**RETI DI CALCOLATORI - LABORATORIO**

Sommario

[LEZIONE 1 1](#_Toc160487296)

[Identificazione Endpoint 1](#_Toc160487297)

[Indirizzi IP 1](#_Toc160487298)

[TCP E UDP 4](#_Toc160487299)

[Modello Client-Server 7](#_Toc160487300)

[LEZIONE 2A 8](#_Toc160487301)

[Configurazione delle interfacce di rete: \*nix (IFCONFIG - IP) 8](#_Toc160487302)

[Comando IWCONFIG 10](#_Toc160487303)

[Internet Control Message Protocol (ICMP) 11](#_Toc160487304)

[Interrogazioni ICMP 13](#_Toc160487305)

# LEZIONE 1

## Identificazione Endpoint

Ogni qual volta due endpoint vogliono comunicare devono identificarsi univocamente

Tale identificazione avviene attraverso due livelli di indirizzamento:

* Il primo determina l'host su cui e' in esecuzione il processo
* Ad ogni host di una rete IP e' associato un indirizzo IP
* Il secondo determina il processo con cui si vuole comunicare
* Ad ogni applicazione in esecuzione su un host è associato un numero di porta.

## Indirizzi IP

Un Indirizzo IP è un numero che identifica univocamente un dispositivo collegato ad una rete informatica che utilizza lo standard IP (Internet Protocol).

Gli indirizzi IPv4 sono costituiti da 32 bit (4 byte), e vengono descritti con 4 numeri decimali rappresentati su 1 byte (quindi ogni numero varia tra 0 e 255) separati da un punto. Es: 192.168.1.134

Questa rappresentazione limita lo spazio di indirizzamento a 4,294,967,296 indirizzi univoci possibili ( 2^32). Inoltre la rete internet esclude 18.000.000 indirizzi utilizzati per le reti private.

Per ovviare al problema della mancanza di indirizzi IP dovuta alla costante crescita di Internet è stato introdotto l'IPv6

Nell’IPv6 gli indirizzi sono lunghi 128 bit (quindi 2^128 possibili indirizzi).

L’indirizzo viene suddiviso in 8 blocchi di 16 bit ciascuno. I blocchi sono separati da “:” e vengono rappresentati in notazione esadecimale.

Es: **3ffe:1001:0001:0100:0a00:20ff:fe83:5531**

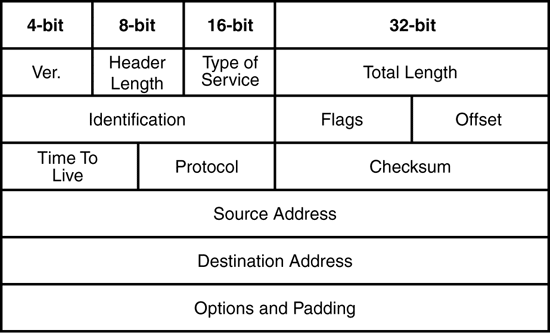
Es: **3ffe:1001:0001:0000:0000:0000:0000:0001**

Esistono delle semplificazioni:

* **si possono omettere gli zeri iniziali** 
  + **3ffe:1001:1:100:a00:20ff:fe83:5531**
* **si possono sostituire gruppi di zeri con** “**::**”
  + **3ffe:1001:1::1**

Gli indirizzi IPv6 compatibili IPv4 si scrivono: “**::163.162.170.171”**

IP HEADER:



In un ambiente multitasking più processi in esecuzione su uno stesso host devono poter comunicare mediante lo stesso sottosistema di rete. E' necessario consentire più connessioni simultaneamente.

Per poter tenere distinte le diverse connessioni su uno stesso host si utilizzano le **porte**

Numeri di Porta  
Le porte sono interi a 16 bit da 0 a 65535

* Da 0 a 1023: porte riservate (ai processi di root)
* Da 1024 a 49151: porte registrate
* Da 49152 a 65535: porte effimere (per i client, ai quali non interessa scegliere una porta specifica)

Esempi di porte riservate

* 21 ftp (trasferimento file)
* 22 ssh (login remoto sicuro)
* 25 smtp (invio email)
* 80 http (web)
* 143 imap (lettura email)

## TCP E UDP

I due principali protocolli relativi al livello di trasporto sono:

* **TCP** (Transport Control Protocol)
* **UDP** (User Datagram Protocol);

**TCP** o Transmission Control Protocol è un protocollo che ha le seguenti caratteristiche:

* **Servizio connesso**: Il protocollo TCP crea un canale logico (connessione logica) tra due host.

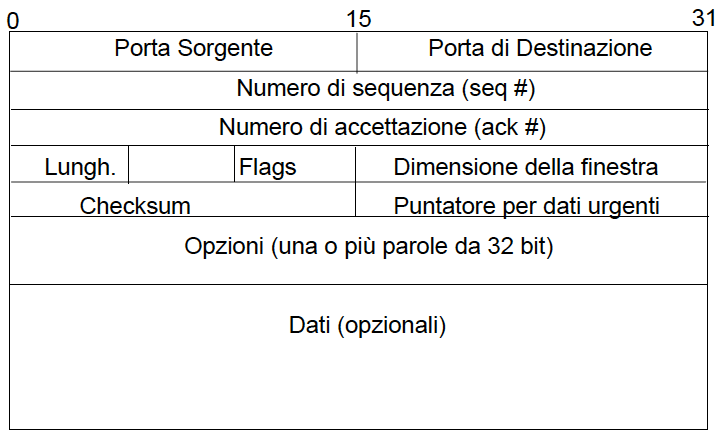
Ciò implica che il TCP lavora in unicast (1:1, un host deve potersi collegare con un altro host a livello logico). La connessione viene instaurata in prima istanza attraverso una tecnica chiamata Three-Way-Handshake e rilasciano la connessione attraverso una tecnica chiamata Four-Way-End-Connection.

* **Controllo dell’errore:** Ogni singolo segmento veicolato dal TCP è sottoposto al controllo dell’errore (impostato nell’header TCP).
* **Affidabile:** Assegna un numero di sequenza ad ogni byte trasmesso, attendendo una conferma di avvenuta ricezione (ACK). ll TCP ricevente, quando restituisce un ACK al mittente, invia anche il **numero massimo di sequenza** che può ricevere nella prossima trasmissione.
* **Ordine dei segmenti:** Il protocollo TCP è in grado di riordinare in automatico i segmenti TCP.

Di conseguenza i segmenti arrivano sempre ordinati.

* **Controllo del flusso:** L’host mittente e destinatario si mettono d’accordo per far si che un host non inondi di segmenti l’altro host che non è in grado di gestire.

Un segmento consiste in un preambolo (header) di 20 byte, più un campo opzionale di 4 byte, seguito da zero o più byte di dati;



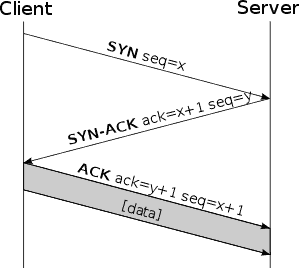
Il protocollo TCP viene utilizzato dai protocolli dei livelli superiori che necessitano di usufruire delle caratteristiche offerte dal TCP. Alcuni protocolli sono: Telnet, HTTP, SMTP, FTP-data.

**Three Way Handshake**

Un handshake a tre vie (Three-way handshake) è un metodo utilizzato in una rete TCP/IP per creare una connessione tra due host. Si tratta di un metodo in tre fasi che richiede ad entrambi gli host di scambiare pacchetti SYN e ACK (riconoscimento-acknowledgment) prima dell’inizio della comunicazione dei dati.

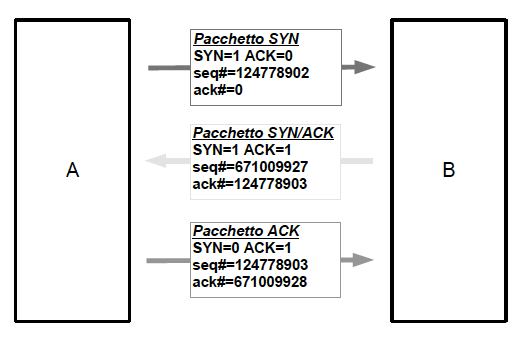
**Funzionamento**

* Un client invia un pacchetto dati SYN su una rete IP a un server sulla stessa rete o a una rete esterna. L’obiettivo di questo pacchetto è di chiedere se il server è disponibile per nuove connessioni.
* Il server di destinazione deve disporre di porte aperte in grado di accettare e avviare nuove connessioni. Quando il server riceve il pacchetto SYN dal nodo client, risponde e restituisce una ricevuta di conferma, il pacchetto ACK o SYN/ACK.
* Ilclient riceve il SYN/ACK dal server e risponde con un pacchetto ACK.
* Al termine di questo processo, viene creata la connessione e l’host e il server sono in grado di comunicare.



1. **A invia un segmento SYN a B** – il *flag SYN* è impostato a 1 e il campo *Sequence number* contiene il valore *x* che specifica l’*Initial Sequence Number* di A;
2. **B invia un segmento SYN/ACK ad A** – i *flag SYN* e *ACK* sono impostati a 1, il campo *Sequence number* contiene il valore *y* che specifica l’*Initial Sequence Number* di B e il campo *Acknowledgment number*contiene il valore *x+1* confermando la ricezione del ISN di A;
3. **A invia un segmento ACK a B** – il *flag ACK* è impostato a 1 e il campo *Acknowledgment number* contiene il valore *y+1* confermando la ricezione del ISN di B.

Il terzo segmento non sarebbe, idealmente, necessario per l’apertura della connessione in quanto già dopo la ricezione da parte di A del secondo segmento, entrambi gli host hanno espresso la loro disponibilità all’apertura della connessione. Tuttavia, esso risulta necessario al fine di permettere anche all’host B una stima del timeout iniziale, come tempo intercorso tra l’invio di un segmento e la ricezione del corrispondente ACK.



**UDP** o **User Datagram Protocol** è un protocollo senza connessione che si trova nel livello di trasporto del modello TCP/IP. Non stabilisce una connessione né controlla se l’host ricevente è pronto per ricevere o meno; invia semplicemente i dati direttamente. UDP viene utilizzato per trasferire i dati a una velocità maggiore. È meno affidabile e quindi utilizzato per la trasmissione di dati come file audio e video, VoIP

UDP non garantisce la consegna dei dati, né ritrasmette i pacchetti persi. È solo un protocollo wrapper che facilita l’applicazione nell’accesso all’IP.

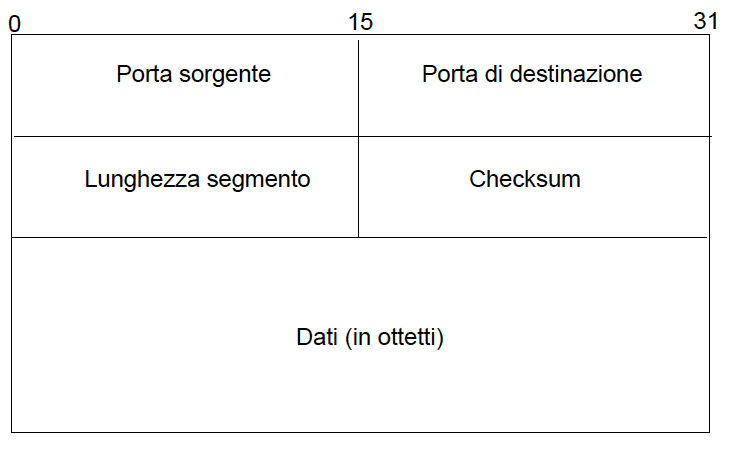
L'UDP fornisce soltanto i servizi basilari del [livello di trasporto](https://it.wikipedia.org/wiki/Livello_di_trasporto), ovvero:

* [multiplazione](https://it.wikipedia.org/wiki/Multiplazione) delle [connessioni](https://it.wikipedia.org/wiki/Connessione_(informatica)), ottenuta attraverso il meccanismo di assegnazione delle [porte](https://it.wikipedia.org/wiki/Porta_(reti));
* verifica degli errori ([integrità dei dati](https://it.wikipedia.org/wiki/Integrit%C3%A0_dei_dati)) mediante una [checksum](https://it.wikipedia.org/wiki/Checksum" \o "Checksum), inserita in un campo dell'intestazione ([header](https://it.wikipedia.org/wiki/Header" \o "Header)) del pacchetto, mentre TCP garantisce anche il trasferimento affidabile dei dati, il [controllo di flusso](https://it.wikipedia.org/wiki/Controllo_di_flusso) e il [controllo della congestione](https://it.wikipedia.org/wiki/Controllo_della_congestione).

Un datagram UDP consiste in un header di 8 byte (64 bit), seguito dai dati.

Il preambolo contiene quattro campi ciascuno della lunghezza di 2 byte (16 bit):

1. Porta sorgente (sprt)  intero a 16 bit (da 0 a 65535)
2. Porta destinazione (dprt)  intero a 16 bit (da 0 a 65535).
3. Lunghezza UDP  espressa in byte e comprende sia il preambolo che i dati
4. Checksum UDP  stringa di 16 bit utilizzata per il controllo degli errori.



Ogni datagramma UDP è inviato in un singolo datagramma IP. Quest'ultimo può venire frammentato durante una trasmissione e riassemblato dall'IP ricevente prima di essere presentato al livello UDP.

## Modello Client-Server

Nel modello client-server si distinguono due entità

* I programmi che forniscono un servizio, chiamati server
* I programmi di utilizzo, detti client che effettuano le richieste

Un server può (di norma deve) essere in grado di rispondere a più di un client.

Inoltre, distinguiamo due classi di server:

* **Concorrenti:** Un server concorrente gestisce più richieste dai client utilizzando il concetto di concorrenza, che è la capacità di eseguire più attività in parallelo.
* **Iterativi**: processa le richieste di servizio una alla volta. Possibile basso utilizzo delle risorse, in quanto non c’è sovrapposizione tra elaborazione ed I/O.

La classica implementazione di un server concorrente prevede ad ogni nuova richiesta client venga generato un **processo** che la gestisce.

TCP e UDP usano 4 informazioni per identificare una comunicazione

* Indirizzo IP del server.
* Numero di porta del servizio lato server
* Indirizzo IP del client
* Numero di porta del servizio lato client

Un endpoint è una coppia (indirizzo IP, porta).

Una connessione è una coppia di endpoints (endpoint sorgente,endpoint destinazione)

# LEZIONE 2A

## Configurazione delle interfacce di rete: \*nix (IFCONFIG - IP)

La configurazione di una rete in un ambiente Unix è possibile impostarla tramite linea di comando.  
E’ possibile utilizzare il comando ***ifconfig***, che mostrerà le informazioni relative a un interfaccia specifica, come nell’esempio seguente:

Immagine che contiene testo, schermata, software, Carattere

Descrizione generata automaticamente

E’ possibile configurare una nuova interfaccia di rete impostando l’indirizzo IP, netmask e indirizzo di broadcast.

Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente

Altri comandi utili per configurare una rete, sono:

Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente

La stessa interfaccia di rete **può avere più indirizzi IP**, assegnando ad essi un **alias** in modo da poterli distinguere e trattarli come se fossero interfacce fittizie.

Immagine che contiene testo, Carattere, schermata, linea

Descrizione generata automaticamente

**si impostano gli alias come se fossero interfacce fittizie**

**eth0 può avere gli alias eth0:0, eth0:1, eth0:2, …**

Una nuova versione del comando **ifconfig** è **ip**.

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

## Comando IWCONFIG

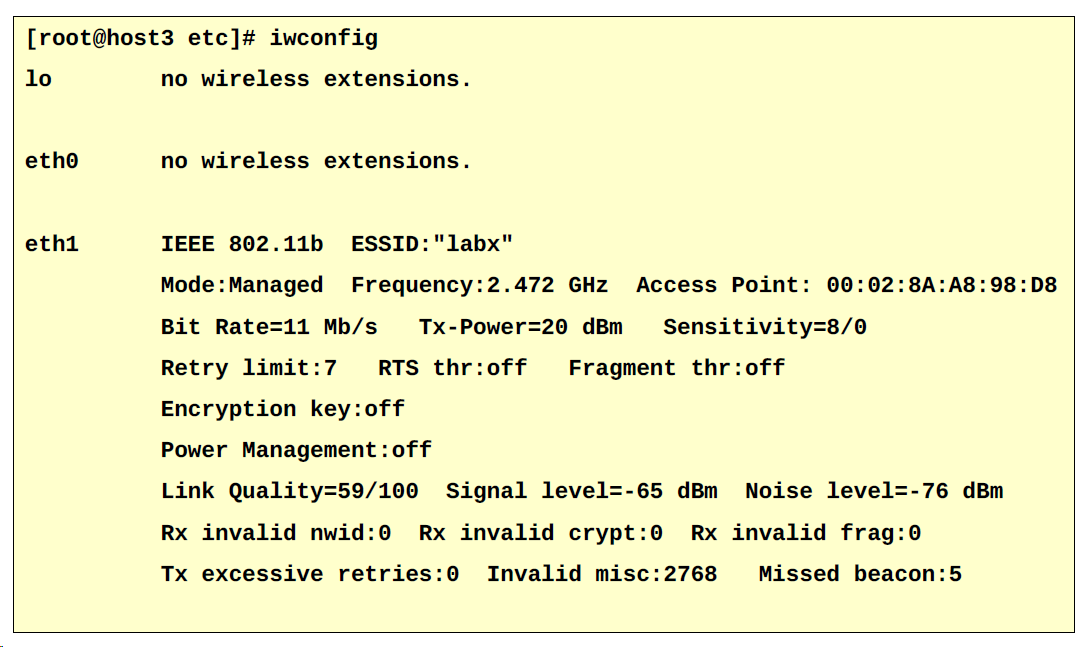
E’ possibile configurare anche le reti WI-FI (IEEE 802.11), con il comando **iwconfig**. Di seguito la sintassi con il quale è possibile configurare la rete:  
**iwconfig <interface> [options]**



Le options possono essere per esempio:

Immagine che contiene testo, Carattere, schermata

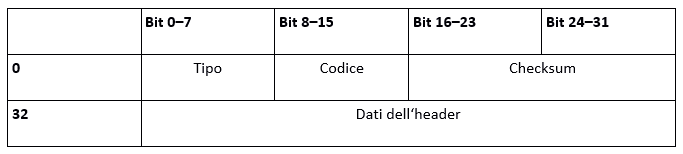
Descrizione generata automaticamente



## Internet Control Message Protocol (ICMP)

**ICMP** è un protocollo del livello **network** per la segnalazione di eventi relativi allo stato della rete. Viene utilizzato, per esempio, quando un host o un router devono informare la sorgente di un datagram circa eventi relativi al trasferimento del datagram. ICMP utilizza IP come se fosse un protocollo di livello trasporto, incapsulando il **messaggio ICMP,** che è individuato da un **codice numerico** composto da un **tipo** e da un eventuale **sottotipo** (code), in un datagram IP (costituisce il campo dati). ICMP è comunque parte integrante di IP e deve essere implementato da ogni modulo IP ed **opera con messaggi di richiesta e risposta.**

Per comprendere meglio il funzionamento del protocollo, conviene dare un’occhiata alla struttura dell’ICMP o meglio del suo **header**. Esso si collega direttamente con l’header dell’IP, nel quale viene indicato dal numero 1 (ICMPv4 se viene usato lo standard IPv4) o 58 (ICMPv6 se viene usato lo standard IPv6) del campo *Protocollo*. L’header dell’Internet Control Message Protocol non è molto ampio e ha la forma seguente:



Il primo campo di 8 bit (*Tipo: messaggio di errore o richiesta di informazioni*) indica la categoria a cui appartiene il messaggio ICMP. Il campo seguente (*Codice*), anch’esso di 8 bit, fornisce una descrizione ulteriore del messaggio. Facendo un esempio, un messaggio ICMP di tipo 3 specifica che **la destinazione del pacchetto di dati non è raggiungibile**, mentre il codice indica se a non essere disponibile sia la rete di destinazione (0), l’host desiderato (1) o la porta (3). L’ultimo campo di 16 bit (*Checksum*) ha la funzione di verificare la correttezza dei dati forniti. La sua struttura è analoga a quella dei checksum degli altri protocolli standard (IP, UDP, TCP).

Per ultimi ci sono i dati ICMP, che sono **creati e strutturati diversamente** in base al tipo di messaggio e all’istanza in funzione. In questo campo vengono spesso menzionati di nuovo l’header dell’IP e i primi 64 bit del pacchetto di dati responsabile del messaggio di errore o dello stato della query. Nel cosiddetto **ICMP Tunneling** questo campo viene utilizzato in modo improprio per inviare dati utente al di là del firewall o per stabilire un canale di comunicazione tra due computer.

**TIPI ICMP**

Immagine che contiene testo, schermata, numero, Carattere

Descrizione generata automaticamente

### Interrogazioni ICMP

Il protocollo ICMP è di grande importanza per la comunicazione nelle reti IP, nelle quali viene utilizzato in particolare dai router, ma non solo: server e client fanno uso dei messaggi legati al protocollo Internet e ricevono in questo modo **importanti informazioni di rete**.

#### Comando ping

Una situazione tipica di impiego è il cosiddetto **ping test**, che può essere eseguito utilizzando l’omonimo comando disponibile in tutti i sistemi operativi. L’utilizzo di questo strumento è il modo più semplice per controllare la disponibilità di un host nella rete. A questo scopo, ping invia un pacchetto ICMP di tipo ***Echo request*** , al quale il destinatario risponde con un messaggio consistente in un pacchetto di tipo ***Echo reply***. Nel caso in cui il sistema a cui è destinato il ping non sia raggiungibile, l’ultimo nodo di rete disponibile invia un pacchetto di risposta di tipo ***Destination unreachable*** (cioè destinazione non raggiungibile).

#### Comando traceroute/tracert

Il comando tracert è un altro esempio d’uso del protocollo ICMP. Permette di tracciare il percorso compiuto dai pacchetti IP da una sorgente ad una destinazione. In pratica elenca i router attraversati nel percorso tra la sorgente e la destinazione. Per determinare gli indirizzi e i nomi dei router il programma tracert invia messaggi ICMP inseriti in pacchetti IP con TTL (Time To Live) via via crescente.

Il primo pacchetto avrà TTL 1 e si fermerà al primo router Il secondo pacchetto avrà TTL 2 e si fermerà al secondo router e così via. Riceve i messaggi di errore (TTL scaduto, tipo 11, codice 0) dai router e costruisce il percorso tra sorgente e destinazione. Il timer associato ad ogni pacchetto IP permette di calcolare il tempo trascorso per il raggiungimento dell’ennesimo router.

*Esempio*

Immaginiamo di voler tracciare il percorso verso il server web di Google. Utilizzeremo il comando **traceroute** da una finestra di terminale. Supponiamo che l'indirizzo IP del server di Google sia 172.217.17.142.

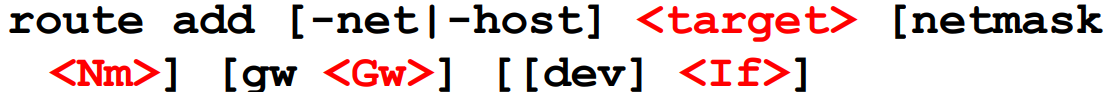
***traceroute*** 172.217.17.142

1. **Invio dei pacchetti ICMP:** **traceroute** invia pacchetti ICMP Echo Request (ping) a 172.217.17.142, con un limite di tempo iniziale (time-to-live, TTL) di 1.
2. **Router 1:** Il primo router lungo il percorso riceve il pacchetto, ma il TTL è insufficiente, quindi il router restituisce un messaggio ICMP "Time Exceeded." Questo ci dice che il pacchetto è passato attraverso il primo router.
3. **Incremento del TTL:** **traceroute** invia un secondo pacchetto con TTL incrementato a 2. Ora, il secondo router lungo il percorso riceve il pacchetto e restituisce anch'esso un messaggio "Time Exceeded."
4. **Ripetizione del processo:** Questo processo viene ripetuto con TTL crescenti fino a raggiungere il server di destinazione.
5. **Destinazione raggiunta:** Infine, quando il pacchetto raggiunge il server di Google, riceveremo risposte ICMP Echo Reply.

I risultati mostreranno l'indirizzo IP di ciascun router intermedio e il tempo impiegato per raggiungerlo. Questo ci dà un'idea del percorso che i pacchetti seguono attraverso la rete per raggiungere la destinazione desiderata.

#### Comando Route

Il comando route consente di visualizzare e manipolare la tabella di routing del kernel. Permette di impostare l’indirizzo del gateway (indirizzo dell’host che inoltra i pacchetti all’esterno della rete; solitamente è il primo indirizzo della sottorete) e anche di aggiungere indirizzamenti nella tabella di routing, con la seguente forma:



Dove:

Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente

Oppure



E’ possibile verificare la tabella di routing con il comando: 

#### DNS

Un name server, o DNS server, è un computer host in grado di fornire la traduzione di un nome di dominio in un numero IP e viceversa. Il computer che fornisce questo servizio è in grado di rispondere direttamente alle richieste riferite ai nomi di dominio di competenza della sua zona, e per gli altri, deve interpellare altri name server competenti.

Il comando ***/etc/resolv.conf*** viene usato per conoscere l'indirizzo o gli indirizzi dei name server di competenza della rete cui si appartiene. Se non si vuole gestire un name server nella propria rete locale, si deve almeno indicarne uno esterno per accedere a Internet.



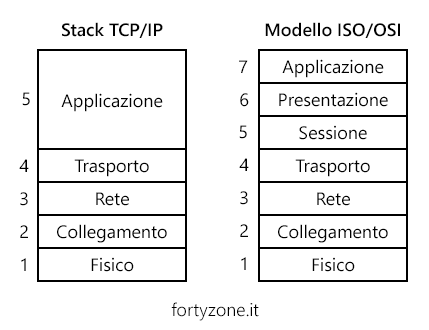
Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

# LEZIONE 2B

## Il livello applicazione

I protocolli di **livello applicazione** si collocano al di **sopra** dello **stack TCP/IP,** immediatamente prima del **livello** di **trasporto**rappresentano l'interfaccia utente al protocollo di rete



Tali protocolli utilizzano **TCP** o **UDP** come meccanismo di trasporto.

A differenza del TCP, le applicazioni basate su UDP devono provvedere a definire delle proprie routine per **gestire il controllo del flusso** e **situazioni di errore**, dato che UDP garantisce prestazioni migliori a causa del ridotto overhead per la mancanza di queste gestioni.

I protocolli che impiegano il livello di trasporto utilizzano la **nozione di porta** per l'individuazione di un **servizio.**

Le porte sono numeri interi a 16 bit (da 0 a 65535, sia per TCP che UDP) che, abbinati agli indirizzi IP, sono utilizzati per stabilire una connessione tra le parti in comunicazione

**lANA** (Internet Assigned Numbers Authority) si occupa dell'attribuzione ufficiale di un servizio ad una porta. Distinguiamo:

* porte “ben note” (da 0 a 1023; servizi base o “storici”);
* porte registrate (da 1024 a 49151, servizi registrati);
* porte dinamiche o private;

## Modello Client Server

Molte applicazioni adottano il **modello client-server**:

gli utenti interagiscono con la parte **client** costruendo la richiesta per un particolare servizio e inviandola alla componente server utilizzando TCP/UDP.

il **server** è un programma che riceve una richiesta, esegue il servizio richiesto ed invia i risultati al client.

La comunicazione è effettuata mediante opportune API, tra cui:

* **Socket API:** Sono un'interfaccia di programmazione che permette la comunicazione tra processi, sia sulla stessa macchina (comunicazione locale) che su macchine diverse in una rete (comunicazione remota).

Una socket è una rappresentazione astratta della connessione tra due programmi. Un processo può creare una socket per ascoltare le connessioni in arrivo o per connettersi a un'altra socket.

* **Remote Procedure Call (RPC) API:** Le RPC (Remote Procedure Call) sono un paradigma di programmazione che permette a un programma di eseguire procedure su un altro sistema come se fossero locali. Esistono diversi framework e protocolli RPC, come JSON-RPC, XML-RPC e gRPC.

## Modello Peer-to-Peer (P2P)

Il modello **peer-to-peer** (**P2P**) è un paradigma di progettazione per le **applicazioni distribuite** in cui le entità partecipanti condividono le proprie risorse per contribuire attivamente alla fornitura del servizio.

A differenza dei modelli client-server tradizionali, in cui un server centralizzato fornisce servizi a clienti remoti, il modello P2P coinvolge partecipanti che agiscono sia come clienti che come fornitori di servizi o risorse.

Quindi, ogni entità (**peer**) partecipa alla fornitura del servizio **agendo** contemporaneamente come **client** e come **server**

Caratteristiche:

* **Sistema distribuito e decentralizzato:** Nel modello P2P, non c'è un'entità centralizzata che controlla o gestisce l'intera rete. I partecipanti sono equivalenti tra loro e collaborano per fornire e accedere alle risorse.
* **Condivisione di risorse:** I partecipanti nel modello P2P condividono le proprie risorse, come potenza di calcolo, larghezza di banda, file e altro. Ogni partecipante può agire sia come cliente che come fornitore di risorse.
* **Autoorganizzazione:** Il modello P2P è progettato per essere resiliente e autoorganizzante. Quando nuovi partecipanti si uniscono alla rete o quando alcuni partecipanti vanno offline, la rete può continuare a funzionare senza dipendere da un singolo punto di fallimento.

Nei sistemi P2P gli utenti accedono alle risorse in seguito ad una fase di **ricerca** di un **nodo** già **connesso.** E’ comunque necessario uno spazio di indirizzamento ed un algoritmo di routing.

* **Architettura overlay:** Spesso, il modello P2P utilizza un'architettura overlay, che è una rete virtuale costruita in cima alla rete fisica. Questa rete overlay facilita le comunicazioni tra i partecipanti e può essere più efficiente nella gestione delle risorse rispetto a una rete puramente fisica.

## Servizi di rete

I **servizi di rete** vengono attivati all'avvio del sistema operativo attraverso la procedura di inizializzazione del sistema, dopo l’assegnazione degli indirizzi alle interfacce di rete e dopo la definizione degli instradamenti.

I servizi sono dei processi che prendono il nome di **demoni** ed operano silentemente in background.

La **gestione** dei demoni può essere:

* **Standalone**: La gestione è autonoma. Si occupano di ascoltare su una determinata porta di rete e provvedono da soli al controllo degli accessi al sevizio. **NFS** (Network File System) è un servizio standalone. NFS opera a livello di sistema operativo, consentendo la condivisione di risorse come se fossero montate direttamente sul sistema del client. Ciò significa che il sistema operativo gestisce direttamente le operazioni di condivisione file senza richiedere l'installazione di software aggiuntivo.
* Gestiti da un processo **supervisore** (Internet service daemon): vengono avviati dal supervisore in caso di richiesta del servizio.

Il supervisore si occupa di ascoltare su tutte le porte dei servizi che controlla.

**FTP** (File Transfer Protocol) è un servizio gestito.

La configurazione di un servizio **standalone** si effettua attraverso **file di configurazione** del servizio stesso, mentre la configurazione dei **servizi gestiti** si effettua tramite i **file di configurazione del supervisore** di rete (**inetd.conf** su sistemi UNIX(-like) che utilizzano il supervisore **inetd**, **xinetd.d/<service>** sui sistemi UNIX(-like) che utilizzano il supervisore **xinetd**).

Sui sistemi windows si utilizza la funzionalità servizi della Microsoft Management Console (MMC).

## Domain Name Service

La gestione degli indirizzi IP in forma numerica è una pretesa inaccettabile dal lato utente.

Per tale ragione agli indirizzi IP sono di solito associati dei nomi.

La **trasformazione di un indirizzo in un nome** può avvenire in due modi:

* Tramite un elenco indirizzo-nome contenuto nel file **/etc/hosts**
* Utilizzando il **servizio DNS** (lavora al livello 7 della pila ISO-OSI)

Uno scenario d’esempio potrebbe essere quello in cui ricerchiamo un particolare indirizzo nel browser(es: [www.google.com](http://www.google.com)). Il browser controllerà in prima istanza se nella sua memoria già c’è la corrispondenza [www.google.com](http://www.google.com) *indirizzo IP*, se non la trova andrà a controllare il file **hosts** della macchina. Qualora non venga trovato neanche nel file hosts, verrà contattato un **server DNS** (primario o secondario (nel caso in cui il primario sia irraggiungibile) per la risoluzione dell’indirizzo.

Solitamente, nel caso in cui il client non abbia un indirizzo IP statico, ma è assegnato automaticamente dal DHCP, verrà contattato in prima istanza il server DNS dell’ISP (Internet Service Provider)

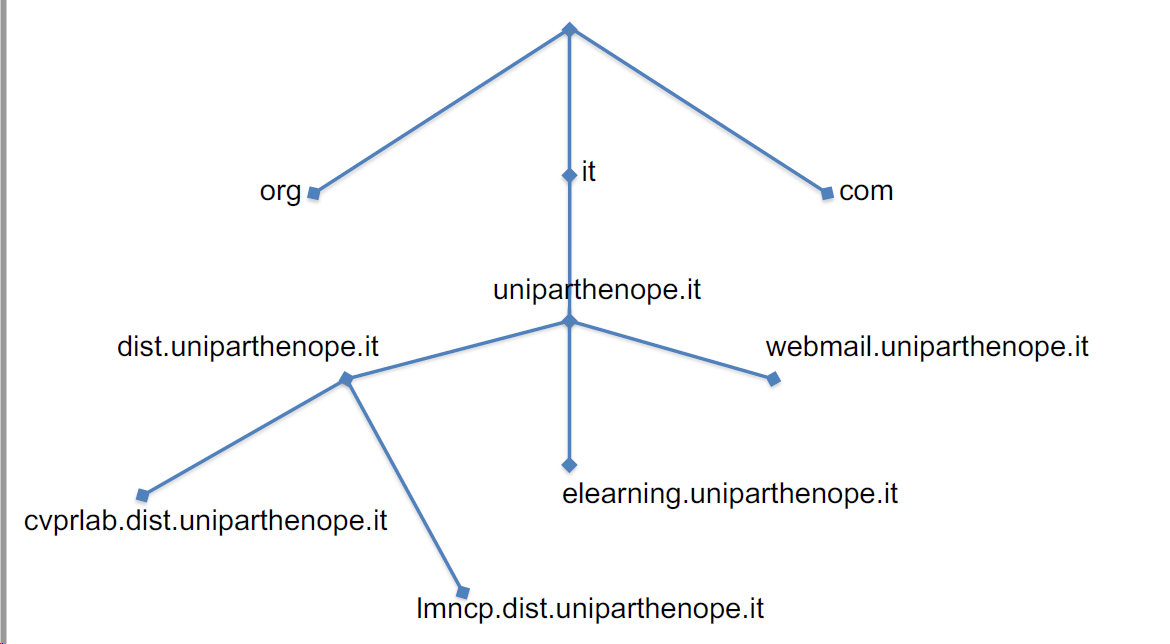
L’utilizzo del DNS impone l’utilizzo della convenzione dei **nomi di dominio.**

Tale convenzione è rappresentabile da una **struttura ad albero**, dove la **radice dell’albero** è il dominio principale, rappresentato da un punto singolo (solitamente è sottinteso)

Ogni **nodo** dell’albero è un **dominio.**

Il **nome di un dominio** si ottiene effettuando l’unione dei nomi dei nodi attraversati per giungere alla radice, separando tali nomi l’un dall’altro con un punto.

I **noti terminali** dell’albero corrispondono agli **host** di rete.



I nomi di dominio utilizzati in Internet si ottengono attraverso una fase chiamata registrazione.

La registrazione avviene facendo una richiesta all’autorità competente per la zona a cui appartiene il nome. Generalmente si registrano nomi di secondo livello, per cui ci si rivolge all'**autorità di registrazione (RA)** competente per il dominio di primo livello.

La RA di gerarchia più alta di tutte è l'Internet Corporation For Assigned Names and Numbers (**ICANN**), ovvero un'organizzazione no-profit responsabile della gestione e del coordinamento del DNS a livello mondiale, che:

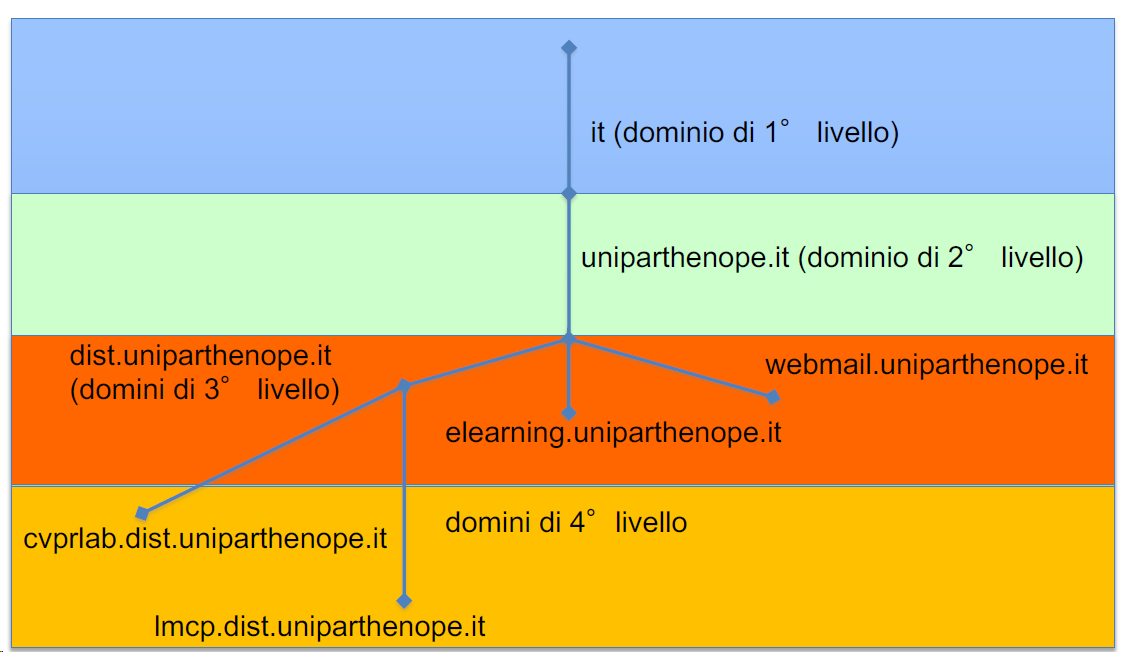
* è competente per la zona relativa alla radice dell'albero dei domini
* demandando la gestione dei domini di primo livello ad opportune RA, le Top Level Domain RA (**TLD RA**)
* delega **13 named authorities** ubicate in diverse parti del mondo, con funzione di **root server**
* ciascuno dei root server dell'ICANN contiene uno stesso database di informazioni nomi-indirizzi, relativo alle TLD RA.

Per visualizzare le **informazioni sui domini** come l’azienda a cui è intestato il dominio, il dominio di validità, i nomi DNS e/o gli indirizzi IP degli host che fungono da server autoritativi è possibile utilizzare il **protocollo NICKNAME (WHOIS).**

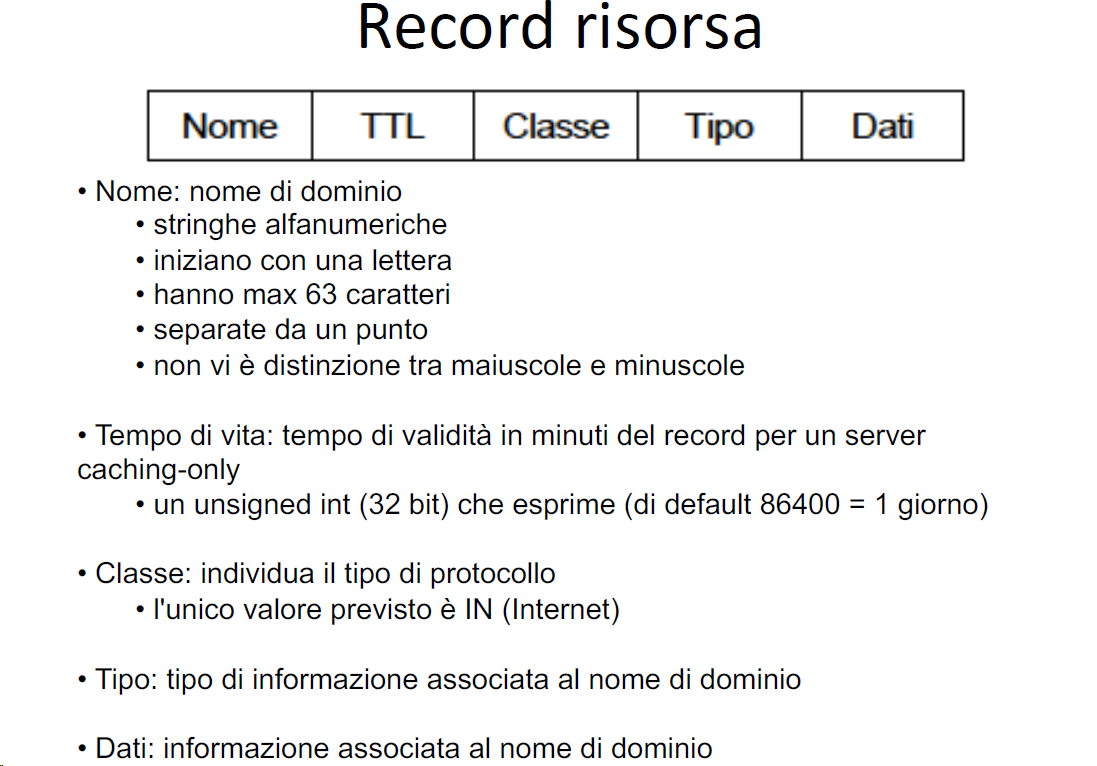
Il DNS è un **servizio distribuito**, ovvero un **insieme di server** di nomi (host su cui è in esecuzione il servizio), ciascuno con una propria **zona di competenza.**

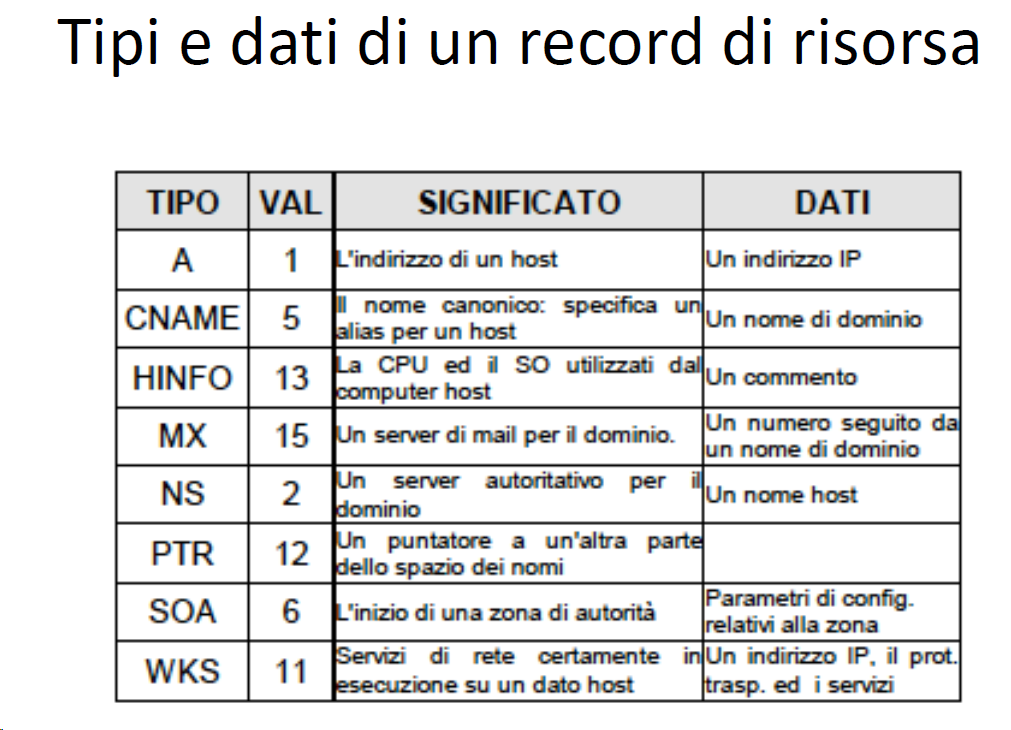
Una **richiesta di traduzione** nome-indirizzo è inviata al server di zona; se tale informazione non è disponibile, invia la richiesta e recupera l'informazione da un altro server DNS.

Le **zone di competenza si sovrappongono all'albero dei domini,** in modo da garantire che una qualsiasi richiesta possa essere soddisfatta grazie all'interrogazione di una opportuna catena di server



Ogni zona organizza le informazioni di sua competenza in **record di risorsa** che definiscono l'**associazione** tra un **nome** di dominio ed un'altra **informazione**, **in base al tipo di record.**





Per risolvere un nome in un indirizzo ci sono due modalità:

1. **Risoluzione diretta (query iterativa):**

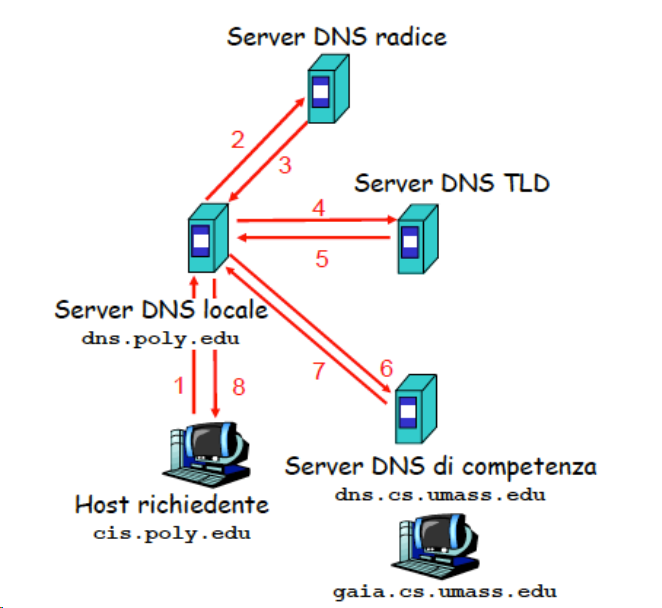
Supponiamo che l’host “PIPPO” voglia risolvere l’indirizzo [www.pluto.com](http://www.pluto.com)

L’host gira la richiesta di traduzione al server locale DNS, definito nella configurazione dell’host stesso (se l’IP è assegnato dal DHCP, il server locale sarà quello dell’ISP).

Il server locale contatta un **root server** (13 nel mondo di 12 organizzazioni, ma ci sono le repliche) che si occupa di andare a risolvere la prima parte dell’indirizzo (**com.**). Il root server risponderà con l’indirizzo del **server TLD** da contattare.

Il server DNS locale contatterà tale **server TLD** che ha la funzione di sapere tutto quello che serve per un determinato dominio (nel nostro caso **com.**) e quindi conosce chi ha registrato tale dominio. Infatti il server TLD risponderà con l’indirizzo dell’**authoritative server**, il quale, essendo responsabile della registrazione del sito web, conosce l’indirizzo IP di [www.pluto.com](http://www.pluto.com) .

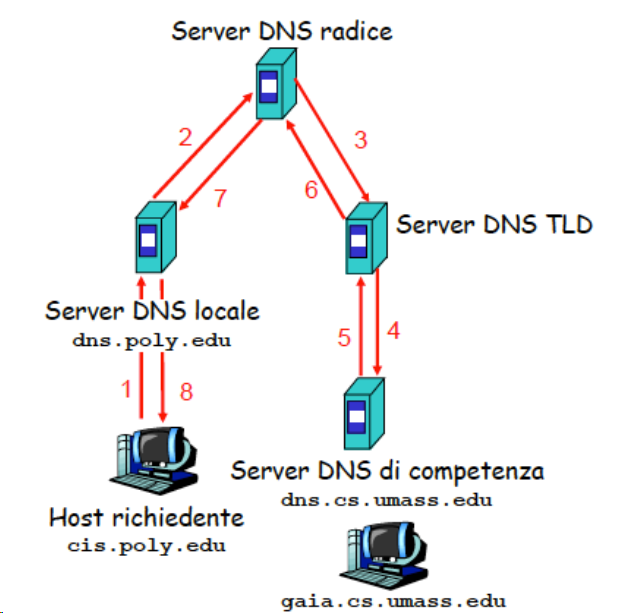
A questo punto il server DNS locale memorizza la corrispondenza e restituisce la risposta all’host PIPPO.



1. **Risoluzione diretta (query ricorsiva):**

Il server locale affida al server contattato il compito di tradurre il nome. Ogni server che non è in grado di rispondere si rivolge esso stesso ad un altro server ed attende la risposta.

La risposta torna al server locale seguendo lo stesso percorso.



1. **Risoluzione indiretta:**

Il DNS fornisce anche un meccanismo per la traduzione di un indirizzo IP in un nome di domino (risoluzione inversa).

Questo meccanismo, vista la natura degli indirizzi IP, non può sfruttare l'approccio gerarchico usato nella ricerca di un IP a partire da un nome di domini, infatti necessita di uno **spazio dei nomi specifico**.

Il dominio utilizzato è **in-addr.arpa**, per il quale gli indirizzi IP sono rappresentati al contrario (es: 192.168.1.5 diviene 5.1.168.192), per omogeneità con la codifica dei nomi di dominio

Quindi, in sostanza, L'interrogazione di un servizio DNS corrisponde all'interrogazione di una base

di dati distribuita, in cui il risultato è il record desiderato.

La base di dati del DNS ha due scopi

1. trovare l'indirizzo numerico associato ad un nome
2. trovare il nome a partire da un indirizzo numerico

Esistono diverse modalità per interrogare il servizio direttamente.

Il programma classico per l'interrogazione del DNS è **nslookup**, disponibile sia su sistemi di tipo Unix che Windows.

Su alcuni sistemi Linux, ad nslookup sono preferiti programmi quali **host** e **dig**, e l'implementazione di nslookup può essere solo parziale.

Nell'implementazione completa **nslookup** prevede il comando **ls**, che permette di ottenere l'elenco di tutti i record presenti sul server DNS, ma generalmente viene impedito.

Esempi:

• nslookup google.com [server dns]

• nslookup –type=mx google.com [server dns]

• nslookup –type=ns google.com [server dns]

• nslookup –type=any google.com [server dns]