



---

# ARCHITETTURA RETI

---

Appunti Lezioni – Anno 2020/2021



GAIA SERAVELLI



# INTERNET (RFC 1122)

È una **rete globale di reti** che permette la comunicazione tra computer in modo diretto e trasparente.

Consente di condividere servizi con il resto del mondo ed è una fonte globale e condivisa di informazioni.

In Internet il ruolo fondamentale ce l'ha l'**Internet Protocol (IP)** che fornisce le funzioni necessarie per l'invio di un pacchetto di bit (**datagram IP**) da una sorgente ad una destinazione, identificabili da un **indirizzo IP**, utilizzando un sistema interconnesso di reti.

**IPv4** fornisce anche i servizi di frammentazione e riassemblaggio di datagrammi, quando le reti hanno una capacità di trasporto inferiore a quella del pacchetto originale. **IPv6**, tuttavia, abolisce questo comportamento.

Scopo dell'IP:

- Dare uno **spazio di indirizzamento**
- **Tecnica** di indirizzamento
- Portare un pacchetto da una sorgente ad una destinazione

## ISP

Gli **Internet Service Provider** sono quelle entità con cui l'utente stabilisce un contratto per ottenere le credenziali che gli consentono un accesso ad Internet.

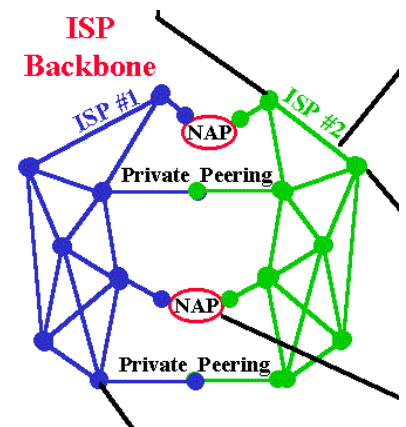
**Erogazione del servizio:** gli ISP hanno un grande Data Center e numerosi Autonomous Systems che raggruppano gli indirizzi IP. All'utenza aziendale verrà assegnata una certa classe di indirizzi IP, all'utenza domestica un'altra.

Se un utente TIM vuole mandare un messaggio ad un utente VODAFONE, il pacchetto viaggerà fino a quando non incontrerà un fornitore comune ai due ISP, che gli permetterà di essere reindirizzata sulla rete VODAFONE. Tutto ciò ovviamente è time consuming, ecco perché gli ISP generalmente hanno un accordo tra loro che gli permette di creare dei **NAP** (Network Access Point) sul territorio nazionale. Tutti gli ISP hanno finanziato una società che ha acquistato un edificio con un Data Center dove tutto il traffico di rete viene accettato.

Oppure è possibile un **private peering**, quando due IPS sanno di avere una fascia di utenza importante, decidono di creare un link con cui si connettono l'uno con l'altro, sgravando il traffico di rete. Questi link si possono usare anche per traffico che non è indirizzato necessariamente a quell'ISP, in modo da creare una seconda strada per il traffico di rete.

**Online Content** → sorgenti di info del mondo reale. Questi server richiedono collegamenti veloci, processori potenti e grandi quantità di memoria. Devono essere **fault tolerant** e **load balanced**.

Le info elettroniche esistenti sono connesse con i **Legacy Systems**, ovvero sistemi basati su tecnologie obsolete ma di cui non si può fare a meno per la loro affidabilità e sicurezza.



# INTERNET GOVERNANCE

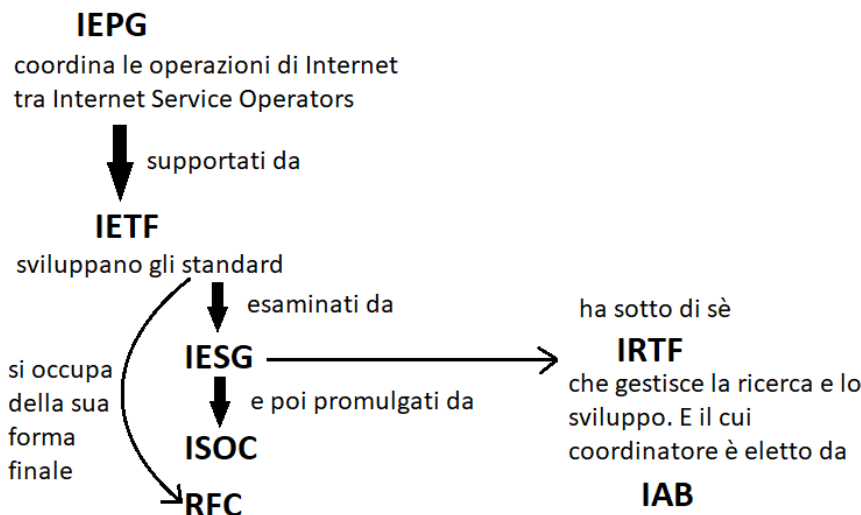
Chi governa Internet?

- **Leader Storici:** coordinano i vari organismi di standardizzazione
- **Rappresentanti dei Governi** più forti
- **Aziende** che hanno un core business in quel segmento
- **Tecnici e Sviluppatori** che mantengono e sviluppano i principali software
- **Utenti** che con i canoni che pagano fanno andare avanti le aziende. Inoltre eleggono i rappresentanti

**IANA** gestisce gli indirizzi IP e gli AS, ha però delegato ad alcune entità regionali la gestione locale.

Attualmente questa gestione è sotto la responsabilità dell'**ICANN** che ha una struttura più democratica rispetto a IANA.

## OPERATIVITA' di INTERNET



Le operazioni di Internet sono coordinate a livello mondiale dall'Internet Engineering Planning Group (**IEPG**), che coordina le attività e l'interoperabilità tra i vasi Internet Service Operators mondiali.

Missione IEPG (RFC 1690):

- Facilitare le **operazioni di gestione** dei servizi globali di Internet
- Promuovere l'introduzione di **nuovi servizi**
- Collegamenti dei **gruppi operativi** di Internet con quelli di **servizio tecnico**.

A livello di implementazione vengono supportati dall'Engineering Task Force (**IETF**) che sviluppano gli standard, i quali verranno poi esaminati dall'Internet Engineering Steering Group (**IESG**) e poi promulgati dall'Internet society (**ISOC**) come standard Internazionali.

L'**RFC** Editor è poi responsabile della preparazione e organizzazione dello standard nella sua forma finale.

**IRTF** (Internet Research Task Force) lavora sotto il controllo dell'IESG e ha lo scopo di promuovere la ricerca e lo sviluppo di Internet. Il suo coordinatore è nominato dall'Internet Activity Board (**IAB**).

# INIZIATIVE DI INTERNET

**GIPI** (Global Internet Policy Initiative) ha siglato con protocollo d'intesa con l'**UNDP** (United Nations Development Program) al fine di promuovere Internet dei paesi in via di sviluppo.

## INTRANET ED EXTRANET

**INTRANET**→prevede l'uso della tecnologia Internet a scopo aziendale interno. Tipicamente una Intranet aziendale si caratterizza per l'uso della tecnologia Internet ma in maniera blindata, in modo tale che la configurazione e l'accesso a questa rete sia tale da non pregiudicare la fuoriuscita di segreti aziendali.

Intranet collega tutte le postazioni, i computer e i server in maniera sicura, consente l'attivazione di una politica di controllo degli accessi alla rete locale e consente l'accesso controllato e selettivo a computer e servizi.

**EXTRANET**→è una porzione della rete aziendale dove viene abbassato il livello di guardia. I servizi vengono quindi posti in un'area speciale chiamata **DMZ** in cui il controllo del Firewall è più debole, In questo modo si consente un accesso più generalizzato.

In questa zona, più vulnerabile ad attacchi, vengono messi i server con i quali l'azienda può comunicare i propri servizi e i propri dati al pubblico, oppure dei database replica di database interni.

## ISO/OSI

**Standard**→ definisce le caratteristiche fisiche ed operative degli apparati di rete. L'adozione di standard è vitale sia per lo sviluppo tecnologico e sia per permettere ai produttori di HW e SW di avere delle basi oggettive su cui basarsi nella creazione dei loro prodotti.

Esistono:     - **Standard de Jure**: emessi da un'organizzazione riconosciuta a livello internazionale - che emana standard

                  -**Standard de Facto**: standard affermatasi grazie alla scelta degli utenti.

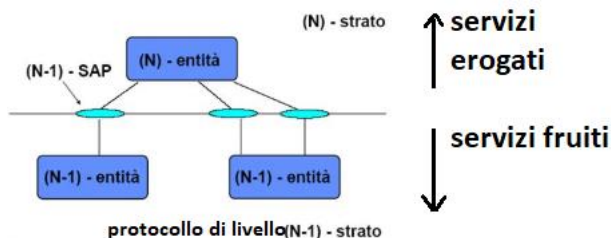
Tra le Organizzazioni di Standardizzazione troviamo:

- l'**IEEE** che nel sottocomitato .802 ha deliberato una serie di standard molto importanti per le reti locali.
- **CCITT** che ora fa parte del **ITU** (International Telecommunication Union) norma gran parte degli standard di Telefonia.
- **ISO** (International Standard Organization) tra i cui prodotti principali c'è l'**OSI**

Il Modello ISO/OSI è un modello suddiviso su 7 livelli tra i quali viene divisa la comunicazione. L'OSI introduce il concetto di **Sistema** che contiene risorse HW e SW, periferiche e programmi; e il concetto di **Applicazione**, ovvero un programma che elabora dati in un certo modo per ottenere un certo scopo.

OSI si occupa dello **scambio di info tra Sistemi**.

Avendo questa suddivisione, si possono dividere logicamente le problematiche di ciascun livello. I livelli, inoltre, ci consentono di capire come avviene la comunicazione in Internet.



Ogni livello aggiunge un suo **Header** al pacchetto.

Ogni livello è costituito da una o più **Peer Entities** che usano i servizi del livello inferiore e forniscono servizi al livello superiore mediante il proprio Service Access Point (**SAP**).

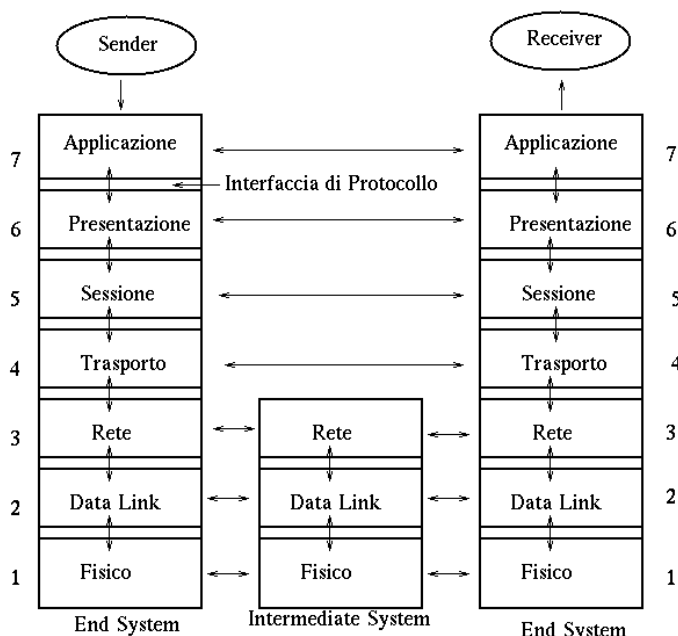
Le operazioni specifiche di ogni livello si chiamano **Protocolli**.

Il ruolo degli **Intermediate System** non è solo quello di instradare i pacchetti nella rete, ma anche quello di poter offrire una connessione tra protocolli differenti.

## 1. Physical Layer

Regole che specificano le connessioni elettriche e fisiche tra i dispositivi fisici. (connessioni dei cavi, tipo di segnale elettrico associati ai pin delle interfacce, ecc...)

Corrisponde agli standard di interfaccia dei vari dispositivi e le sue regole definiscono la trasmissione dati per i terminali, i modem e schede di rete.



Comunicazione tra Livelli

## 2. Data Link

È diviso a sua volta in:

-**LLC** (Logical Link Control) che trasferisce i dati logici al livello MAC, facendo il Checksum.

-**MAC** che controlla se il mezzo trasmissivo del livello fisico è disponibile per ricevere i dati che ha ricevuto dal LLC.

Descrive come un dispositivo realizza la comunicazione con un nodo adiacente.

Definisce il formato dei dati, la frammentazione dei dati, e le procedure di controllo dell'errore.

A questo livello appartengono: le schede di rete, HUB, Switch, Bridge, che suddividono il traffico in base al MAC address.

## 3. Network

È responsabile della realizzazione di una connessione tra due nodi della rete: il nodo sorgente e quello destinatario, inclusa la scelta e la gestione del routing, e lo scambio di info tra due nodi.

Fanno parte di questo livello: i Router, gli Switch, e il Protocollo IP.

## 4. Transport

Trasforma l'info in pacchetti (se l'info è pesante), la estrapola e controlla che arrivi a destinazione correttamente. Si occupa anche del controllo dell'errore, di verificare la sequenza delle info, e di analizzare i fattori di affidabilità dello scambio tra i due nodi.

Fanno parte di questo livello: i protocolli TCP e UDP. E' quindi il primo livello end-to-end, ovvero può connettersi direttamente al nodo di destinazione.

## 5. Session

Fornisce regole per attivare e terminare flussi di dati tra nodi della rete.

È responsabile dell'organizzazione del dialogo tra programmi applicativi e del relativo scambio di dati.

Consente inoltre di servire a sessioni end-to-end servizi più avanzati come:

- Attivazione e terminazione della connessione tra nodi
- Controllo del flusso di messaggi tra nodi
- Controllo del dialogo
- Controllo dei dati da ambo i nodi

## 6. Presentation

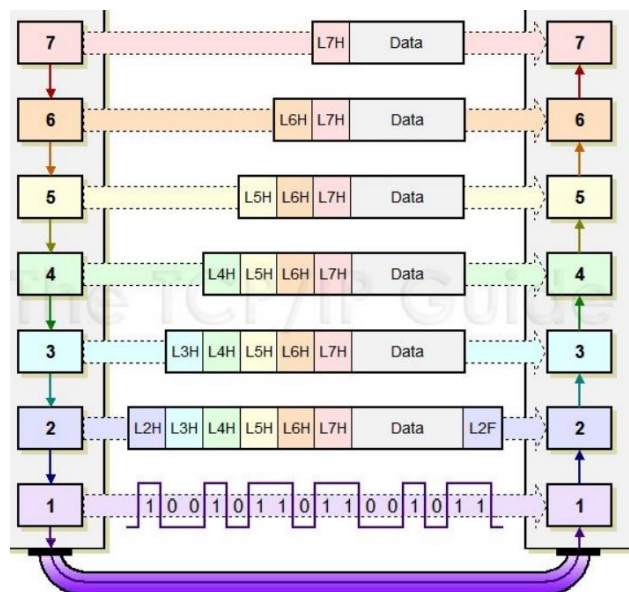
Converte i dati ricevuti in modo da poterli rappresentare opportunamente nel dispositivo di ricezione. Trasforma/Formatta quindi i dati e si occupa di una rappresentazione astratta, di una locale e di una per il trasferimento.

Inoltre si occupa della **crittografia** e **de-crittografia** dei dati e della loro **compressione** e **decompressione**.

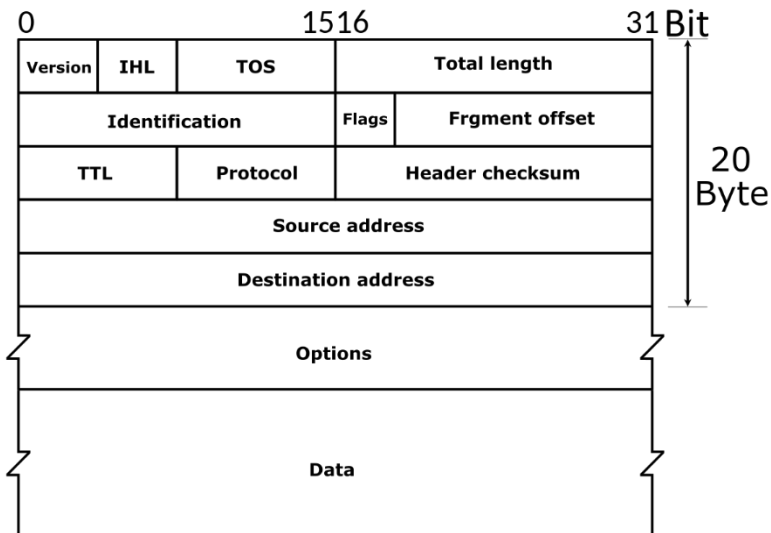
## 7. Application

Comprende tutti i programmi applicativi che consentono l'uso della rete.

Si comporta come una finestra attraverso la quale l'app accede a tutti i servizi messi a disposizione dal modello.



# IP PROTOCOL



E' un protocollo per l'interconnessione di reti, i cui pacchetti sono:

- Consegna senza connessione:** ogni pacchetto è consegnato indipendentemente dagli altri.
- Servizio Inaffidabile:** i pacchetti possono essere persi o consegnati fuori sequenza.
- Consegna Best-Effort:** si fa di tutto per consegnare i pacchetti, l'inaffidabilità si verifica solo per malfunzionamenti HW.

L'IP svolge le funzioni di **routing** scegliendo il percorso che dovranno seguire i dati, decide come gli host e i gateway debbano **elaborare** i pacchetti, come e quando generare **messaggi di errore** e le condizioni in cui **scartare i pacchetti**.

L'unità fondamentale di trasferimento è il **Datagram IP**, il quale è suddiviso in area di intestazione e area dati.

Correntemente sono usate due versioni del protocollo IP: **IPv4** e **IPv6**.

L'**IPv4** prevede indirizzi IP di 32 bit che permettono di identificare univocamente una rete ed uno specifico host appartenente a quella rete.

L'indirizzo si divide in due parti: **rete** e **host**.

Esistono 4 tipi di classi di Indirizzi IP→

Network	Host	Host	Host
255	0	0	0

**Class A**  
Subnet Mask rete < 128

Network	Network	Host	Host
255	255	0	0

**Class B**  
Subnet Mask 128 <= rete <= 191

Network	Network	Network	Host
255	255	255	0

**Class C**  
Subnet Mask 192 <= rete <= 223

Network	Network	Network	Network
---------	---------	---------	---------

**Class D**  
Subnet Mask rete > 223 (multicast)

Un IP Address può essere localmente modificato usando parzialmente i bit relativi agli host come bit di indirizzo per sottoreti.

Una **sottorete** è infatti definita applicando all'indirizzo IP una maschera di bit chiamata **Subnet Mask**: se un bit è 1 = rete, se un bit è 0=host.

Una Subnet Mask quindi definisce la dimensione della sottorete IP a cui appartiene un host, al fine di ridurre il traffico di rete e facilitare la ricerca e il raggiungimento di un host. Questa operazione si chiama **Subnetting** (aggiungere 1 per aumentare la sottorete) ed è usata per dividere una



singola rete in più aree, è possibile fare anche il **Supernetting** (togliere 1 e rimpiazzarli con 0 così da ridurre la sottorete) così da trattare la rete come unica.

**Indirizzi Privati** → Identificano reti private, non instradabili dai router. Sono gestiti ed amministrati dai server **NAT** che convertono gli indirizzi privati in pubblici.

- 10.0.0.0 /8
- 172.16.0.0 /12
- 192.168.0.0 /16

**Indirizzi Link Local** → sono validi in una rete locale e non vengono instradati dai router. Vengono assegnati ad un'interfaccia dal sistema operativo quando ci sono problemi con l'assegnazione di indirizzi da parte di un server **DHCP**. IPv4 riserva gli indirizzi:

- 169.254.0.0 /16

**Indirizzo di Rete** → indirizzo IP dove i bit che indicano la parte host sono pari a 0. Denota la rete stessa.

**Indirizzo Broadcast** → indirizzo IP dove i bit della parte host sono pari a 1. Riservato a tutti gli host della rete.

Il **Piano di Indirizzamento IP** è il documento che l'Amministratore di Rete deve scrivere e tenere aggiornato per descrivere l'utilizzo del proprio spazio di indirizzamento IP: indirizzi utilizzati, da chi e perché.

## CONFIGURAZIONE IP

Per configurare un host IP bisogna specificare:

- Indirizzo IP
- Subnet Mask
- Default Gateway
- Indirizzo IP del DNS

Alternativamente si può utilizzare il protocollo **DHCP** (più adatto per grandi organizzazioni perché non devo intervenire sui client).

Se si sbaglia la configurazione del **Default Gateway** non è possibile comunicare con altre reti, ma solo nella nostra rete locale.

Se si sbaglia la configurazione delle **Subnet Mask** si possono verificare problemi di connettività.

## PERCHE' FARE IL SUBNETTING?

- Superamento limiti di distanza
- Connessioni di reti fisiche diverse
- Filtro del traffico tra reti: traffico locale rimane nella sottorete locale.
- Amministrazione
- Visibilità di Strutture
- Isolamento del Traffico
- Ottimizzazione dell'uso dello spazio di indirizzamento IP
- Limitazione del dominio di Broadcast IP
- Limitare gli effetti di eventuali malfunzionamenti

# ARP

Un pacchetto che arriva dal livello di Rete(3) e passa al livello Data Link(2) deve inserire nella sua header l'indirizzo fisico MAC del destinatario, in modo che possa arrivarci. Infatti un host può comunicare con un altro host solo se ne conosce l'indirizzo fisico del protocollo di rete locale (es.: indirizzo Ethernet).

Se l'host di destinazione fa parte della stessa rete del mittente, bisogna conoscere l'indirizzo MAC, altrimenti bisogna conoscere l'indirizzo MAC del Gateway.

Gli indirizzi fisici vengono preassegnati dai produttori hardware e sono specifici della scheda di rete.

Dal momento che in genere i programmi applicativi conoscono solo il nome dell'host o il suo indirizzo in IP, il protocollo ARP permette di tradurre questi nomi/indirizzi-IP in indirizzi MAC.

Esiste, inoltre, una cache in ogni macchina che memorizza gli indirizzi risolti via protocollo ARP per le consultazioni successive. Queste informazioni, però, possono diventare obsolete in qualsiasi momento, ecco perché si utilizzano dei Timer che fanno scadere la validità dell'informazione (soft-state).

Se A vuole inviare un messaggio a B (che si trova sulla sua stessa rete locale), manderà un messaggio broadcast a tutte le macchine della sua rete dove specificherà: il suo indirizzo-IP e il suo indirizzo MAC, e poi chiederà qual è l'indirizzo MAC corrispondente all'indirizzo-IP di B.

In questo modo tutti gli host che ricevono il pacchetto aggiorneranno le informazioni nella cache.

Problemi:

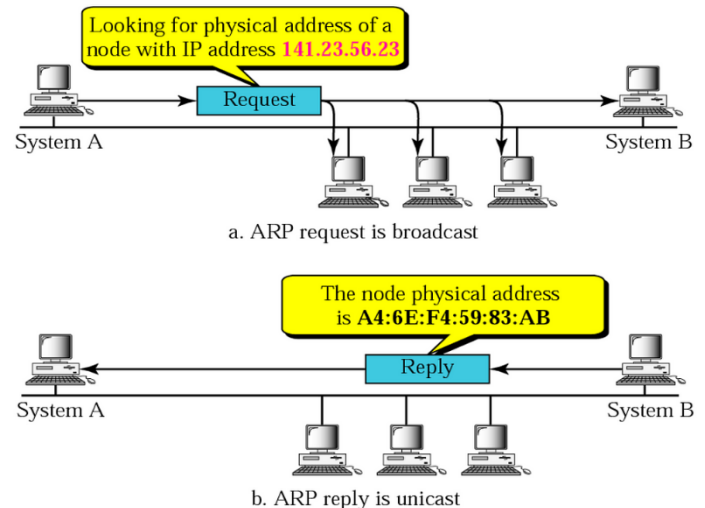
- Il meccanismo di richiesta via Broadcast può generare problemi (errori hardware, metodo best-effort di Ethernet)
- L'aggiornamento dei dati in cache può portare a ritardi (jitter)
- Impatto sull'operatività di altri protocolli in presenza di richieste ARP pendenti

# RARP

Converte l'indirizzo MAC in indirizzo IP.

Utilizzato dalle workstation diskless (non hanno capacità di memoria, quindi non conoscono il proprio indirizzo IP) che devono caricare il sistema operativo e la configurazione da uno o più server, a cui manderanno le richieste RARP per conoscere il proprio indirizzo IP.

Ormai obsoleto, perché rimpiazzato dal protocollo BOOTP e dal suo successore DHCP, che offrono più funzionalità.







# ICMP

Internet Control Message Protocol.

E' stato progettato per riportare anomalie che accadono nel routing dei pacchetti IP e per verificare lo stato della rete. (Genera messaggi d'errore)

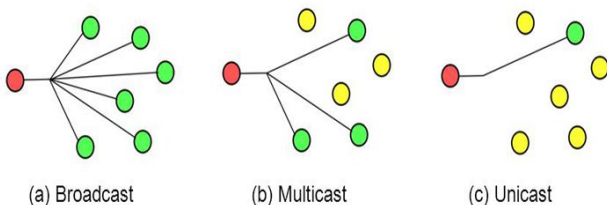
Tipi di messaggi ICMP:

-  **Echo Request** → Ping
-  **Echo Reply** → risposta al Ping
-  **Redirect** → invita ad un istradamento migliore dei pacchetti in quanto un router è stato attraversato inutilmente (ha dovuto ritrasmettere il messaggio sulla stessa rete da cui lo ha ricevuto). Quando un host riceve un pacchetto di Redirect associa un router diverso da quello di Default a quella destinazione.
-  **Mask Request e Address Mask Reply** → per scoprire automaticamente la Netmask usata in quella rete.

## IP MULTICASTING

Consiste nell'invio di un datagram ad un gruppo di host identificato da un indirizzo IP di destinazione.

Un datagram Multicast è inviato a tutti i membri del suo gruppo di host di destinazione con la stessa affidabilità Best-Effort dei datagrammi Unicast, quindi non è detto che arrivi a tutti i membri del gruppo o nello stesso ordine rispetto ad altri datagrammi.



Gli host possono aggiungersi o uscire dal gruppo senza limitazioni, ed è possibile definire una chiave di accesso che renda l'ingresso nel gruppo selettivo.

Un gruppo può essere Permanente o Transitorio.

Nel caso di gruppi Transitori spetta al Multicast Agent (entità che gira sui router o host speciali) mantenere le informazioni relative alla composizione del gruppo.

I Multicast Agents sono anche responsabili dell'invio dei datagram Multicast. Se un host del gruppo si trova in una rete diversa rispetto a quella degli altri host, il Multicast Agent diventa egli stesso destinatario del datagram e lo consegna ad altri Agents fino alla consegna all'host destinatario.

## IGMP

Nato dall'ICMP, ma serve per la gestione delle reti Multicast, supporta infatti le funzioni di IP Multicasting, consentendo ad un host di creare, unirsi ad un gruppo Multicast o di abbandonarlo.

Consente anche di inviare un datagram IP ad un gruppo di host.

# UDP

User Datagram Protocol.

È un protocollo di trasporto molto semplice, che offre due servizi all'IP:

1. **Multiplexing e Demultiplexing**
2. **Controllo dell'errore sui Dati:**

Checksum, che contiene l'IP del destinatario per intercettare false consegne di pacchetti. Il destinatario poi ricalcolerà il Checksum e se il risultato sarà errato cancellerà il pacchetto senza segnalare errori.

L'UDP è un protocollo di consegna non affidabile e senza connessione: non c'è l'Handshaking, quindi non si crea una sincronizzazione tra mittente e destinatario. Questo vuol dire che se un pacchetto viene perso non potrà essere ritrasmesso.

Non c'è Controllo di Flusso, ovvero un controllo di quanto il destinatario può ricevere per non sovraccaricarlo, n'è controllo della Congestione della rete.

È più veloce del TCP ecco perché viene usato per applicazioni che richiedono velocità e dove la perdita di un pacchetto non è di fondamentale importanza (multimedia).

Un'applicazione che usa UDP, infatti, accetta l'intera responsabilità di gestire il problema dell'affidabilità che comprende: perdita dei pacchetti, ritardo, consegna fuori ordine, ecc...

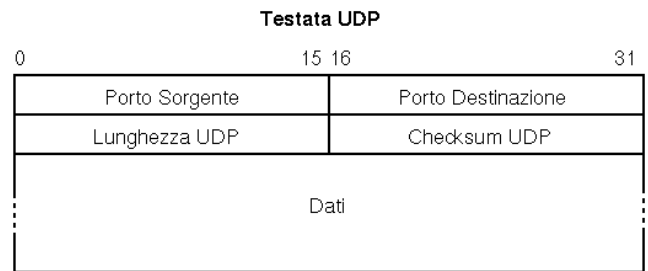
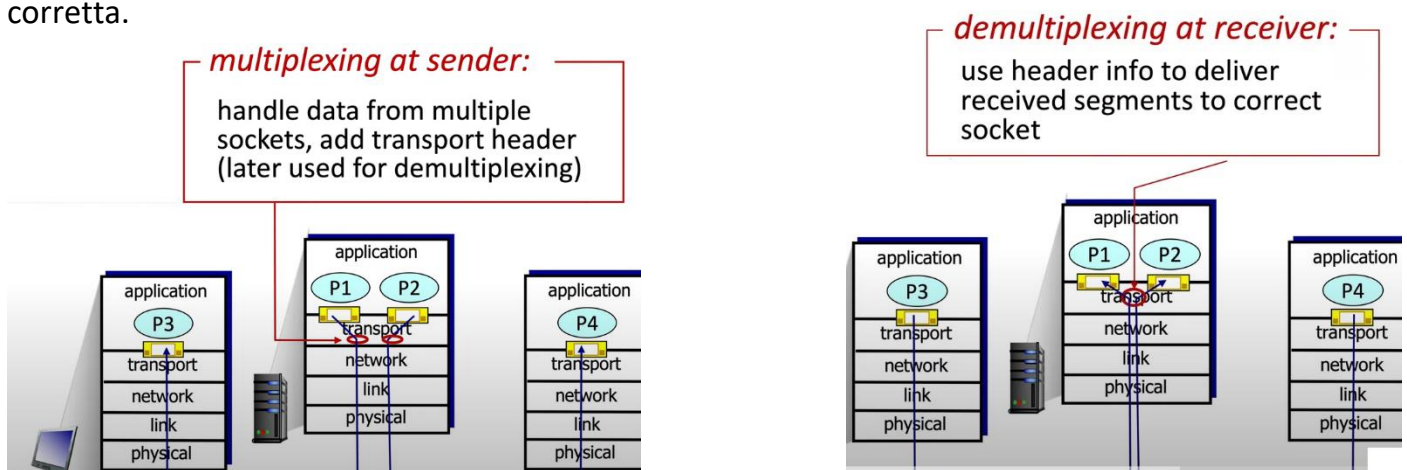
Generalmente applicazioni che usano UDP funzionano bene in ambito locale e falliscono quando utilizzati attraverso un'internet di dimensioni maggiori.

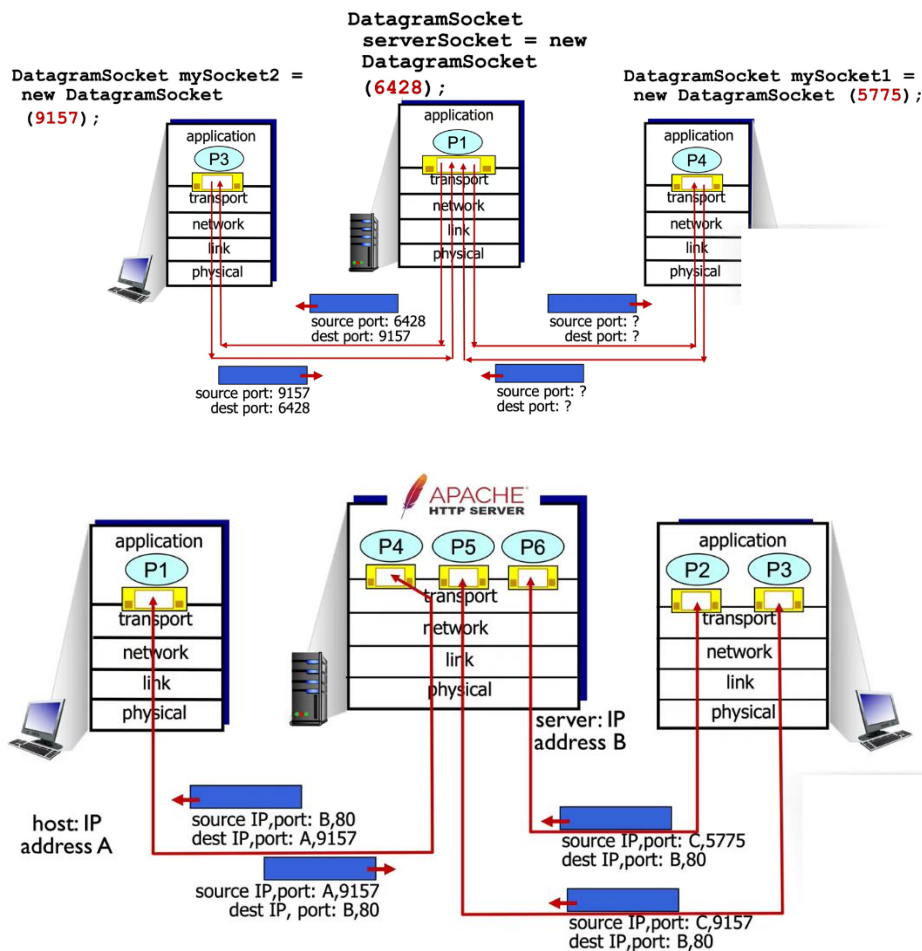
## Multiplexing e Demultiplexing

Su una macchina generalmente ci sono più richieste inviate dalle Applicazioni del livello 7 (diversi processi e quindi diverse Socket).

Il Multiplexing gestisce i dati che arrivano da più Socket e gli aggiunge la Header del livello di Trasporto, che verrà poi usata per fare il Demultiplexing in ricezione (l'header contiene le porte).

In ricezione, per assicurarsi che ogni risposta arrivi alla Socket del processo che l'ha richiesta, il Demultiplexing andrà a leggere la Header di Trasporto e consegnerà i pacchetti ricevuti alla Socket corretta.





Nell'UDP queste operazioni sono **Connection-less**, questo vuol dire che i pacchetti UDP che arrivano allo stesso host e alla stessa porta, verranno tutti consegnati alla stessa Socket indifferentemente da quale host e porta remota arrivano. Questo perché, come sappiamo, una Socket UDP è costituita da una Porta Sorgente e una Porta Destinazione. Non essendoci quindi gli indirizzi IP specifici degli host mittenti (come invece succede nel TCP), il processo che invia le risposte non può distinguere i vari host remoti che gli hanno fatto le richieste. Ecco perché chi invierà la risposta deve tenere a mente la socket UDP di chi fa la

richiesta in modo da inviare la risposta al mittente corretto, dato che tutte le richieste gli arrivano sulla stessa sua Socket.

TCP invece può avere più socket su un'unica porta, sempre che gli indirizzi IP siano diversi e specifichino determinati host.

# TCP

Trasmission Control Protocol.

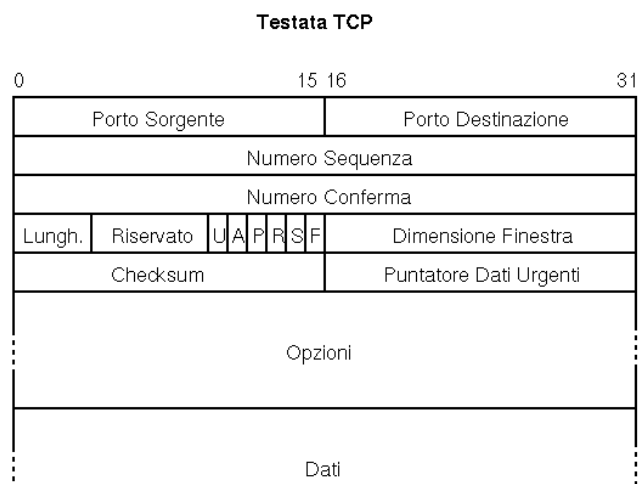
È un protocollo di trasporto con **consegna Affidabile** delle informazioni e con **Connessione**.

L'Affidabilità è garantita dall'invio degli **ACK** una volta ricevuto il pacchetto, tuttavia ciò può risultare pesante computazionalmente ed ecco perché si usa la tecnica della **Finestra Scorrevole**: ovvero si possono mandare più pacchetti senza ricevere l'ACK fino a quando rientriamo in una finestra predefinita di stream.

C'è il **controllo dell'Errore** con il Checksum, il **controllo del Flusso** e il controllo della **Congestione**. È più lento dell'UDP ma molti protocolli preferiscono usare TCP per la sua affidabilità.

Il TCP è:

- **Orientato allo stream** → i dati trasferiti vengono immagazzinati come sequenze di bit (stream) suddivisi in byte. Il servizio di consegna passa dal mittente al destinatario la stessa sequenza di ottetti.
- **Connessione di Circuito Virtuale** → solo dopo aver verificato la sussistenza delle condizioni necessarie ha inizio il trasferimento.
- **Trasferimento Bufferizzato** → il TCP mittente apre un Buffer dove salva l'info divisa in pacchetti (oppure raggruppata in ottetti, se è troppo piccola), solo dopo inizia il trasferimento dei singoli pacchetti e il TCP destinatario li salverà mano a mano nel suo Buffer. Se qualche pacchetto dovesse perdersi, il ricevente potrà comunicarlo al mittente e aspettare il suo re-invio.
- **Stream non strutturato** → non rispetta eventuali strutture presenti in dati strutturati, sono i programmi che usano il servizio di trasferimento che devono comprendere la struttura dei dati trasmessi.
- **Connessione Full-Duplex** → trasferimento simultaneo in entrambe le direzioni.



Per **gestire la connessione**, dato che alcuni pacchetti relativi a comunicazioni precedenti potrebbero arrivare in ritardo, il TCP utilizza un Initial Sequence Number (**ISN**), un campo da 32 bit scelto casualmente durante il setup della connessione. Successivamente tutti i numeri dei pacchetti saranno in ordine crescente, partendo da quell'ISN.

Il ricevente, quando riceve un pacchetto, risponde con **ACK+il Sequence Number** del prossimo byte che deve ricevere.

Inoltre, il TCP, esegue un periodo di time-out alla fine della connessione chiamato Maximum Segment Lifetime (**MSL**), così che tutti i pacchetti vecchi "muoiano" prima di iniziare la prossima connessione.

# ROUTING

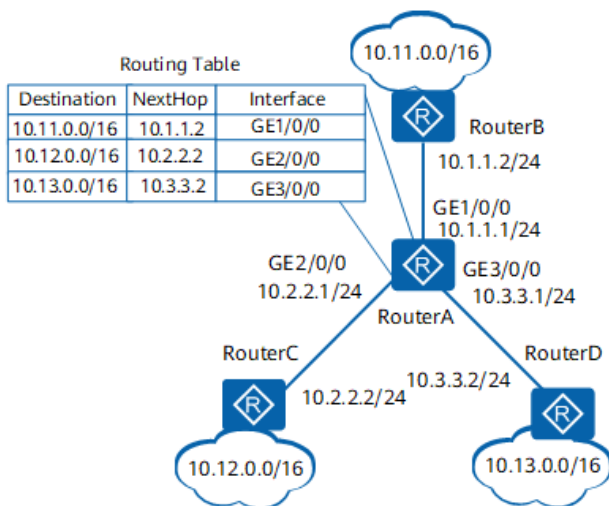
Il Routing (liv. 3) è l'azione di scambiare informazioni in rete da una sorgente ad una destinazione incontrando almeno un nodo intermedio.

Si occupa di due attività basilari: determinare il **percorso ottimale** di routing, e **trasportare i pacchetti** attraverso la rete.

Per stabilire il percorso ottimale si utilizzano le **Routing Metrics**, ovvero una modalità standard di misura che si può basare su diverse caratteristiche:

- Lunghezza del Percorso
- Affidabilità
- Ritardo
- Larghezza di Banda
- Carico
- Costo di Comunicazione

Un router considera l'associazione **Destination-Next\_Hop** per calcolare quale sistema intermedio rappresenti il miglior percorso.



I router comunicano tra loro e mantengono aggiornate le **Routing Tables** mediante la trasmissione di vari messaggi come:

➤ **Routing Update** → messaggio che contiene tutta o parte di una tabella di routing. Analizzando questi messaggi un router è in grado di costruire un disegno dettagliato della topologia della rete.

➤ **Link-State Advertisement** → messaggio che informa i router che usano il protocollo OSPF dello stato del link del mittente. Ogni volta che un link cambia stato il router lo comunica ai router vicini, in modo che questi possano poi calcolare il percorso migliore per una determinata destinazione.

Le **Routing Tables** vengono usate dai router per capire a quale altro router mandare il pacchetto per farlo arrivare a destinazione seguendo il percorso migliore.

Esse ricevono informazioni da: i file di configurazione creati dall'Amministratore di Rete salvati sul disco della macchina, e da protocolli di routing dinamici di aggiornamento.

A differenza dei router, gli host tendono a mantenere congelata la loro tabella di routing, quindi non eseguono protocolli di routing dinamici.

Definizioni di Routing:

1. **Minimale:** operazioni minimo di aggiornamento della Tabella di routing effettuate durante la definizione dell'interfaccia.
2. **Statico:** l'istadamento viene gestito mediante informazioni di routing predefinite e costanti. E' sufficiente questa definizione in configurazioni dove la topologia della rete è molto semplice (rete connessa in un solo modo alla backbone).
3. **Dinamico:** l'istadamento viene gestito via software da protocolli di routing che adattano le informazioni di routing a tutti i cambiamenti della rete. I protocolli di routing usano dei



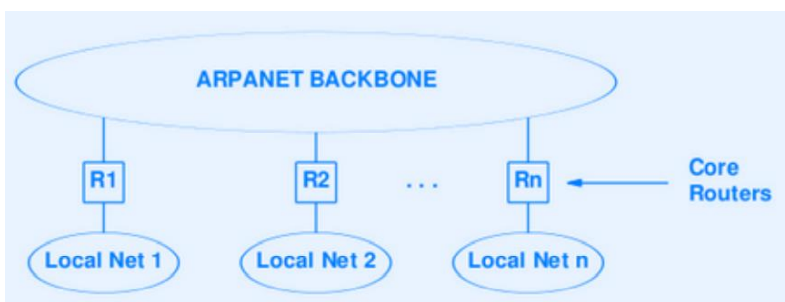
pacchetti per lo scambio delle info necessarie all'aggiornamento delle informazioni in tabella (i messaggi che abbiamo visto prima).

A seconda della complessità della configurazione, le informazioni di routing potranno provenire da tutti e tre i tipi di definizione. Ricordiamo però che, qual ora ci siano contraddizioni, le info date dal routing Dinamico hanno sopravvento su quelle Statiche.

Nel caso di **router Interni** ad un AS (Interior Gateway Protocols, IGP) le informazioni che un singolo router possiede sono parziali, in quanto esso conosce solo i percorsi per raggiungere le reti ad esso collegate e le routing statiche che indirizzano le reti collegate a dei gateway.

Questi router si affidano al **Default Router** per instradare correttamente i pacchetti che sono destinati a reti che loro non conoscono.

## ROUTING DELL'INTERNET ORIGINALE



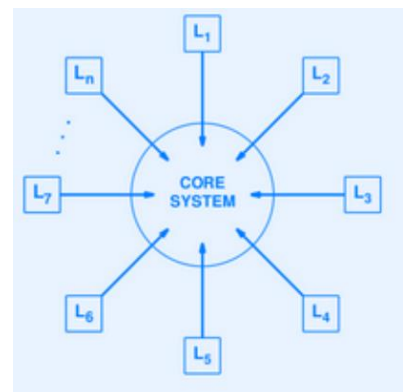
Costituita da una **Backbone** centrale e da una serie di router, ciascuno che connette una propria rete.

Se ogni router è oggetto di una default route può accadere nel peggiore dei casi che pacchetti destinati a reti inesistenti girino nella rete fino allo scadere del TTL.

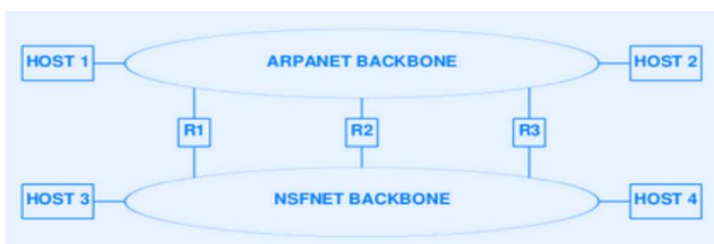
L'Architettura dell'Internet Iniziale prevedeva un insieme di Router Centrali (**Core Routers**) che conoscevano completamente le destinazioni di tutte le reti.

Gli altri router conoscevano le informazioni locali e utilizzavano i Core Router come router centrali.

Questo ovviamente poteva provocare dei **colli di bottiglia** nei Core Routers, inoltre non sono possibili **scorciatoie** e questo sistema non è **scalabile** con l'aumentare del traffico.



Un unico Core System diventa insufficiente al crescere degli ISP con proprie reti di dorsale, quindi nascono 2 dorsali con **NSF** (National Scientific Foundation) e **ARPA**, che sono diventate **Peer Backbones**, ovvero due strutture alla pari che comunicano attraverso dei router che le interconnettono.



Nel nuovo schema, tuttavia, non veniva supportato il **Partial Core**, quindi pacchetti destinati a reti inesistenti rimanevano intrappolati nella rete fino a che non scadeva il TTL.

Si è trovata una soluzione facendo sì che i router del Core System conoscano le destinazioni di tutte le reti.

Un meccanismo consente ai router di contattare i Core Router per conoscere le informazioni di routing. Analogamente viene implementato un algoritmo che consente ai router di apprendere gli aggiornamenti in modo automatico.



Questi algoritmi in particolare possono essere 2: Distance-Vector e Link-State.

## ALGORITMI DI ROUTING

Gli Algoritmi di Routing devono essere Ottimali, quindi scegliere la strada migliore, e consumare meno risorse possibili.

Inoltre, devono comportarsi correttamente in condizioni inusuali e mai viste prima e adattarsi ad esse.

Vengono classificati in:

- Statici VS Dinamici
- Single-Path VS Multi-Path
- Piatti VS Gerarchici
- Host Intelligent VS Router Intelligent
- Intradomain VS Interdomain
- Link-State VS Distance-Vector

### Distance Vector

Basa la scelta del percorso ottimale in basa alla **distanza** sottoforma di **router** o **AS** da attraversare.

I router si scambiano informazioni che sono tuple: rete e distanza.

Una volta che il router riceve queste info, le inserisce nella **Routing Table** e ricalcola l'algoritmo.

Queste informazioni si aggiornano allo scadere di certi **timer**.

È possibile comunicare solo gli update o l'intera tabella di routing aggiornata.

### Link-State Update

Alternativo al Distance-Vector, in questo caso le informazioni di routing vengono aggiornate quando cambia lo **stato del link**.

Le info vengono diffuse via **broadcast** in modo da permettere a ciascun router di calcolare il cammino ottimale e di capire la topologia della rete così da calcolare anche **strade alternative** se un router dovesse non funzionare.

Periodicamente le coppie di router connessi verificano l'esistenza del link attivo tra loro e propagano lo stato dei link, tutti i router riceveranno gli update e ricalcoleranno i percorsi ottimali sulla base delle loro informazioni.

# Autonomous System

Per **dominio di routing** si intende l'insieme delle reti che sono soggette all'amministrazione ed al controllo di una stessa Organizzazione.

Il dominio di routing prende il nome di **AS** ed identifica la Politica di Routing adottata.

L'AS quindi condiziona il routing, consentendo ad un'organizzazione di attivare diverse policy.

Anche se ci possono essere diversi AS gestiti da un ISP, la sua politica di routing viene identificata da un Autonomous System Number (**ASN**) ufficialmente registrato.

Un ASN viene assegnato a ciascun AS per poter effettuare routing BGP ed identifica univocamente la rete ai fini del routing.

Fino al 2007 questi ASN erano dei numeri interi a **16bit**, denominati **asplain**.

Ora invece è stata adottata una nuova sintassi che prende il nome di **asdot**, **32bit**, ovvero due numeri separati da un punto: x.y

Se il primo numero è 0 (0.y) coincide con l'asplain.

**IANA** in passato ha assegnato l'**ASN 23456** come variabile nel caso in cui router BGP in grado di gestire la nuova sintassi asdot, comunicassero con router BGP di vecchia generazione. Ci sono anche altri ASN **riservati** e che quindi non possono essere usati dagli operatori.

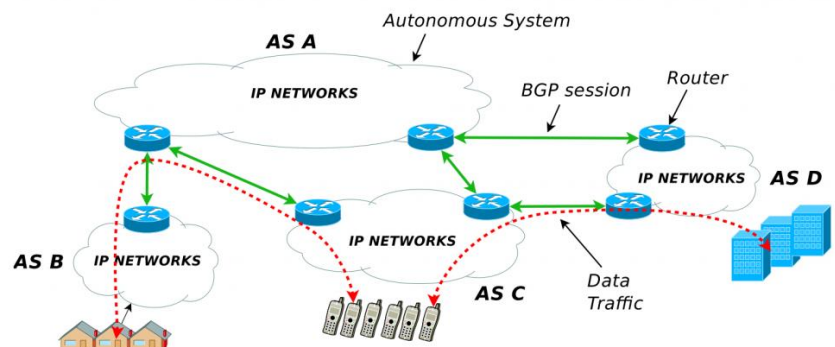
Ora gli ASN vengono assegnati da **ICANN**.

Una condizione necessaria per ottenere un ASN è quella di possedere due distinte connessioni ad Internet attraverso diversi ISP.

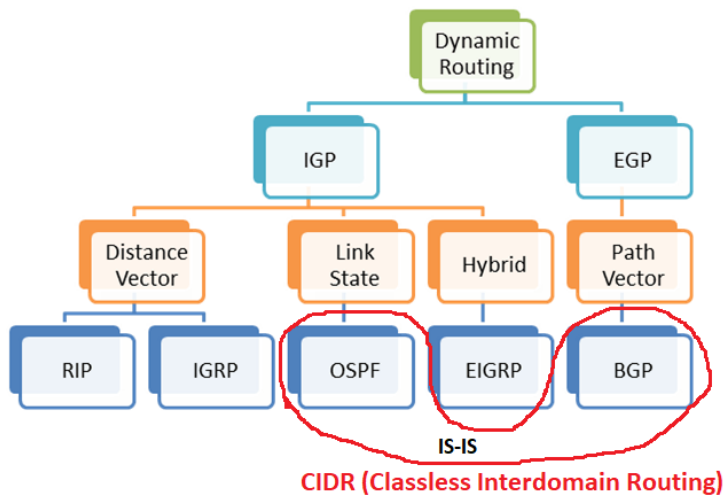
Gli AS si suddividono in:

- **Multihomed AS**: un AS che mantiene connessioni con più di un AS, in questo modo può rimanere connesso a Internet anche in presenza di un malfunzionamento in una delle connessioni.
- **Stub AS**: AS connesso solamente con un altro AS. Consente il private peering tra ISP.
- **Transit AS**: AS che fornisce attraverso di sé connessioni con altre reti. È un accordo di transito. La rete A può usare la rete B appartenente ad un Transit AS per connettersi alla rete C.

I router che istradano messaggi all'interno di uno stesso AS sono detti **Interior Router** (eseguono l'Interior Gateway Protocol **IGP**), mentre quelli che instradano messaggi anche tra AS diversi sono detti **Exterior Router** (eseguono l'Exterior Gateway Protocol **EGP**).



# Protocolli di Routing



## OSPF

Open Shortest Path First.

Protocollo Open Source **efficiente**, ma **complesso** da configurare e gestire.

- Supporta reti con **subnet diverse**
- Implementa **routing dinamico**
- Esegue il **bilanciamento del carico** quando instrada i pacchetti
- Supporta l'**autenticazione dei messaggi** (i

vari router si autenticano vicendevolmente)

- Supporta **sistemi gerarchici** (aree, dove l'autorità massima è sempre l'AS, non posso avere OSPF distribuito su più AS)
- Computazionalmente **pesante**

È un **Link-State Protocol** perché invia dei Link-State Advertisements (**LSA**) a tutti i router di una stessa area gerarchica, dove specifica informazioni relative alle **interfacce attive**, alle **metriche** usate, ed altre variabili.

I router di una stessa area hanno lo stesso **Topological Database**, quindi capiscono com'è fatta la rete. La suddivisione in aree, inoltre, riduce il traffico di routing nell'AS ed introduce il traffico di routing **Inter-Area** e **Intra-Area**.

L'OSPF **Backbone** sarà responsabile di distribuire le informazioni di routing tra aree.

Le aree possono essere definite in modo tale che la Backbone non sia continua fisicamente, ma si può rendere continua con dei **virtual-links** definiti tra i Backbone Routers. Questo permette ai gestori di rete di definire una topologia logica differente da quella fisica.

OSPF può apprendere informazioni da altri Exterior Gateway, Interior Gateway o mediante istruzioni di Configurazione.

OSPF distingue 4 tipi di router:

1. **Internal Router**: interni ad un'area
2. **Area Border Router**: che connettono 2 o più aree
3. **Backbone Router**: che appartengono alla dorsale (area 0)
4. **Border AS router**: router di confine tra AS (si trova vicino al router di frontiera, dove finisce l'AS e va verso il BGB)

Attraverso l'invio di pacchetti OSPF è possibile costruire una **mappa topologica** della rete così da vedere quali sono i router adiacenti, quelli di default e quelli di backup.

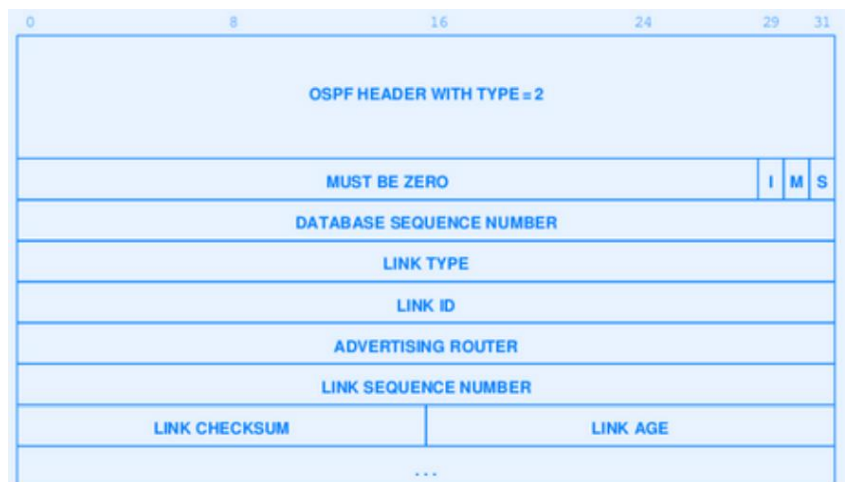
Solo router adiacenti possono scambiare tra loro informazioni.

0	7	15	31
Version	Type	Packet length	
Router ID			
Area ID			
Checksum		AuType	
Authentication			
Authentication			

I pacchetti OSPF sono:

- 🚦 **Hello** → per scoprire i neighbors
- 🚦 **Link State Update** → aggiornamento dello stato di un link
- 🚦 **Link State ACK** → conferma di aver ricevuto il Link State Update
- 🚦 **Database Description** → comunica gli aggiornamenti che conosce per inizializzare il Topological Database. Nello scambio uno dei due router funge da Master e l'altro da Slave, e lo Slave confermerà la ricezione di ogni messaggio con una risposta.\*
- 🚦 **Link State Update** → richiesta di informazioni di stato ai neighbor routers

\*Poiché questi messaggi possono essere molto estesi vengono utilizzati i **bit I** (impostato a 1 nel messaggio iniziale per far capire che è il primo) e **M** (impostato a 1 nei messaggi che seguono per far capire che più messaggi Database Description seguono questo qui). Il **bit S** indica se il messaggio è stato inviato ad un Master(1) o a uno Slave(0). Il **Database Sequence Number** numera in sequenza i messaggi, così da identificare eventuali messaggi persi. Poi con gli altri campi viene descritto il link (age, length), mentre con **Link Type** viene indicato il tipo di link.



## RIP

Routing Information Protocol. Usa **UDP** e la porta **520**.

Basato sull'algoritmo **Vettore-Distanza**, quindi meno sono i router o gli AS da attraversare, migliore è considerato il percorso.

RIP ha il limite di **15 hops**: reti più distanti sono considerate irraggiungibili.

È **poco efficiente** perché invia tutta la Routing Table, o parte di essa, ai router vicini ad intervalli di tempo regolari (30 sec), tuttavia è **più leggero** di OSPF ed è più **semplice** da implementare e gestire.

RIP ha due forme:

- **Attiva**: usata dai router. Invia in Broadcast periodicamente aggiornamenti di routing e usa i messaggi in arrivo per aggiornare la sua routing table.
- **Passiva**: usata dagli host. Usa i messaggi in arrivo per aggiornare la routing table e non invia aggiornamenti.

Gli aggiornamenti contengono coppie di valori: **Indirizzo\_destinazione-Distanza**.

**RIP v.1** non supporta le subnet variabili. **RIP v.2** permette di ridurre le informazioni inviate scegliendo di inviarle solo a determinati router invece che in Broadcast.

Con RIP v.2 inoltre si possono alterare localmente le info di routing apprese tramite il protocollo per favorire o sfavorire un determinato percorso rispetto agli altri.

Infine è stata introdotta l'**autenticazione**: router facenti parte di uno stesso dominio di routing si identificano l'uno con l'altro.

command(1)	version (1)	must be zero (2)
address family identifier (2)		must be zero (2)
IP address (4)		
must be zero (4)		
must be zero (4)		
metric (4)		

⋮

Per evitare oscillazioni tra percorsi di costo uguale, RIP specifica che le informazioni esistenti devono essere mantenute fino a quando non si abbia un percorso con **costo rigorosamente più basso**.

Se un router si guasta gli altri router cancelleranno le informazioni di routing relative allo scadere dei timer.

RIP deve gestire **3 errori** causati dall'algoritmo Vettore-Distanza:

1. Non vengono rilevati i **Routing Loops**
2. Deve usare un numero basso di passi come distanza massima (15) per evitare **instabilità**
3. L'algoritmo **converge lentamente**

Per evitare quest'ultimo problema si sono adottate le seguenti soluzioni:

- **Split horizon update**: il router non propaga informazioni al router che ha generato tale aggiornamento.
- **Hold Down**: il router ignora aggiornamenti inerenti a una rete per un certo tempo (60 sec), una volta che ha ricevuto un messaggio di rete irraggiungibile.
- **Poison Reverse**: quando un collegamento scompare il router continua ad annunciarlo ancora per un po' assegnandogli distanza infinita.
- **Triggered Update**: i router sono obbligati ad annunciare la scomparsa di route immediatamente senza attendere che i timer si azzerino.

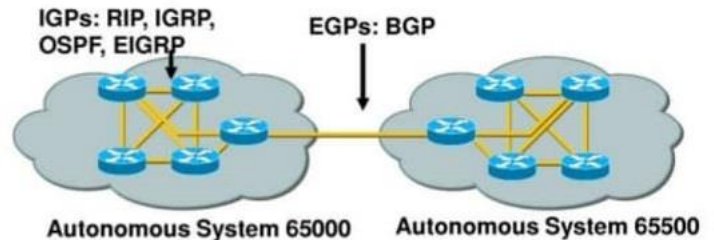
# BGP

Border Gateway Protocol.

Si basa sull'algoritmo **Vettore-Distanza** evoluto e si occupa del transito di dati di terze parti su una certa rete che vengono suddivise in: reti **stub** (unica connessione al grafo BGP), **multiconnesse**, di **transito** (generalmente reti backbone disponibili al transito di traffico di terze parti).

Questo protocollo viene utilizzato nei **router di frontiera**, ovvero quei router affacciati direttamente all'internet globale e che devono fare il routing con tutti gli indirizzi attivi di internet.

I router BGP hanno in memoria tutti i prefissi di rete che servono per indirizzare internet in un dato istante.



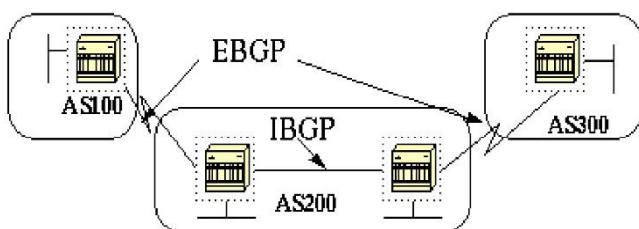
I router di internet devono essere **divisi in gruppi** perché altrimenti non esisterebbe un protocollo di routing in grado di scambiare informazioni di routing in modo efficiente se tutte le organizzazioni facessero parte di un'unica rete. Se il numero di router è grande, il traffico diventa insostenibile.

Inoltre, le reti e i router non possono essere gestiti tutti da una singola entità, ma da diverse entità che possono decidere le loro politiche di routing differenti.

Un'**architettura d'instradamento** deve fornire un modo perché ciascun gruppo controlli indipendentemente l'instradamento e l'accesso.

I **problemi** che impattano sulla capacità dei router di scambiarsi informazioni sono:

- Il **ritardo**: più sono i router più tempo ci mettono le informazioni di routing a propagarsi.
- L'**overhead**: dato che ogni router invia messaggi per aggiornare le informazioni di routing, più router sono coinvolti maggiore è il traffico. Poi, dato che le gli aggiornamenti contengono l'elenco delle possibili destinazioni, anche le dimensioni dei messaggi aumenta al crescere dei router.



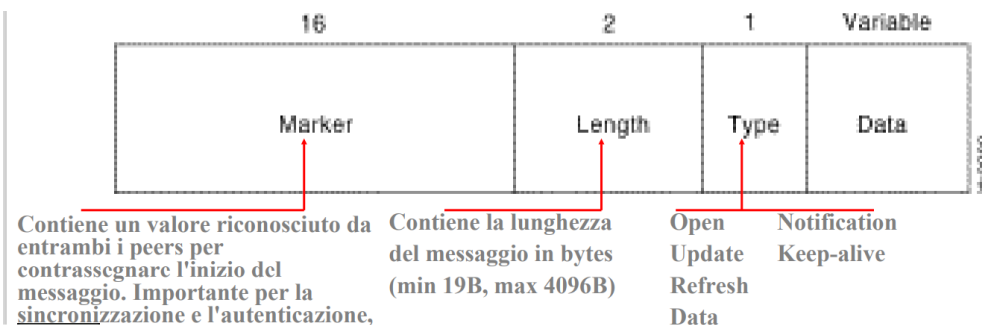
**Paradigma del Salto Successivo** → se un router esterno ad un gruppo sceglie come router di default un router partecipante ad un gruppo, egli manderà a questo router tutti pacchetti che escono dal suo gruppo, compresi quelli destinati ad altri gruppi che non sono quelli del router di default.

BGP permette quindi la comunicazione tra AS, utilizza **TCP** alla porta **179** e il suo trasporto affidabile con il quale diffonde le **informazioni di routing** e di **instradamento**, e info sulle **reti raggiungibili** attraverso degli **aggiornamenti incrementali** (non viene inviata tutta la Routing Table, ma solo un sottoinsieme di informazioni).

Supporta inoltre l'istradamento **classless** (/24 ad es.) e permette ai peer di **autenticarsi** l'uno con l'altro.

**Funzionamento BGP:** i peer si autenticano l'un l'altro, si scambiano dei messaggi per stabilire la correttezza delle operazioni e per sapere se entrambi sono disponibili a comunicare. Poi cominciano a inviarsi a vicenda le informazioni relative alle reti raggiungibili e non raggiungibili. Quando si collegano per la prima volta si scambiano l'intera tabella di routing per capire la topologia della rete, successivamente si scambieranno solo gli aggiornamenti. Infine si scambiano continuamente messaggi per verificare che i peer e la connessione di rete stanno funzionando correttamente. Per fare tutto ciò usa i seguenti messaggi:

- ✚ **Open** → è il primo messaggio inviato e apre una sessione BGP tra peers. Questo messaggio è confermato da un Keepalive prima che inizi lo scambio di messaggi ordinario.
- ✚ **Update** → aggiornamento del routing.
- ✚ **Notification** → inviato quando si verificano errori, è utilizzato per chiudere una sessione attiva e per avvisare i router connessi del perché la sessione viene chiusa.
- ✚ **Keepalive** → informa che il dispositivo è attivo in modo da non far chiudere la sessione.
- ✚ **Refresh** → richiede il re-invio delle informazioni di routing.



← I pacchetti BGP contengono una **Header** che lo scopo di identificare lo scopo del pacchetto in questione.

BGP effettua i seguenti tipi di routing:

- **Inter-AS** → tra due o più router appartenenti ad AS diversi. Usato per calcolare il percorso migliore attraverso internet.
- **Intra-AS** → tra due o più router appartenenti allo stesso AS.
- **Pass-Through-AS** → tra due o più router BGP che scambiano traffico attraverso un AS che non esegue BGP. L'AS attraversato permette di attraversare la sua rete anche se i dati non sono destinati a lui.

Ogni router BGP mantiene una **lista** di tutti i percorsi fattibili verso una particolare rete e non aggiorna le tabelle di routing fino a che non riceve un **aggiornamento incrementale**.

Inoltre mantengono un **numero di versione** della Routing Table che deve essere lo stesso per il rispettivo peer BGP. Il numero di versione cambia ogni volta che BGP aggiorna la Routing Table.

BGP usa una **Metric singola** che consiste in un numero che specifica il grado di preferenza per quel dato link, che viene assegnato dall'Amministratore di Rete. Questa preferenza può basarsi su varie cose: distanza, scalabilità, velocità, costo, ecc...



# IPv6

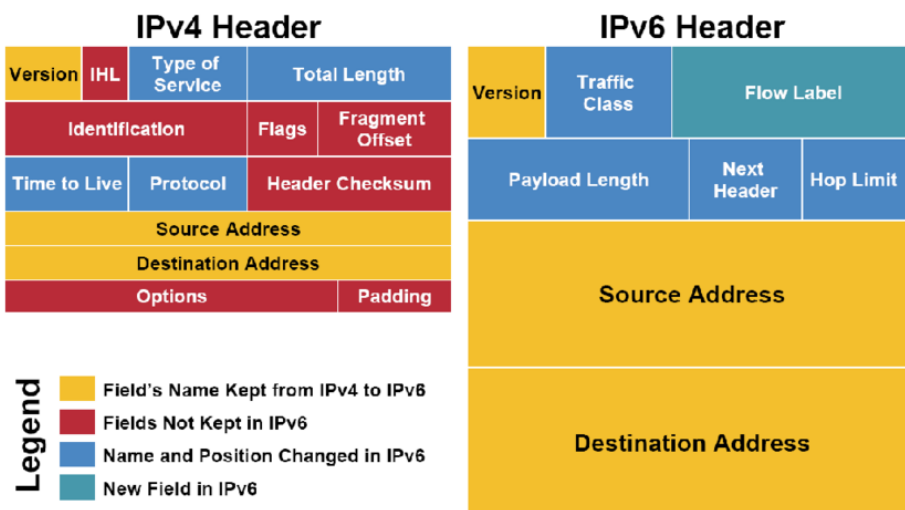
L'IPv6 è nato perché lo spazio di indirizzamento fornito dall'IPv4 è **limitato** e **non sufficiente**. Precisamente abbiamo solo **2<sup>32</sup>** indirizzi da poter assegnare a tutte le macchine esistenti al mondo.

Questo problema è stato mitigato dall'adozione di **Reti Private** in combinazione con i protocolli **NAT** e **DHCP** che permettono di tradurre tanti indirizzi privati, non instradabili dai router, in un unico indirizzo pubblico, quello appunto del router NAT.

L'IPv6 ci offre il **quadruplo** degli indirizzi dell'IPv4, dato che prevede degli indirizzi di **128bit** invece che 32bit. Dove ogni campo è composto da **16 bit** in notazione **HEX**: a:b:c:d:e:f:g:h

Campi successivi di zero sono rappresentati da :: ma solo una volta in un indirizzo, non di più.

Oltre ad uno spazio di indirizzamento più grande con l'IPv6 possiamo avere una migliore **gestione** del traffico IP, la possibilità di non avere più **host nascosti** e quindi ogni host può essere raggiungibile e fungere da **server**, e implementare sistemi di **sicurezza Punto-Punto**.



Inoltre l'**header IP** non ha dimensione fissa ma si può estendere migliorando la gestione delle opzioni.

Es.:

IPv6 H. → Routing H. → ESP

H. → TCP H. → Dati

Avere **meno campi** nell'header rende il routing più efficiente e migliora le prestazioni.

## Tipi di Indirizzi

- **Unicast**: primi 64 bit rete, ultimi 64 bit host.
  - **Unspecified**: indica l'assenza di indirizzo, può essere usato nella richiesta iniziale DHCP per ricavare la Default Route. :: /0
  - **Loopback**: identifica l'host stesso, il localhost, come 127.0.0.1 in IPv4, e ::1 in IPv6.
  - **Indirizzi Scoped**: hanno valore solo in un certo scope.
  - **Link-Local**: è un indirizzo Scoped, ha infatti valore solo nelle LAN o VLAN e non può essere instradato dai router. Viene automaticamente configurato su ogni interfaccia in base al MAC. FE:0:0:0:MAC
  - **Site-Local**: è un'indirizzo Scoped, utilizzato solo tra nodi della stessa rete di link, simile agli indirizzi privati IPv4. Non viene configurato di default. Generalmente usato per numerare un sito prima di connetterlo ad internet, oppure per indirizzamento privato. FEC0:0:0:SubnetID:MAC
- **Agreeable Global**: la politica di assegnazione degli indirizzi deve essere modificata dato che con l'IPv6 abbiamo molti più indirizzi:  
/23 per i Regional Registries



/35 per i Local Internet Registries

/48 per le Organizzazioni (utenti finali)

/64 per le sottoreti degli utenti

- **Multicast:** per mandare un pacchetto ad un gruppo di utenti.  
FF <flags><scope>::- **Anycast:** un pacchetto inviato ad un indirizzo Anycast viene inviato al componente del gruppo più vicino, questi indirizzi infatti vengono assegnati ad un gruppo di interfacce. Serve generalmente per funzioni di Discovery e non sono distinguibili dagli indirizzi Unicast (perché allocati dallo stesso spazio di indirizzamento Unicast) ecco perché bisogna esplicitamente scrivere Anycast dopo l'indirizzo Unicast. Alcuni indirizzi sono riservati per usi specifici.  
**Es.:** usato per scoprire quale server DNS è più vicino a me per potergli inviare le mie query.

Dato che ogni nodo ha **più indirizzi IPv6** la scelta di quali indirizzi usare come sorgente e destinazione per ogni invio di dati viene fatta in base allo **scope** più adatto alla destinazione (global, site, local) e all'**indirizzo** più simile alla destinazione (IPv4 o IPv6). L'**algoritmo di scelta** può essere sovrascritto dall'Applicazione o dai protocolli del **TCP/IP**.

# TELNET (RFC 854-855)

Telnet è un servizio di rete di **emulazione di un terminale** a carattere **ASCII** che può essere acceduto **remotamente** attraverso la rete.

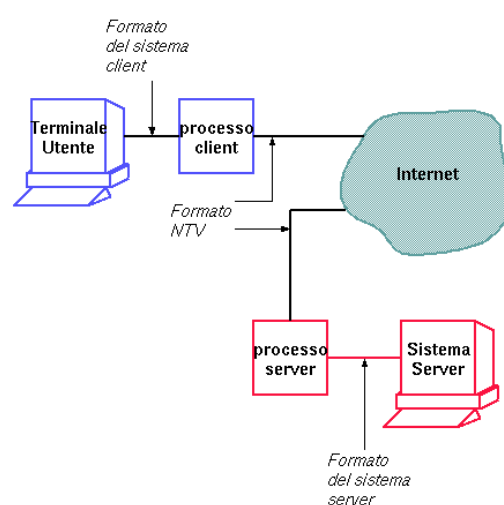
Si basa su **TCP**, quindi ha una connessione affidabile, ed è strutturato sul paradigma **Client/Server**:

- Il **Client** è invocato dall'utente e realizza una connessione col Server Telnet remoto che si trova in esecuzione su un altro host, in ascolto sulla **porta 23**.  
Appena connesso l'utente può eseguire comandi come se il terminale fosse locale.
- Il **Server** quindi accetta le connessioni di rete e passa i caratteri digitati dall'utente al sistema operativo, come se fossero digitati in una tastiera locale. Poi invia l'output sulla connessione del client.

Questo servizio si basa su 3 aspetti:

1. **Network Virtual Terminal (NVT)** → un terminale virtuale con caratteristiche generali. Ogni Server o Client traduce i controlli nativi in quelli del NVT, quindi non è necessario usare specifici strumenti per sfruttare questo servizio.
2. **Opzioni Negoziare** → tra client e server per aumentare le funzionalità della sessione Telnet da aprire.
3. **Viste Simmetriche** → ai lati della comunicazione ci sono dei programmi invece di una tastiera ed un monitor fisici. Quindi sono i programmi che emulano le componenti HW.

Il Formato *Network Virtual Terminal* usato da Telnet



Essendo un servizio funzionante su **sistemi eterogenei**, i client e i server possono essere implementati in modo diverso.

Telnet è ormai obsoleto perché prevede l'invio di pw e username in chiaro. Al suo posto si usa Secure Shell (**SSH**) che, invece, crittografa queste informazioni.

## COMANDI R (remote)

### rlogin

Alternativa al Telnet dato che svolge funzioni analoghe.

È stato inventato per sistemi **UNIX** e permette all'Amministratore di configurare una serie di macchine in modo tale che se un utente ha uno stesso identificativo in queste macchine, l'accesso avvenga senza digitare la password.

Questo facilita la configurazione di ambienti distribuiti, tuttavia comporta dei **problemi di sicurezza**.

## rsh

Simile a rlogin, anch'esso è per i sistemi **UNIX**.

Permette l'esecuzione remota di un singolo comando, il cui esito verrà visualizzato nella finestra dell'utente nel sistema locale, che quindi deve prevedere un'applicazione che consenta l'accesso da terminale a carattere.

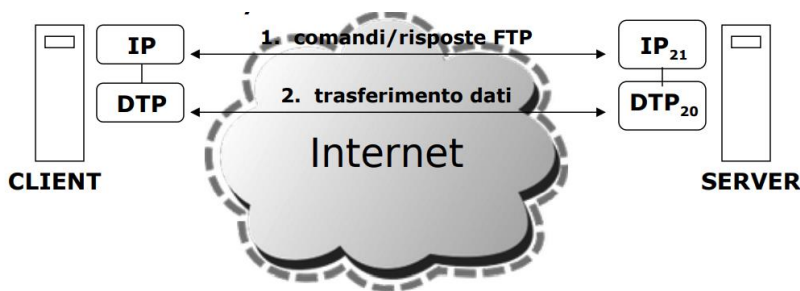
## FTP

File Transfer Protocol.

Protocollo per il trasferimento di file tra host in una rete TCP/IP. Usando **TCP** quindi è orientato alla connessione ed è affidabile.

In ogni trasferimento avvengono 2 processi bidirezionali:

1. **Data Transfer Process (DTP)** → si occupa del trasferimento dei dati vero e proprio tra client e server. Quando gli arriva una richiesta di trasferimento dati, quindi, apre una connessione alla **Porta 20** con il DTP del client.
2. **Protocol Interpreter (IP)** → si occupa di trasmettere comandi fra client e server (dà inizio al processo FTP). **Porta 21**. Rimane attiva durante la sessione DTP.



In caso di **OS diversi** si deve specificare la **codifica**, così che FTP possa tradurre opportunamente i file per l'altro OS. Se però devo trasferire altri tipi di file (immagini, eseguibili, ecc) devo specificare a FTP che non deve fare niente altrimenti renderebbe i file non più leggibili.

È possibile anche effettuare delle **sessioni anonime**: l'account usato è **anonymous**, la pw generalmente è l'email, in questo caso però i client hanno solo diritto di **lettura** cosa che evita il caricamento di file non autorizzati sul server.

### Client

- Contatta il server
- Specifica i file
- Specifica la direzione del trasferimento (upload/download)

### Server

- Mantiene un insieme di file nel disco locale
- Rimane in attesa di richieste di connessione
- Onora le richieste dei client.

## TRIVIAL FTP (TFTP)

È una variante del protocollo FTP che ha un **codice più leggero**, inteso per essere usato nelle **LAN**. Si esegue usando **UDP** ed è quindi molto più **veloce** di FTP.

Usato generalmente dalle macchine **diskless** quando devono scaricare il sistema operativo.

# CRITTOGRAFIA

## Crittografia Simmetrica

Più **leggera** e **veloce** di quella Asimmetrica, ma **meno sicura**.

Mittente e Destinatario sono entrambi a conoscenza di una **singola chiave** con cui è stato crittografato il messaggio che gli permetterà di decifrarlo.

Algoritmi: **DES**, **3DES** e **AES**.

## Crittografia Asimmetrica

Molto **pesante** computazionalmente ecco perché non si può crittografare tutto il messaggio con questo tipo di crittografia, ma si crittografa solo un riassunto del messaggio (MAC). Quello che succede, generalmente, è che il messaggio viene crittografato con la crittografia simmetrica e poi la chiave simmetrica viene crittografata con crittografia Asimmetrica.

Il Mittente e il Destinatario entrambi possiedono una **coppia di chiavi univoca**: chiave **pubblica** (conosciuta da tutti) e chiave **privata** (che conosce solo il proprietario). Queste due chiavi sono strettamente legate nella crittografia e decrittografia (posso decifrare messaggi criptati con la chiave pubblica usando la chiave privata e viceversa).

Garantisce **autenticazione**, **integrità** e **confidenzialità** del messaggio.

Algoritmi: **RSA**, **DSA** e **DH**.

# SSH

Secure Shell.

Ha sostituito **Telnet** perché più sicuro e serve per eseguire comandi/programmi su una macchina remota attraverso una linea di comando.

**SSH v.1** utilizza l'algoritmo di crittografia asimmetrica **RSA** per la negoziazione delle chiavi e gli algoritmi simmetrici **3DES** e **AES** per la crittografia dei dati.

Inoltre utilizza il **CRC** (Cyclic Redundancy Check) per verificare l'**integrità** dei dati.

**SSH v.2** usa gli algoritmi asimmetrici **DSA** (Digital Signature Algorithm) e **DH** (Diffie Helmann).

Al posto del CRC si usa l'algoritmo **HMAC** (Keyed-Hash Message Authentication Code).

SSH memorizza in **file ASCII** le **chiavi pubbliche e private** nella directory `$HOME/.ssh`.

Nella stessa cartella viene salvato anche il file `known_hosts` che contiene le **chiavi pubbliche** dei server ai quali ci si collega.

Per gli host le cui chiavi pubbliche sono salvate nel file `authorized_keys`, al momento del login non viene richiesta la pw.

Alcuni software SSH consentono di gestire le chiavi pubbliche e private mediante un certificato **X.509**.

**Port Forwarding locale e remoto.**

# DNS

Domain Name System.

Gestisce la risoluzione **nome-indirizzo IP**. Porta **53**, **UDP** per le query, **TCP** per trasferire gli zone file dal primary server al secondary.

Utilizza una nomenclatura di tipo **gerarchica** che va da dx verso sx. A dx avremo il nome più grande (il top level domain).

[www.dmi.unipg.it](http://www.dmi.unipg.it) → applicazione, sottodominio, dominio, top level domain (TLD).

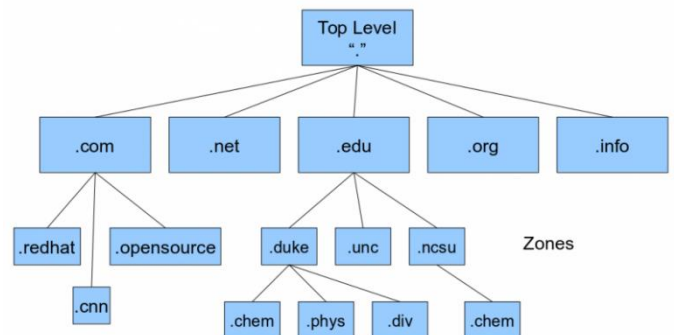
Internet usa quindi uno spazio dei nomi gerarchico, suddiviso per organizzazione.

I TLD vengono definiti da **ICANN** e sono tutti figli della **root** (il punto .).

**RFC 2606** definisce dei **domini riservati**: .example, .invalid, .localhost, .test .

I TLD si dividono in 3 categorie:

- **Country-Code TLD**: lista delle sigle dei paesi del mondo (.it, .fr, .uk, ecc...).
- **Generic TLD**: usati da particolari organizzazioni (.com, .mil, .edu, .gov).
- **Infrastrutturali**: l'unico è .arpa, usato nella risoluzione inversa dei nomi.



A livello mondiale ci sono una 30ina di DNS che hanno l'autorità di gestire la root (.) e che quindi hanno un record **SOA** (Start Of Authority) che gli sancisce questa autorità.

Tutti i DNS autoritativi per un certo dominio hanno scritto nella tabella DNS il record SOA.

L'autoritativo del dominio più esterno delega il dominio più interno, quindi:

.it delega .unipg, e .unipg delega .dmi

## DNS DATABASE

È costituito da:

- **Record**
- **Nomi**: può essere associato a diverse classi (host, mail exchanger, server DNS, ecc...).
- **Classi**

È possibile associare ad ogni interfaccia di rete un nome (hostname) dato che questi sono più facili da ricordare degli indirizzi IP.

L'assegnazione di un nome ad un indirizzo è responsabilità dell'Amministratore di Rete.

## Risoluzione nome-indirizzoIP

- **Statica** → le associazioni nomi-indirizzo vengono definite una volta per tutte in una **host table**, un file ASCII.
- **Dinamica** → le associazioni nomi-indirizzoIP vengono stabilite dinamicamente da richieste fatte ai **server DNS**.

# DNS BIND

In UNIX esiste un'implementazione del DNS: il DNS BIND (Berkeley Internet Name Domain).

BIND è quindi un package di sw che comprende i principali componenti del DNS (**named**, **resolver**) e strumenti per la verifica del corretto funzionamento del server DNS (**nslookup**).

- **resolver** (BIND client) → è una libreria di funzioni che permette di generare e inviare al server delle richieste.
- **named** (BIND server) → è un processo demone in grado di servire le richieste del resolver.

La configurazione del BIND (sia client che server) avviene tramite dei **file di testo** che ne descrivono il servizio fornito.

**Config. resolver:** funzioni del resolver sono configurate nel file **/etc/resolv.conf** che viene letto all'avvio del processo che usa il resolver.

Le voci da inserire in questo file sono: **nameserver address**, **domain name**, **search domain**.

Le richieste vengono inviate al nameserver specificato. Il domain definisce il nome di dominio di default, il resolver aggiunge name a qualsiasi nome host che non contiene il carattere punto. Search ha la stessa funzione di domain con la possibilità di avere più domini da provare ad aggiungere al nome dell'host. In questo caso viene aggiunto solo l'intero nome dei domini indicati.

**Config. named:** a differenza di resolver, servono più file per configurare named.

**named.conf** → parametri generali di configurazione del named ed i puntatori agli zone files (file contenenti le info dei domini gestiti dal server).

**named.ca** → puntatori ai root domains server.

**named.local** → reverse per l'indirizzo loopback (permette la risoluzione dell'IP 127.0.0.1 nel nome localhost).

**named.hosts** → zone file per la risoluzione diretta. Permette la conversione di hostname in indirizzo IP.

**named.rev** → zone file per la risoluzione inversa.

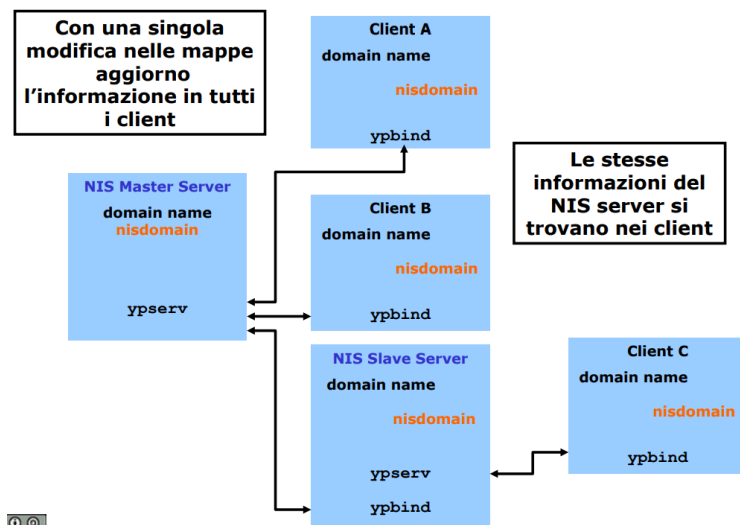
Inoltre BIND può essere configurato in 3 modi diversi:

1. **Caching-Only:** ogni richiesta viene rediretta su altri server ed il risultato è memorizzato in una **cache locale** per servire future richieste.
2. **Primary:** è l'**authoritative** server di informazioni riguardanti specifici domini. Legge le info dagli **zone files**.
3. **Secondary:** scarica gli zone files dal Primary server e li memorizza localmente in appositi file detti **zone file transfer**.

I **Record** che possono avere gli **zone file** sono:

- **SOA**
- **NS** → name-server
- **A** → usato per associare un hostname ad un indirizzo IPv4
- **AAAA** → usato per associare un hostname ad un indirizzo IPv6
- **MX** → server che gestisce la posta per un host o un dominio
- **CNAME** (alias per il nome di un host)

# NIS



Network Information Service (o Yellow Pages).

È un servizio che realizza un database di amministrazione che permette un **controllo centralizzato** di **risorse** e la loro **condivisione automatica**. Quindi un utente può spostarsi da un host all'altro mantenendo però lo stesso login, pw, home directory e autorizzazioni possedute.

È fondamentale per **applicazioni parallele e distribuite** e per la creazione di cluster di

computer perché consente di usare una notazione semplificata per l'accesso a file e risorse.

Converte i principali file UNIX in un **formato database** detto **NIS map** che può essere interrogato per rendere le informazioni disponibili attraverso la rete.

Le NIS map vengono memorizzate nel **Master server** che le rende disponibili ai client tramite il processo **ypserv**.

I client possono aggiornare le loro informazioni ricevendo i database tramite il demone **ypbind**. Sia client che server fanno parte del **NIS domain**.

Generalmente le informazioni NIS sono memorizzate nella directory **/var/yp** e vanno ridistribuite ogni volta che viene effettuata una **modifica** al **Master server**.

Il grande vantaggio di NIS quindi è che ci permette di avere un controllo centralizzato dei **File di Amministrazione** in un **singolo server** contattabile da ogni altro host in rete.

I **servizi** condivisi attraverso **NIS**: Autenticazione (login, pw), Home directory, NSF, Risorse di rete.

# NSF

Network File System.

Permette di **condividere** su una rete, **directory** e **file**. In questo modo utenti e programmi possono accedere a file memorizzati su sistemi remoti (server) come se fossero file locali.

**Vantaggi:** riduzione spazio occupato sul disco locale, semplificazione dei task di supporto (aggiornamento centralizzato dei file), manipolazione dei file remoti con comandi UNIX locali.

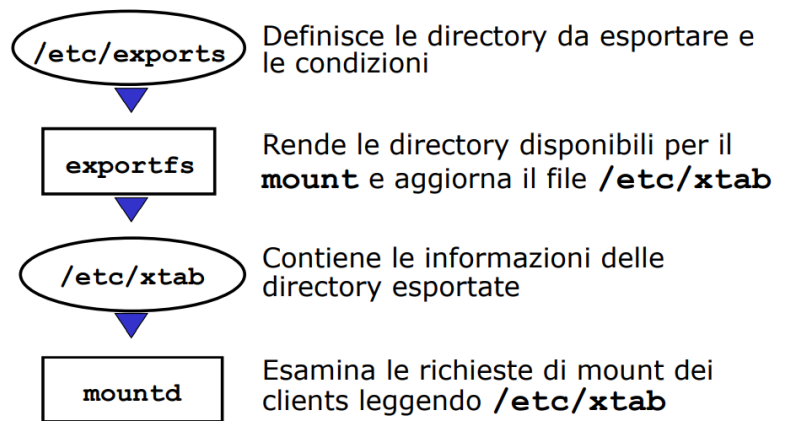
**Lato client** → l'inserimento di una directory di un host remoto nel filesystem locale è detto mounting. Comando: **# mount**.

**Lato server** → la condivisione di una directory locale ad host specifici per l'accesso remoto è detta sharing. Comando: **# export**.

Sul **server NSF** va configurato **/etc/exports**, un file ASCII che contiene le directory da esportare e l'elenco degli host che hanno accesso a questi file.

## Programmi NSF:

- nfsd** : demone che gestisce le richieste NFS
- biod** : demone che gestisce I/O lato client
- rpc.lockd** : gestisce i lock file (server-client)
- rpc.statd** : controlla lo stato della rete (server-client)
- rpc.mountd** : esegue le richieste di mount del client (server)



NSF è stato implementato in **3 parti indipendenti**. Oltre a **NSF**, infatti, sono state create due sottoinsiemi: Remote Procedure Call (**RPC**) e eXternal Data Representation (**XDR**).

Questo perché così, gli ultimi due possono essere usati anche da altri protocolli e programmi applicativi.

I programmi possono accedere ai file remoti con le stesse procedure che utilizzano per quelli locali, proprio grazie a RPC e XDR.

**RPC** consente di eseguire **procedure remote**.

**XDR** consente lo scambio di dati tra macchine con architetture eterogenee, senza preoccuparsi della conversione tra le diverse rappresentazioni dei dati a livello HW. La **rappresentazione dei dati** viene resa **indipendente** dalla macchina.

Ai lati del canale di comunicazione, infatti, i programmi usano le procedure XDR per convertire i dati dalla rappresentazione HW locale ad una indipendente dall'architettura del computer. Poi una volta che i dati vengono ricevuti, verranno riconvertiti dalla forma XDR indipendente, a quella locale.



# SNMP

Simple Network Management Protocol. **UDP** porta **161**.

Gira a livello applicazione e comunica con i dispositivi utilizzando i servizi di trasporto del TCP/IP in modo da poter controllare qualsiasi dispositivo connesso in Internet, invece che limitare il controllo ai dispositivi di rete locale.

È un protocollo di **gestione di reti, sistemi e applicazioni**.

Usando SNMP gli Amministratori di Sistema possono **monitorare lo stato delle risorse** e delle app, in modo anche da prevenire i guasti.

L'**Architettura SNMP** è composta da:

- **Agent** (nodi gestiti): dispositivi in grado di collezionare dati SNMP e di rispondere alle richieste del Manager.
- **Manager** (stazione di gestione): programma che controlla lo stato degli Agent e gli invia comandi quando si verificano guasti, in modo da poter gestire intelligentemente gli eventi.
- **SNMP** (protocollo di gestione): definisce le modalità di interazione tra Manager e Agent.
- **MIB** (Management Information Base– informazioni di gestione): archivio di informazioni di gestione immesse dagli agent. Queste info vengono chiamate **oggetti**.

Ogni oggetto mantiene una serie di variabili SNMP che descrivono il suo stato.

La **collezione** di tutti i possibili **oggetti** in una rete è il MIB.

Il Manager comunica con gli Agent mediante SNMP che gli consente di conoscere lo stato delle variabili MIB e di modificarle se opportuno.

- **SMI** (Struttura dell'Informazione di gestione): definisce come le informazioni sono collegate tra di loro. È l'insieme delle regole che definiscono il **nome** delle variabili MIB. Include definizioni di base come **Indirizzo** e **Contatore** e specifica la **sintassi dei nomi** (leggibile dagli utenti) e la **codifica binaria** (per i messaggi).

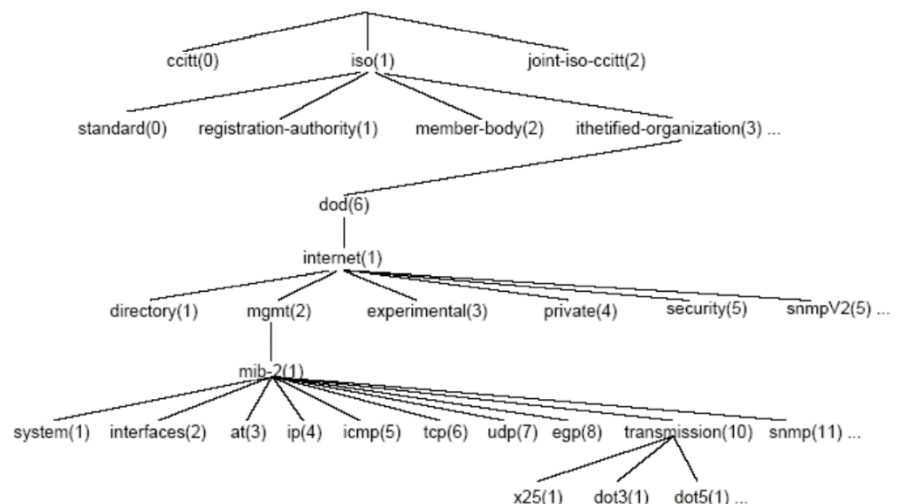
Viene specificato con un sottoinsieme delle regole dell'Abstract Syntax Notation 1

(**ASN.1**) uno **standard ISO**, che permette la definizione degli oggetti SNMP e del modo di codificare e trasferire queste informazioni in rete.

Tra i **tipi** permessi in SNMP dell'**ASN.1**

troviamo: integer, bit string, octet string, null, e **Object Identifier**.

Quest'ultimo in particolare contiene i criteri per definire un oggetto che va messo in una posizione precisa di un **albero di standard**. Tutti gli oggetti sono identificati da un'**etichetta** che corrisponde al cammino sull'albero per arrivare a tale oggetto.



**Es.:** {iso (1) identified-organization (3) dod (6) internet (1) mgmt (2) mib-2 (1) ...}  
o alternativamente: { 1 3 6 1 2 1 ... }

ASN.1 definisce il modo univoco in cui i valori dei tipi ASN.1 sono convertiti senza ambiguità in una sequenza di byte detta **BER** (Basic Encoding Rules).

La **codifica è ricorsiva**, questo significa che corrisponde alla concatenazione delle codifiche degli oggetti che la compongono.

Gli **oggetti** che appartengono al **MIB** sono raggruppati in **12 categorie** che corrispondono a 12 nodi al di sotto del nodo mib-2 della struttura ad albero. Esse servono per definire le categorie di base in modo che il Manager possa gestire le varie risorse.

**Forma sintattica MIB:** 1.3.6.1.2.1

MIB category	Includes Information About
system	The host or router operating system
interfaces	Individual network interfaces
at	Address translation (e.g., ARP mappings)
ip	Internet Protocol software
icmp	Internet Control Message Protocol software
tcp	Transmission Control Protocol software
udp	User Datagram Protocol software
ospf	Open Shortest Path First software
bgp	Border Gateway Protocol software
rmon	Remote network monitoring
rip-2	Routing Information Protocol software
dns	Domain Name System software

SNMP ha avuto diverse versioni dove la **sicurezza** rappresentava un problema (credenziali in chiaro). Ciò è stato risolto in **SNMP v.3** che introduce:

- **L'Autenticazione:** (tra Manager e Agent) per evitare modifiche alle informazioni, il mascheramento e la modifica della sequenza di messaggi.
- **Privacy:** riservatezza delle info (crittografia).
- **Configurazione remota** di sistemi gestibili mediante un insieme di operazioni sicure.

## DHCP (RFC 2131)

Dynamic Host Configuration Protocol. **UDP** porta **67** (server) **68**(client).

Estende il protocollo **BOOTP** con: l'**allocazione automatica** di indirizzi di rete riusabili, e opzioni di configurazione aggiuntive.

Fornisce supporto per lo **scambio** tra host di **informazioni di configurazione** degli stessi host, in una rete TCP/IP.

È costituito da **2 componenti**:

1. Un **protocollo** per la **trasmissione** dei **parametri** di configurazione specifici di un host da un DHCP server all'host interessato.
2. Un meccanismo per **assegnare** gli **indirizzi** di rete agli host.

Gli host designati dall'Amministratore ad essere **server DHCP** assegnano indirizzi di rete e comunicano i vari parametri di configurazione ai client.

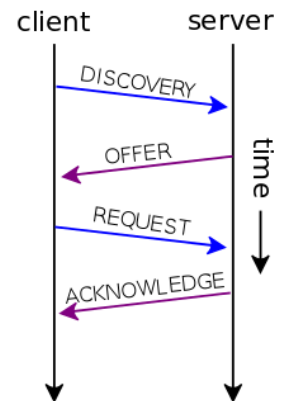
**NON** è utilizzato per configurare i **router**, che invece hanno una configurazione statica.

**BOOTP** viene eseguito quando la macchina viene accesa. È un **meccanismo di trasporto** per la raccolta delle info di configurazione di host che sfrutta gli **indirizzi Broadcast** per permettere l'assegnamento degli indirizzi IP. L'uso del broadcast, tuttavia, limita la sua operatività solo nella rete locale, perché i router non istradano tali pacchetti.

Questo limite viene superato con il **DHCP** che permette di configurare un router come **Relay Agent** che riconosce il **traffico BOOTP**.

DHCP gira come applicazione e usa i **servizi Broadcast limitati di IP** (dato che il client non conosce ancora il proprio indirizzo IP e quindi non può usare ARP) per inviare dei Broadcast IP in rete locale anche quando IP non è completamente inizializzato.

In pratica un client invia dei messaggi broadcast a tutti i server della rete locale chiedendogli un indirizzo IP (**1.Discovering**). I server gli mandano delle **offerte** (**2.Offering**) che il client raccoglierà, e tra queste ne selezionerà una. Invierà quindi in broadcast un messaggio che annuncia quale offerta ha accettato (**3.Requesting**). Di questa offerta andrà poi verificata l'accettazione da parte del server (**4.Acknowledgment**).



3 Tipi di **Allocazione**:

- **Automatica**→DHCP assegna un indirizzo IP ad un client permanentemente.
- **Dinamica**→DHCP assegna un indirizzo IP ad un client per un periodo limitato di tempo o fin quando il client non rilascia l'indirizzo. (riusabilità degli indirizzi)  
In questo caso il rinnovo degli indirizzi avviene o al riavvio del client, o a metà durata dell'assegnamento, oppure in prossimità dello scadere dell'assegnamento.
- **Manuale**→l'indirizzo IP viene assegnato al client dall'Amministratore e il DHCP è usato solo per comunicare l'indirizzo scelto al client.

Per configurare un server DHCP bisogna:

1. **installare il software**: dipende dal SO del server.
2. configurare uno o più **pool di indirizzi IP** da assegnare ai client: specificare quindi un indirizzo iniziale e uno finale, eventuali intervalli da escludere, e la validità degli indirizzi assegnati.
3. **Opzioni di configurazione** dei client: vengono applicate a tutti i client che hanno un indirizzo IP compreso nel pool di indirizzi del punto 2. E possibile impostare delle opzioni globali in server con più pool, opzioni di base, ed è anche possibile riservare un particolare indirizzo IP ad un client associandolo al suo MAC.

## NAT (RFC 1631)

Network Address Translation. **TCP o UDP** a seconda del pacchetto inviato dall'indirizzo privato.

Servizio utilizzato per la **conservazione** degli indirizzi IP.

Permette a **reti private** di connettersi ad Internet. In questo modo un grande numero di host può condividere un ristretto numero di indirizzi IP.

NAT viene eseguito da un **router** che connette normalmente due reti, **traducendo** gli **indirizzi** della **rete interna** in un **unico indirizzo pubblico** (quello del router), e poi ritornando al client la rispettiva risposta. La mappatura tra **IP-Privati+Porta** e **IP-Pubblici+Porta** viene scritta nella **NAT Table**, così che i pacchetti possano essere istradati in modo non ambiguo.

Il fatto che gli indirizzi privati siano tradotti in un altro indirizzo fornisce un elemento di **sicurezza** molto robusto, perché nasconde la rete all'esterno.

# POSTA ELETTRONICA

È il servizio più antico e importante di Internet che ha stravolto la produttività di chi la usa.

Un servizio basato sulla Posta Elettronica sono le **Mailing-List**: un portale dove gli utenti si possono registrare e dove i messaggi verranno distribuiti a tutti i membri.

Generalmente sono moderate da dei **moderatori** che controllano il contenuto del messaggio prima che questo venga inviato ai membri.

Le Mailing-List hanno permesso la nascita delle prime **Community** in Internet.

La Posta Elettronica si basa su due grossi pilastri:

1. **Mail User Agent (MUA)** → è l'interfaccia dell'utente verso l'applicazione, quindi funge da ponte tra il server di posta e le attività dell'utente (accesso, archiviazione messaggi, scrittura messaggi, ecc.).

Svolge le funzioni di:

- **Composizione** del messaggio rispettando la sintassi opportuna, mediante un editor di testo. Gestione degli strumenti di **Agenda Elettronica** e la gestione, quanto più automatica possibile dei **campi dell'intestazione** (CC, CCN, Mittente, Destinatario...)
- **Visualizzazione** del messaggio separando i suoi header dal corpo e gestendo eventuali allegati. Fornire strumenti di **archiviazione** dei messaggi, e interagire con il sistema per **convertire i file** oppure per aprire automaticamente applicazioni che possono leggerli.
- **Eliminazione** dei messaggi dal server per evitare che la mailbox non raggiunga dimensioni troppo grandi che potrebbero andare a danneggiare il server di posta.

2. **Mail Transfer Agent (MTA o Sendmail)** → è il programma di trasporto delle email.

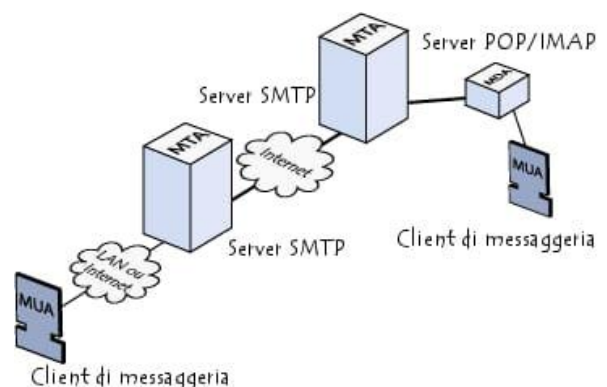
Svolge le funzioni di:

- **Trasferimento**: il MUA gli passa il messaggio inviato dall'utente, e l'MTA andrà ad interpretare l'indirizzo di posta elettronica e a trasportare il messaggio. Quindi Sendmail riceve e spedisce posta secondo il protocollo **SMTP** (Simple Mail Transfer Protocol).  
Ciò comporta l'attivazione di una **sessione** con il server di destinazione o qualche macchina intermedia.
- **Notifica** all'utente qualora il messaggio non sia inviato. Alcuni prodotti gestiscono la ricevuta di ritorno, che però non è stata standardizzata.
- Fornisce **Alias** di posta che permettono anche la creazione di Mailing-List prototipali.

Il programma popper (**POP3**) e il protocollo **IMAP** consentono di interagire con l'**MTA** direttamente da un **PC** connesso in rete, in modo da consentire la **manipolazione** della posta direttamente dal PC client. IMAP è più sicuro di POP3 perché ha un'autenticazione più forte.

Gli **indirizzi** di posta elettronica sono del tipo:

[user@host.domain](#) oppure [user@domain](#).



Nel primo caso andrà definito l'**host come MX** nella DNS Table, nel secondo caso andrà definito il record **MX per il dominio**.

Quando un dominio di posta è importante è bene definire un **gestore di posta secondario** in una rete esterna e lontana, così che in caso di malfunzionamenti questo server secondario possa **salvare temporaneamente i messaggi** in transito e fare le veci del server di destinazione.

Quando il server principale ritorna disponibile l'**email-relay** gli invierà tutti i messaggi che aveva salvato.

Questo meccanismo di email-relay è stato abusato dagli **Spammer** che si fingevano server secondari e inviavano posta indesiderata al server di posta, configurato per accettare tutti i messaggi provenienti dall'email-relay.

**RFC822** definisce una serie di **campi di intestazione obbligatori**, importantissimi per il corretto funzionamento della posta elettronica.

```
Received:      elenca i server che ha attraversato
Message-ID:    identificativo del messaggio
                (B00002222000@mailhost.dom.it)
Date:          data Wed, 23 Oct 2000 10:22:00 +0100
From:          utente mittente
Subject:       soggetto
To:            utente destinatario
Cc:            utenti destinatari in carbon copy
Bcc:           utenti destinatari in Blind Carbon Copy
Mime-Version:  intestazioni MIME
```

## SENDMAIL

La **configurazione** di Sandmail è **complessa**, ecco perché viene semplificata mediante l'uso di uno pseudolinguaggio chiamato **M4**, il quale consente di attivare delle funzioni mediante l'invocazione di **macro** (blocco di comandi ricorrente durante l'esecuzione di un programma).

File di configurazione:

- [/etc/mail/relay-domains](#) → elenca i domini per i quali si accetta il **relaying**.
- [/etc/mail/access](#) → elenca i domini o gli host per i quali si **controlla l'accesso** (chi può mandarmi la posta in entrata).
- [/etc/mail/sendmail.cw](#) → elenca i domini per i quali si fa **virtual hosting**.

# POSTA ELETTRONICA CERTIFICATA (PEC)

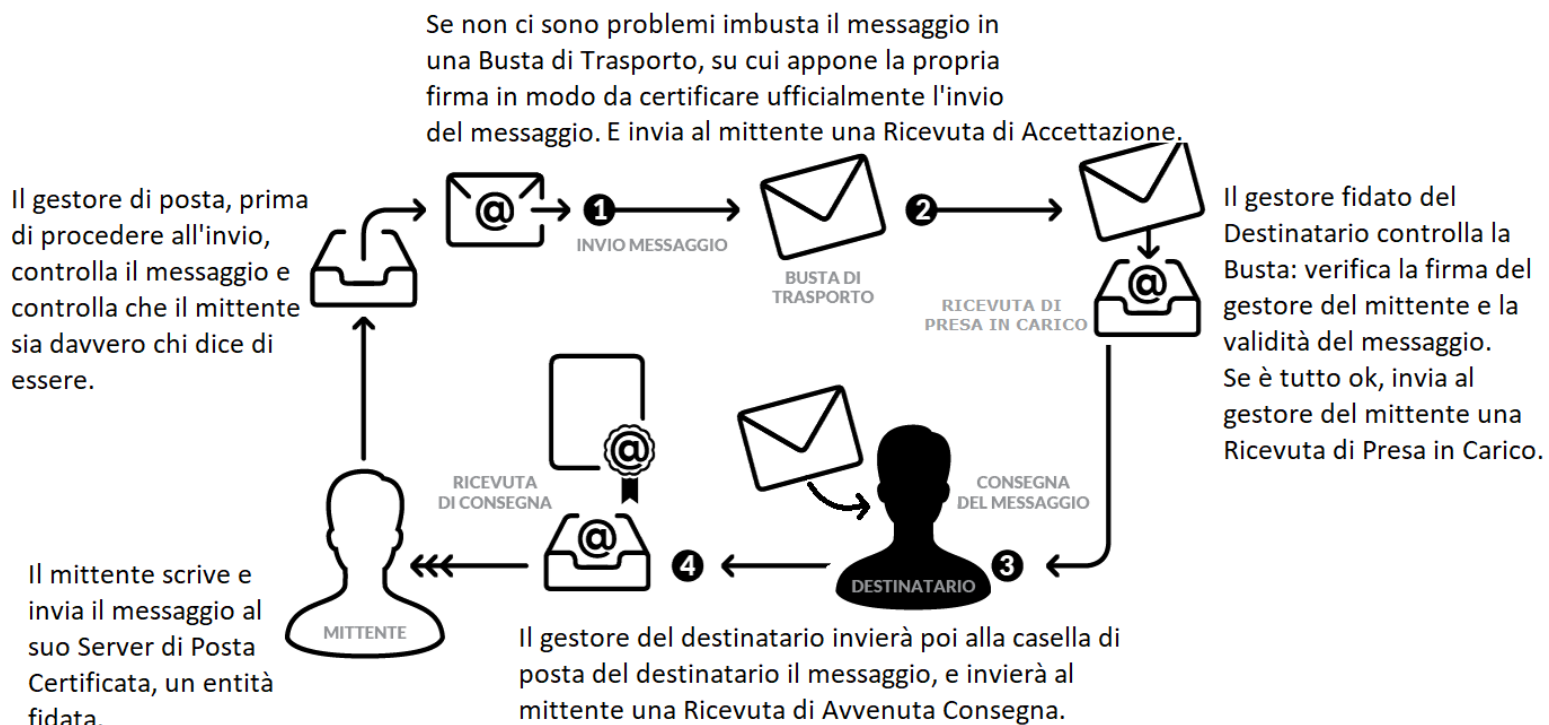
Una trasmissione può essere considerata posta certificata solo se le **caselle** del **mittente** e del **destinatario** sono entrambe caselle di **PEC**, altrimenti il sistema sarà in grado di fornire solo parte delle funzionalità previste.

I gestori di posta certificata sono obbligati a **registrare** tutti i principali **eventi** che riguardano la trasmissione per **30 mesi**, da fornire come **prova** da parte degli interessati. Devono inoltre utilizzare l'**orario esatto** garantitogli da istituti ufficiali.

I servizi di posta elettronica utilizzano esclusivamente **protocolli sicuri** (crittografati) per impedire la manomissione dei messaggi da parte di terzi. (https per webmail, smtps, smtp starttls, pop3s, imaps)

L'identificazione degli utenti avviene tramite **username** e **pw**, oppure tramite **certificati digitali**, così che non si possa verificare la **falsificazione d'identità**.

## Funzionamento PEC:





# MIME

Multipurpose Internet Mail Extensions. RFC 1341 e 1521.

È uno **Standard di Codifica** dei messaggi di Posta Elettronica introdotto per venire in contro alle nuove esigenze di inviare messaggi: in **lingue diverse dall'Inglese** (quindi set di caratteri diversi, lettere accentate, alfabeti non latini come l'ebraico o il russo, lingue prive di alfabeto come il cinese), e con **allegati** (informazioni non testuali).

MIME introduce nuove **headers** rispetto a quelli dell'RFC 822:

**Content-Type** è diventato molto popolare e usato da diverse Applicazioni Internet per identificare il tipo di contenuto di un certo file.

header	significato
<b>MIME-version</b> Content-Description <b>leggibile</b>	<b>identifica la versione MIME</b> <b>descrive il contenuto del messaggio in forma</b>
Content-Id Content-Transfer-Encoding Content-Type	<b>identificatore del messaggio</b> <b>tipo di codifica utilizzata per la trasmissione</b> <b>tipo di contenuto del messaggio.</b>

È costituito da **7 tipi e sottotipi**, separati gli uni dagli altri con uno /. Es: Content-Type: video/mpeg.

Un ruolo fondamentale nella posta elettronica ce l'ha la **Codifica** del messaggio.

I computer rappresentano le informazioni in base alle caratteristiche HW e SW, quindi la stessa sequenza di informazioni binarie ha significato diverso a seconda della configurazione dell'host.

La codifica più diffusa è quella **ASCII**, che rappresenta le informazioni in **7bit** ed è usata nei sistemi **UNIX** e **Windows**.

Un'altra codifica è **EBCDIC**, che utilizza **8bit**.

Entrambe le codifiche, tuttavia, non devono superare i **1000 caratteri** di lunghezza per ciascuna riga del messaggio.

La codifica introdotta da MIME è **Base64**, che rappresenta una qualsiasi sequenza di bit in base alla sequenza di lettere, numeri e simboli (A-Z, a-z, 0-9, +/=). Le linee di un messaggio possono essere lunghe massimo **76 caratteri**, cosa che offre la massima **compatibilità** con i programmi di trasporto della posta elettronica.

Base64 per i messaggi testuali è **inefficiente**, ecco perché si preferisce una codifica definita da MIME come **quoted-printable-encoding**, costituita dalla codifica **ASCII standard**, dove però i caratteri che hanno **codifica superiore a 127** vengono rappresentati da **due caratteri HEX**.

# SMTP (RFC 821)

Simple Mail Transfer Protocol.

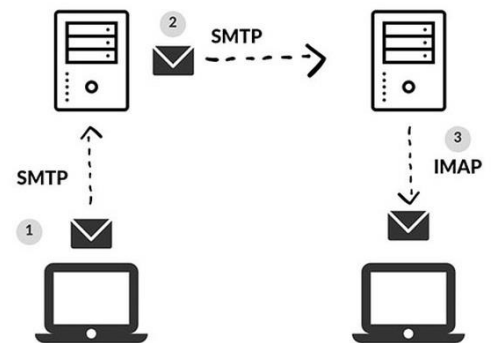
È il protocollo di **trasporto affidabile** ed **efficiente** dei messaggi di posta elettronica. SMTP implementa un **livello di astrazione** superiore, infatti consente di inviare messaggi indipendentemente dai servizi di trasporto specifici sottostanti.

Il **servizio di trasporto** che utilizza è l'Interprocess Communication Environment (**IPCE**), quindi l'invio del messaggio avviene indipendentemente dal tipo di rete alla quale gli host sono collegati, sarà uno scambio dati tra due IPCE (i peer della connessione).

Il risultato di una richiesta di posta dell'utente, si apre un **canale di comunicazione bidirezionale** sulla **porta 25**, tra il server SMTP mittente ed il SMTP destinatario che può essere il destinatario finale o un server intermedio.

I comandi SMTP sono generati dal server SMTP mittente ed inviati a quello destinatario, mentre le risposte sono generate dal server SMTP destinatario ed inviate al mittente. (duh...)

Quando noi inviamo un messaggio, il **MUA** contatta il server **SMTP** che invierà il messaggio al server SMTP del destinatario. Se non lo conosce lo chiede al **DNS**.



Quindi, una volta aperto il canale trasmissivo, l'SMTP-mittente invia il comando **MAIL** indicando colui che invia l'email, se il ricevente può ricevere messaggi risponde con **OK**.

Il mittente poi invia il comando **RCPT** identificando il recipient del messaggio, se il ricevente può ricevere messaggi per quel destinatario risponde con **OK**, altrimenti con un codice di **reject**.

Il server SMTP invia i dati del messaggio col comando **DATA**, terminando con la sequenza speciale **<CRLF>.<CRLF>**. Se il ricevente interpreta correttamente i dati risponde con **OK**.

Tutto questo dialogo è a **passi bloccanti**.

I comandi di mail hanno una sintassi rigida: il primo comando deve essere **HELO**, i comandi **NOOP**, **HELP**, **EXPN**, **VRFY** possono essere utilizzati ovunque nella sessione, mentre **MAIL**, **SEND**, **SOML**, **SAML** iniziano una transazione di mail per l'invio del messaggio e devono essere seguiti dai comandi **RCPT** e **DATA**. Dopo il testo del messaggio deve seguire la sequenza **<CRLF>.<CRLF>** (ritorno.ritorno a capo). Una transazione può essere abortita con **RSET**.

L'ultimo comando è **QUIT** che non può essere usato in nessun altro momento della sessione.



# POP3 (RFC 1225)

Post Office Protocol.

È stato il primo protocollo per **MUA**, che quindi permette la comunicazione tra il **PC** del **client** e il **server di posta**.

Si tratta di un'applicazione **client-server** che consente al client di **ricevere** nuovi messaggi, **cancellare** i messaggi presenti sul server SMTP, mentre **l'invio di posta** tra server di posta mittente e destinatario avviene in **SMTP**.

Quindi il client interagisce con il server POP3 in ascolto sulla **porta 110**, il quale poi dialoga con il server SMTP (chiamato Message Transport System, **MTS**).

Quando il client chiede il servizio apre una **connessione TCP/IP**, poi si autentica inviando login e pw in chiaro (fase di **Authorization**) e una volta aperta il server invia un messaggio di benvenuto. Client e server si scambiano poi comandi e risposte (fase di **Transaction**), fino alla chiusura della sessione (fase di **Update**). I comandi sono composti da una **keyword** più un eventuale **argomento** e terminati con un **doppio CRLF**. Le risposte possono essere o **+OK** oppure **-ERR**.

## ■ Comandi **TRANSACTION**:

- STAT** fornisce numero di msg e dimensione in byte
- LIST [msg]** elenca msgid e dimensione dei(l) msg
- RETR msg** riceve msg
- DELE msg** marca msg come cancellato
- NOOP** non fa nulla
- LAST** indica il più alto msgid acceduto
- RSET** se ci sono msg marcati per essere cancellati, viene tolta detta marcatura

## ■ Comandi **UPDATE**:

- QUIT** chiude la connessione

Con POP3 i messaggi devono essere scaricati sul PC, anche se è possibile lasciarne una copia sull'host.

# IMAP (RFC 1064 e 2060)

È un metodo per l'**accesso dinamico** alle **mailbox** e alle news di server centrali, da parte di **PC** e **stazioni mobili**. **Porta 143** del TCP/IP.

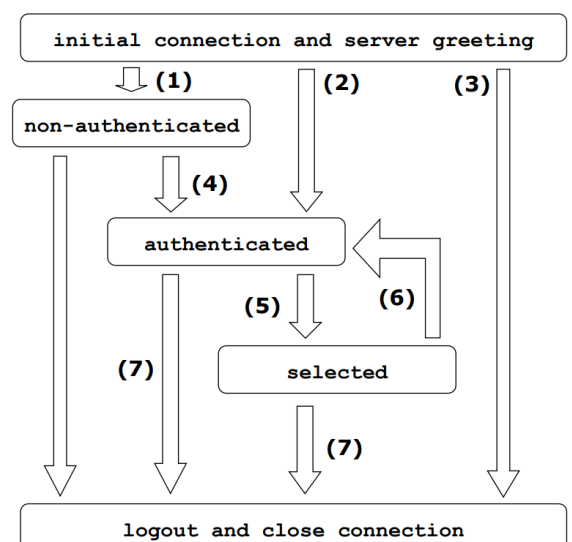
**IMAP v.4** è molto più **efficiente** di POP3 al crescere dei client.

Inoltre fornisce all'utente una serie di informazioni (**System flag**) che sono conseguenza delle azioni fatte dall'utente stesso.

(Es.: leggo il messaggio → attributo=seen. Rispondo al messaggio → attributo=answered)

È dotato di **potenti strumenti di ricerca** delle informazioni di gestione e delle informazioni multimediali (**campi MIME**).

IMAP a differenza di POP3, mantiene tutti i messaggi sul **server** e consente all'utente di **organizzare i messaggi** in cartelle.



# POSTA ELETTRONICA PRIVATA

La **segretezza** dei messaggi di posta è di fondamentale importanza, in modo da avere **confidenzialità, integrità e autenticazione**.

Ciò che ci garantisce queste caratteristiche è la **crittografia** e i **certificati digitali**.

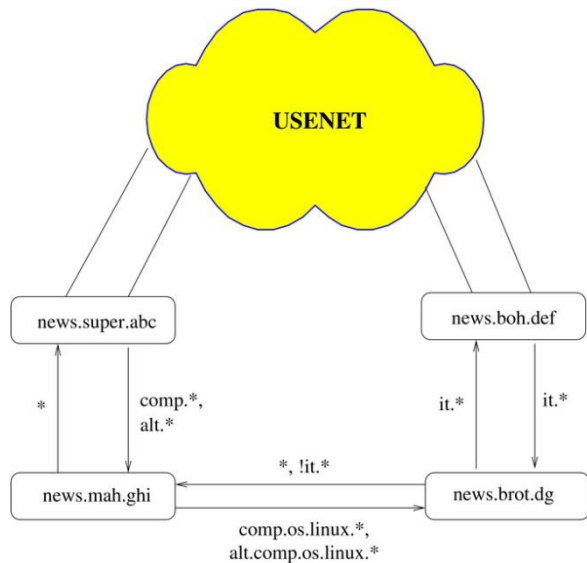
Storicamente il primo prodotto che ci ha consentito di cifrare la posta elettronica è stato Pretty Good Privacy (**PGP**) che **crittografa** e **firma digitalmente** messaggi e dati e permette di accedere ai repository di chiavi pubbliche (**PKI**, Public Key Infrastructure).

Un altro prodotto che ci può aiutare è **GnuPG**, un applicativo che si usa da **linea di comando** ed è interfacciato a moltissime applicazioni. La versione 2 di GnuPG fornisce anche il supporto per **S/MIME** e **SSH**.

## USENET NEWS

È stato un servizio fondamentale per lo sviluppo di Internet.

Le news assomigliano molto alla posta elettronica perché hanno una **strutturazione** dei gruppi di discussione **a domini**, solo che si va **da sx** (gruppo contenitore) **verso dx** (gruppo più specifico). La differenza con la posta, inoltre, sta nel fatto che **le news vanno lette dall'utente**, non gli arrivano in casella. È lui che deve collegarsi al server e scegliere a quali gruppi abbonarsi e che news scaricare e leggere.



Un **gruppo**, quindi, è un **luogo di discussione** su un dato argomento a cui gli utenti interessati a tale argomento possono **isciversi**, in modo che il client riceva dal server la lista dei messaggi del gruppo. Il corpo della news viene scaricato sul client solo quando l'utente lo accede.

Con le news si è sviluppata tutta una serie di **modi di dire** come: BTW, ROFL, FAQ, ecc..., così come sono nate le **emoticon** :-).

Quando un utente invia un messaggio si dice che effettua un **post**, quando questo messaggio è inviato a diversi gruppi si parla di **crosspost**.

I gruppi possono essere **moderati** o meno, con il controllo del messaggio da parte del moderatore prima che questo venga inviato. In questo modo si possono avere gruppi di alta qualità, ma scarsa spontaneità.

Inoltre, la **Netiquette** impone di non inviare materiale al gruppo che non sia coerente con le finalità del gruppo stesso.

Per aprire un newsgroup serio occorre effettuare una **raccolta di consensi** (CFV, call for votes), attivata da un messaggio in **news.group** che definisce le finalità del gruppo e se deve o no essere moderato.

Non seguono questa procedura i gruppi della gerarchia **alt**, infatti in questa gerarchia si può trovare di tutto, anche materiale al limite della legalità.

I messaggi Usenet News vengono scambiati tra i server utilizzando il Network News Transfer Protocol (**NNTP**) sulla **porta 119**.

Il server che riceve il messaggio invia il messaggio di possedere tale news a tutti gli altri server, trasmettendolo ai suoi **server prossimi** che poi lo invieranno ai loro server prossimi e così via (meccanismo **newsfeed**). Se i server che ricevono il messaggio non hanno già quella determinata news la accetteranno, altrimenti no (metodo **I HAVE**).

I messaggi rimangono **archiviati** nel server fino alla data di scadenza, quando poi vengono **cancellati** dal server stesso.

## World Wide Web

Il WWW è un'**architettura software** che consente di accedere a documenti tra loro collegati e distribuiti su migliaia di macchine nell'Internet Globale. Nasce al **CERN** nel **1989**.

Il grande successo del WWW sta nella sua **interfaccia grafica** facile e piacevole da utilizzare per accedere ai servizi Internet.

Il **primo browser** WWW, **Mosaic**, viene sviluppato presso il **NCSA** nel **1993**. Ha talmente tanto successo che viene poi fondata la **Netscape Corp.** per lo sviluppo di servizi per il Web.

Nel 1994 nasce **W3C** con lo scopo di coordinare lo sviluppo delle tecnologie Web, ed è oggi la struttura di riferimento per lo sviluppo del Web.

## HTTP

Hypertext Transfer Protocol.

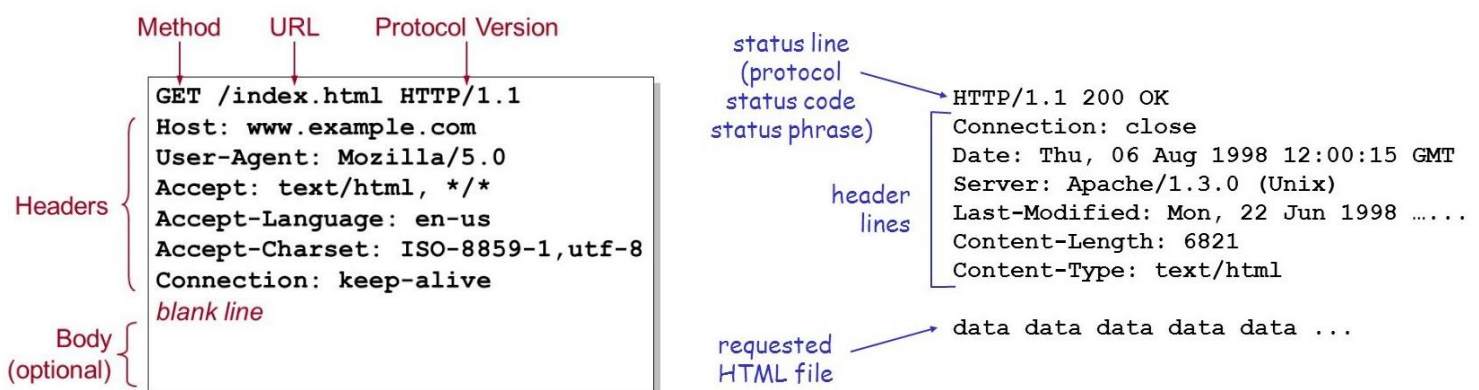
È un protocollo a **livello applicativo** per realizzare **sistemi informativi distribuiti ipermediali** e collaborativi. È alla base del Web, infatti è usato dal **WWW** dal 1990.

In sostanza HTTP fornisce un insieme aperto di **metodi** per specificare lo **scopo di una richiesta** e fornisce attraverso gli **URL/URI/URN** un modo per **identificare** la risorsa alla quale il metodo deve essere applicato.

**HTTP/1.0** consente il trasferimento di **dati grezzi** e dei messaggi di tipo **MIME**, tuttavia non tiene in considerazione i proxy gerarchici, il caching, le connessioni persistenti e dei virtual host.

**HTTP/1.1** è stata il consolidamento del protocollo, garantendo una **implementazione stabile** delle sue caratteristiche. Questa versione è stata usata per molti anni, ma ultimamente è diventata **inefficiente** e non in grado di fornire un servizio adeguato (**lentezza**).

Ecco perché è nato **HTTP/2** (RFC 7540). HTTP **request** e **response**:



## HTTP/1.1

È di tipo richiesta/risposta **senza stato**.

Utilizza **3 parole** per definire il significato di ogni requisito: **MUST** (l'oggetto è requisito assoluto), **SHOULD** (raccomandato), **MAY** (opzionale).

## HTTP/2

Le differenze con HTTP/1.1 sono che HTTP/2 è:

- **Binario**, non testuale
- Basato su un'unica **connessione, multiplexed** e basata sulla concorrenza. I messaggi sono divisi in **frames** con stesso header.
- Il client può indicare al server quale risorsa è più importante delle altre.
- Usa la **compressione dell'header** per ridurre l'overhead.
- Consente ai server di caricare proattivamente le risposte nella cache dei client.

Le applicazioni non devono cambiare nulla per passare a HTTP/2, si noterà solo un miglioramento nelle prestazioni. C'è infatti un **consumo ridotto di risorse** sia del client che del server.

HTTP/2 offre miglioramenti nelle **transizioni crittografate**, e un'**ottimizzazione** del livello **TCP** (l'unica connessione viene usata per più soggetti, ognuno dei quali avrà una propria finestra nella connessione).

## HTTP/3

Basato sul protocollo di trasporto **QUIC** (Quick UDP Internet Connections), riduce ulteriormente la **latenza** delle connessioni HTTP e ne migliora lo **streaming** all'interno della connessione stessa. QUIC è un protocollo sperimentale sviluppato da **Google**, il quale vuole che rimpiazzì TCP e UDP per il trasporto HTTP.

Diventa obbligatorio l'uso di **TLS 1.3** per compressione e crittografia.

