura.

Il protocollo determina una intestazione di 24 Byte che contiene l'HMAC del Datagramma IP (Header+payload)

L'intestazione che può essere inserita

- ▶ nelle estensioni dei protocolli IPv4 e IPv6 (**Transport Mode**)
- ▶ nell'estensione di un nuova intestazione IP che come payload incapsula il pacchetto IP originale (**Tunnel Mode**)



ESP (Encapsulating Security Payload)

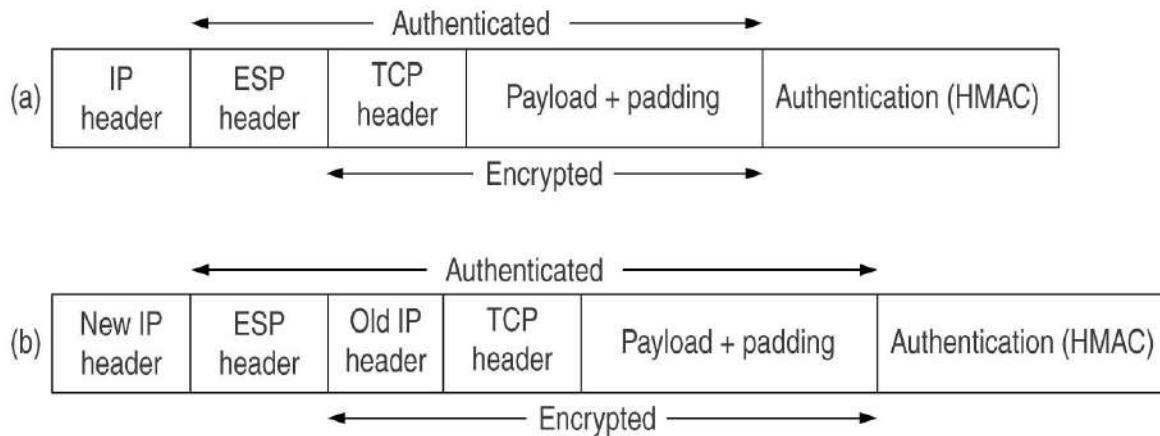
ESP, rispetto a HA, aggiunge la confidenzialità poiché il payload viene cifrato.

Il campo **HMAC** (diversamente da AH)

- ▶ non copre l'Header IP
- ▶ è accodato al payload cifrato. Viene calcolato mentre il pacchetto sta uscendo

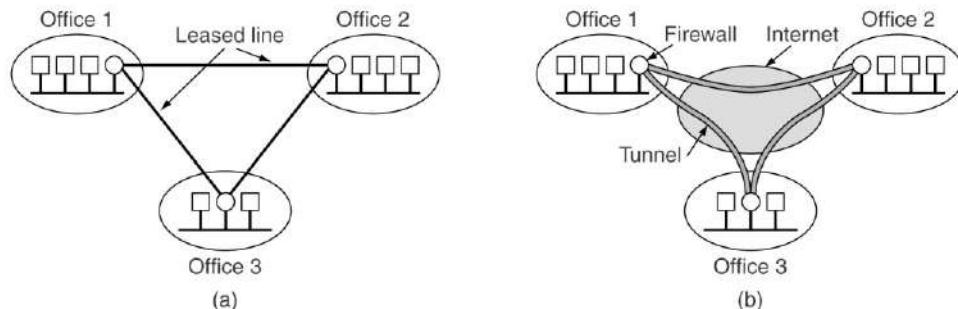
Cifratura:

- Transport:** viene cifrata la trama di trasporto (TCP Header + Payload)
- Tunnel:** viene cifrato il pacchetto IP (old IP header+TCP header+Payload)



VPN (Virtual Private Network)

Una Virtual Private Network o VPN è una rete privata instaurata tra soggetti che utilizzano un mezzo di trasmissione pubblico e condiviso come ATM o, più frequentemente, Internet.



L'utilizzo tipico in Internet è

- ▶ tra 2 o più LAN remote
- ▶ tra una LAN e un singolo host (e.g. Una persona che si trova all'esterno della propria struttura e vuole connettere il proprio portatile come se fosse all'interno.)

In entrambi i casi viene generato un tunnel protetto tra 2 gateway.

I protocolli più utilizzati per realizzare il tunnel cifrati sono:

- ▶ IPsec, SSL/TLS, PPTP (Point-to-Point Tunnelling Protocol di Microsoft)

Use Cases: Forticlient (in uso in UNIPR) <https://forticlient.com/techspec>