

PARTE 6

DOMAIN NAME SYSTEM (DNS)

Ingredienti di Internet

Protocolli

- packet switching
- protocollo IP
- protocollo TCP/UDP
- protocolli applicativi

Naming

- indirizzi IP (*netid, hostid*)
- classi di indirizzi IP
- hostname
- sistema DNS
- name server
(*locali, root, top-level, autoritativi*)

Componenti

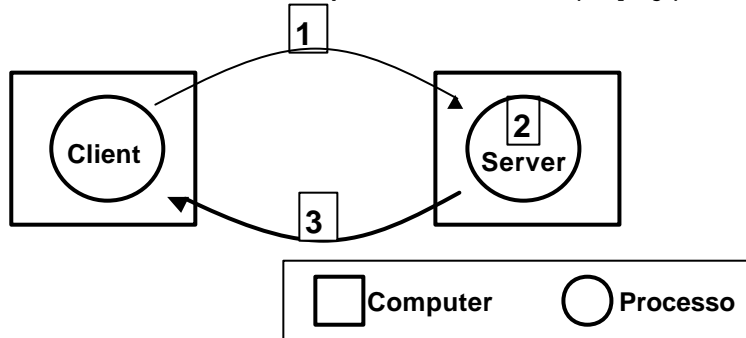
- Host - router
- Modalità di accesso
(diretto, dialup, ISP, POP, NAP)
- Link fisici di comunicazione
- Tipologie di reti
(LAN, Ethernet, MAN, WAN)

Applicazioni di rete

(Non fanno propriamente parte di Internet, ma ne costituiscono la ragion d'essere)

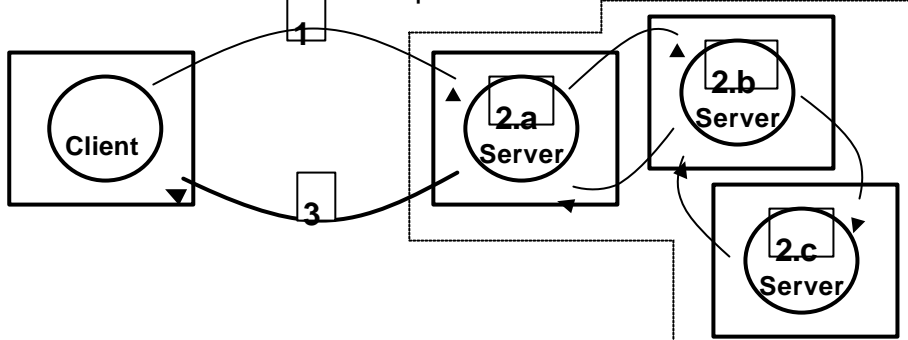
Modello client-server

1. Trasmissione di una richiesta dal processo client al processo server (**request**)
2. Elaborazione della richiesta da parte del server (**processing**)
3. Trasmissione di una risposta al client (**reply**)



Modello client-server (servizio fornito da server multipli)

1. Trasmissione di una richiesta dal processo client al processo server
2. Elaborazione della richiesta da parte del server con invocazione di elaborazioni da parte di uno o più processi
3. Trasmissione di una risposta al client



Domain Name System

***(Un esempio di sistema
distribuito geograficamente
che funziona molto bene)***

Indice

- **Identificatori degli host e alternative nel naming**
- **Organizzazione logica dello spazio dei nomi**
- **Componenti del DNS**
- **Meccanismo distribuito di risoluzione dei nomi**
- **Consistenza ed efficienza del DNS**

Modulo 1: Identificatori degli host

“Identificatori” in Internet

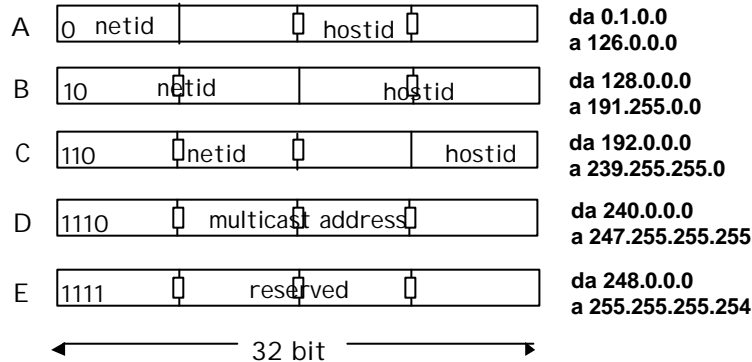
- Le persone hanno molteplici “identificatori”: nome, n° passaporto, codice fiscale, ...
- I telefoni sono denotati da un unico numero
- Tutti i dispositivi collegati ad Internet (*host, router*) hanno almeno due identificatori:
 - **Indirizzo IP (*numero di 32 bit*)**: utilizzato per indirizzare ed instradare i pacchetti nella rete
 - **Hostname (*stringa alfanumerica di al più 255 caratteri*)**: nome logico utilizzato dalle persone

Oltre agli indirizzi IP, ci sono ...

5 classi di indirizzi IP:

classe A, classe B, classe C, classe D, classe E

Classe



... gli *hostname*

- Sequenza di *label* separate da punti
- Ogni label si compone di al più 63 caratteri alfanumerici
- L'intero hostname può essere di al più 255 caratteri

Esempi

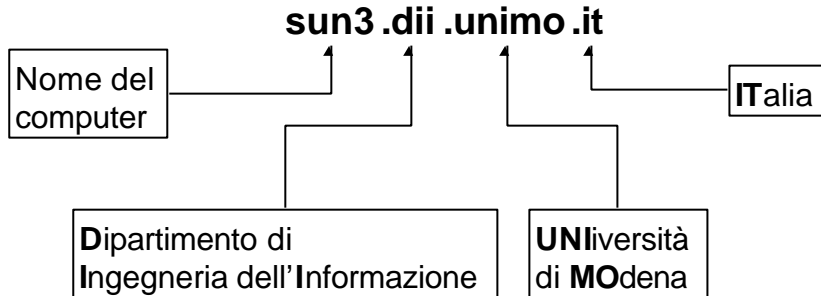
- w3c.org
- samba.ing.unimo.it
- www.unimo.it
- promo.dim.ing.unimo.it

Si noti che non c'è corrispondenza tra le label dell'hostname ed i quattro campi dell'indirizzo IP

Hostname (*canonico*)

- Dato lo scopo rivolto verso l'utente, all'hostname si preferiscono attribuire valori mnemonici

Es.: nome del computer e dominio di appartenenza:



Altre motivazioni oltre l'usabilità

- Specificare l'indirizzo di un host con un valore costante renderebbe l'interazione client/server più veloce, ma:
 - il software del client dovrebbe essere ricompilato ogni volta che il server venisse spostato
 - il client non potrebbe usare più di un server
- L'hostname è un buon compromesso che consente:
 - il collegamento (**binding**) tra hostname e indirizzo IP fino a tempo di esecuzione
 - l'uso di alias per un hostname

Tipico compromesso: maggiore flessibilità a costo di perdita di efficienza

Mapping tra hostname e indirizzo IP

- **Rete con pochi nodi**: *soluzione centralizzata con uno spazio piatto dei nomi*

Per es.,

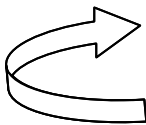
- inizialmente il database centralizzato del NIC
- file **hosts.txt**

- **Rete con milioni di nodi**: *soluzione distribuita con uno spazio gerarchico dei nomi*

- **Domain Name System (DNS)**, operativo dal 1985

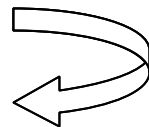
Domain Name System (DNS)

- Realizza uno **spazio dei nomi gerarchico** e permette la traduzione del nome mnemonico di un host in un indirizzo IP e viceversa. Es.



Hostname: **sun3.dii.ing.unimo.it**

Indirizzo IP: **134.56.26.68**
(10001110.00111000.00011010.01000100)



- Implementa un meccanismo efficiente (mediante multipli *name servers*), distribuito su scala geografica, per “risolvere” un hostname in un indirizzo IP e viceversa.

Obbiettivi progettuali del DNS

- Spazio dei nomi **consistente**
- Sistema con elevata **tolleranza ai guasti**
- Sistema **scalabile**
 - Partizionamento del database dei nomi
 - Organizzazione distribuita con possibilità di caching dell'informazione in più punti
 - Decentralizzazione del meccanismo di registrazione degli indirizzi
- Sistema funzionante in **reti eterogenee**
 - soggette a diverse amministrazioni che possono avere differenti politiche di gestione
 - indipendente dal sistema di comunicazione, dai protocolli utilizzati e dal tipo di piattaforme sottostanti

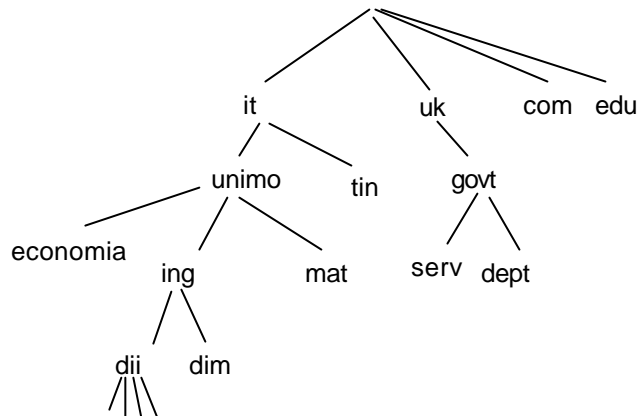
Parte 6

Modulo 2: Organizzazione logica dello spazio dei nomi

Organizzazione gerarchica dei domini

- dii.ing.unimo.it
- dept.govt.uk

La stringa relativa ad un host non ha limiti nel numero di campi



Esempio: Università di Modena “unimo”, con tre Facoltà (“economia”, “lettere”, “ing”), quest’ultima con due dipartimenti Informazione (“dii”) e Meccanica (“dim”), nel primo dei quali sono registrati diversi host

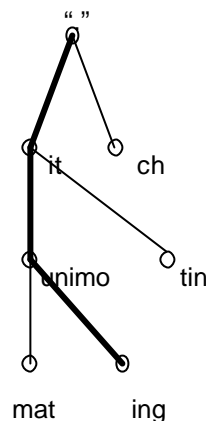
Organizzazione gerarchica

root
name server

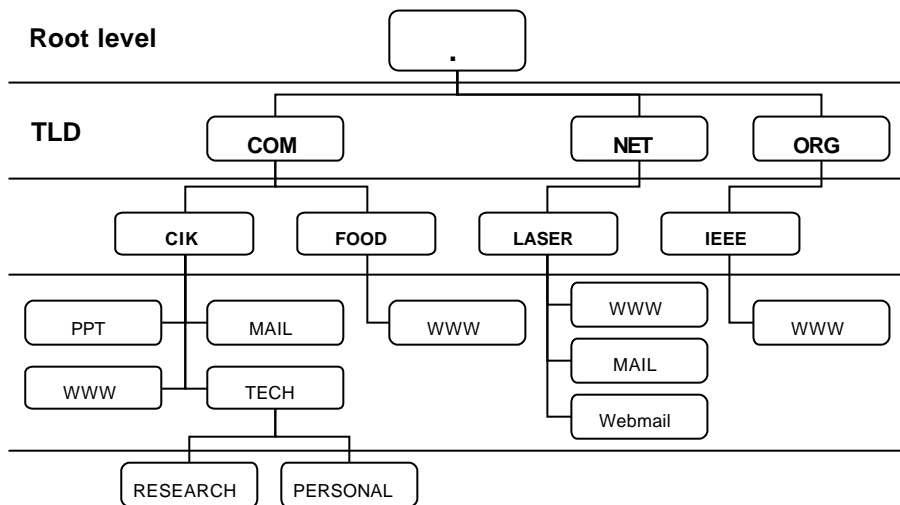
it
name server

unimo.it
name server

ing.unimo.it
name server



Esempio di organizzazione



Assegnamento dei nomi

- Il suffisso (corrispondente al TLD) è assegnato in maniera univoca dall'*Internet Authority*
- Un'organizzazione sceglie il nome desiderato di **secondo livello**. (Tale nome deve essere unico, e vi sono nomi soggetti a leggi internazionali per trademark, copyright, ecc.)
- Un'organizzazione registra il nome presso l'autorità centrale delegata alla gestione del top-level domain
- A questo punto, l'amministratore del dominio ha controllo completo sugli altri campi del dominio
 - Il significato dei segmenti (sotto-domini) è delegata all'organizzazione
 - Non vi è limite sul numero di sotto-domini o numero di livelli
 - Lo spazio dei nomi non è correlato ad una interconnessione fisica. Per esempio, `mat.unimo.it` e `dii.unimo.it` potrebbero trovarsi sullo stesso piano di un edificio o in città differenti

Domini di massimo livello *Top Level Domain (TLD)*

Nome del Dominio	Significato
COM	Organizzazioni commerciali
EDU	Istituzioni USA per l'istruzione
GOV	Istituzioni governative USA
MIL	Istituzioni militari USA
NET	Maggiori centri di supporto per la rete
ORG	Organizzazioni senza scopo di lucro diverse dalle precedenti
ARPA	Dominio temporaneo della rete ARPANET (<i>obsoleto</i>)
INT	Organizzazioni internazionali (<i>schema geografico</i>)
Codice nazionale (it, ch, fr, id, ...)	Nomi nazionali (<i>schema geografico</i>)

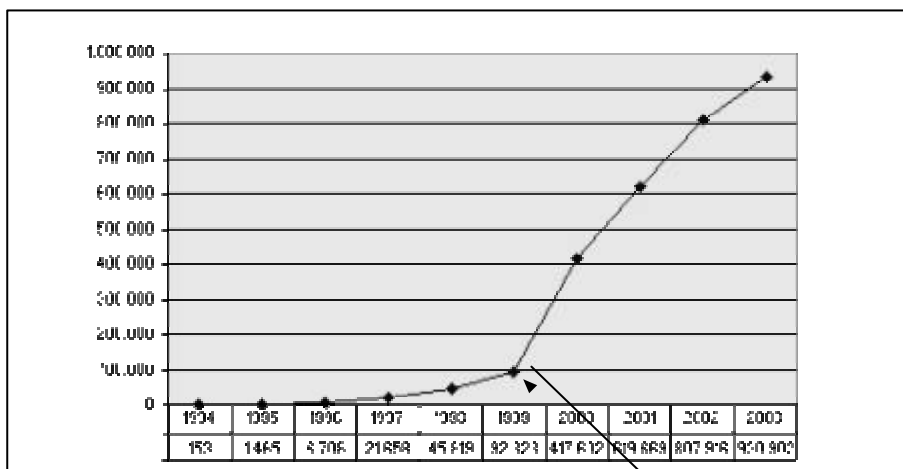
I nuovi TLD

- ICANN e IETF hanno approvato i seguenti nuovi Top Level Domain (i primi 3 attivi da gennaio 2002)
 - **.biz** (**www.NIC.biz**)
 - **.info** (**www.NIC.info**)
 - **.name** (**www.NIC.name**)
 - **.pro**
 - **.museum**
 - **.coop**
 - **.aero**
- Altri TLD al vaglio di ICANN e IETF:
 - **.eu** (*europe*)
 - **.pid** (*personal identifier*)
 -

Enti responsabili della registrazione

- **NIC: il primo (e unico ente) responsabile del DNS**
- **ICANN: gestione dei server root e nuovi domini**
- **InterNIC: domini .org, .net, .edu, .com (quest'ultimo con deleghe nazionali)**
- **IANA: domini .us e standard di Internet**
- **RIPE: domini europei**
- **APNIC: domini asiatici**

Domini registrati in Italia (Top Level Domain .it)



Registration Authority italiana: Istituto CNR

Liberalizzazione
delle registrazioni

Organizzazione distribuita del DNS

- Ciascuna organizzazione che possiede e gestisce un dominio è responsabile dell'operatività di un name server che:
 - deve essere registrato presso il dominio gerarchicamente superiore (p.es., TLD→Root)
 - deve fornire il mapping tra tutti gli hostname del dominio ed i rispettivi indirizzi IP

Esempio

- Qualche name server gestito dall'organizzazione **UNIMO** è responsabile degli hostname in **unimo.it**

Parte 6

Modulo 3: Componenti del DNS

Componenti del DNS - INDICE

- **Domain Name Space e Resource Records**

→ Ovvero i dati: zone, descrittori, informazioni



- **Name Server**

→ Ovvero i possessori e gestori dell'informazione, con le funzionalità di **server** abilitati a rispondere alle query dei client

- **Resolver**

→ Ovvero i (primi) **client** del sistema DNS che sottomettono query per informazioni su hostname e indirizzi IP per conto delle applicazioni di rete

Definizioni

- **Domain name per un host**

Sequenza di label che va dall'**hostname** (*la foglia dell'albero di naming*), costituita dalla label più a sinistra, al **top** dell'albero di naming mondiale, costituita dalla label più a destra

- **Dominio** (riferito alla **struttura gerarchica dei nomi**)

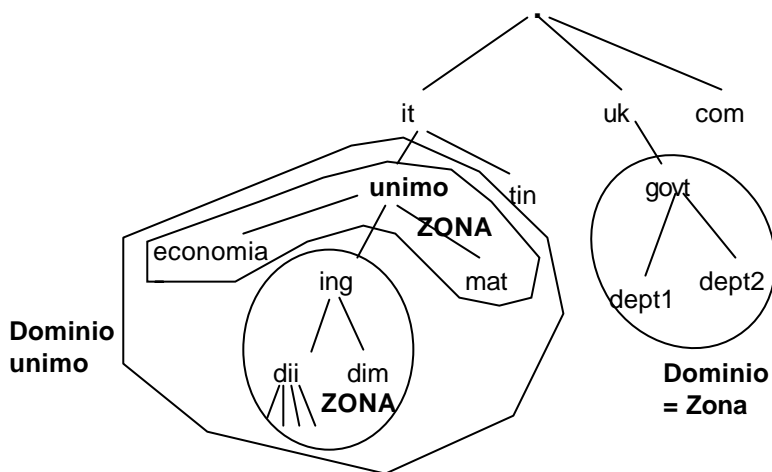
Sottoalbero dell'albero di naming mondiale

- **Zona** (riferito all'**organizzazione dei name server**)

Dati relativi ai nomi di un Dominio, meno qualche sotto-dominio quando è amministrato da autorità di livello inferiore

~~Zona e Dominio possono coincidere o meno~~

Zone e Domini



Dati del database di una ZONA

1. Dati relativi a tutti i nomi di un Dominio, meno alcuni sotto-domini amministrati da autorità di livello inferiore. Per esempio, una zona potrebbe contenere i dati di **ing.unimo.it**, meno quelli relativi a **weblab.ing.unimo.it**
2. Hostname ed indirizzi IP del o dei name server che forniscono **dati autoritativi** per la **Zona** (si possono ritenere consistenti e ragionevolmente aggiornati)
3. Hostname ed indirizzi IP dei name server che possiedono dati autoritativi per **sotto-zone** delegate
4. Parametri relativi alle modalità di gestione della Zona. Es.
 - per gestire caching/replica delle informazioni
 - per gestire modalità e frequenza degli aggiornamenti

Resource Records (RR) (*Descrittori di risorsa*)

- Legati ai nodi nell'albero del DNS
 - Tutti i nodi terminali hanno RR
 - La maggior parte dei nodi non terminali hanno RR
 - Tutti i RR in una zona hanno la classe della zona

- **Ciascun RR contiene:**

- | | |
|-------------------------|--------------------|
| – Nome del dominio | www.unimo.it |
| – RR Time-To-Live (TTL) | 86400 (in secondi) |
| – RR Class | IN (=INternet) |
| – RR Type | A (=Address) |
| – RR Data | 134.56.26.68 |

Informazioni del DNS

- I database del DNS contengono altre informazioni oltre ai resource-record A con informazioni hostname-to-address:
 - Name server records NS
 - Hostname aliases CNAME
Un host può avere più hostname di cui uno canonico (**canonical name**) ed altri **alias**
 - Mail Exchangers MX
 - Host Information HINFO

Tipi di Resource Record (RR Type)

- **SOA:** Start Of Authority (parametri per gestire la zona)
- **NS:** Name Server autoritativo per una zona
- **A:** Host Address (indirizzo IP)
- **MX:** Mail eXchanger (dominio che accetta email, coppie <host, pref.>)
- **CNAME:** Canonical NAME per un alias (alias del nome)
- **PTR:** PoinTeR to another node (nome per indirizzo IP, per reverse lookup)
- **HINFO:** Host Information (descrizione CPU e Sistema Operativo)
- **TXT:** arbitrary TeXT (in formato ASCII)

Esempio RR: *Start of Authority* (SOA)

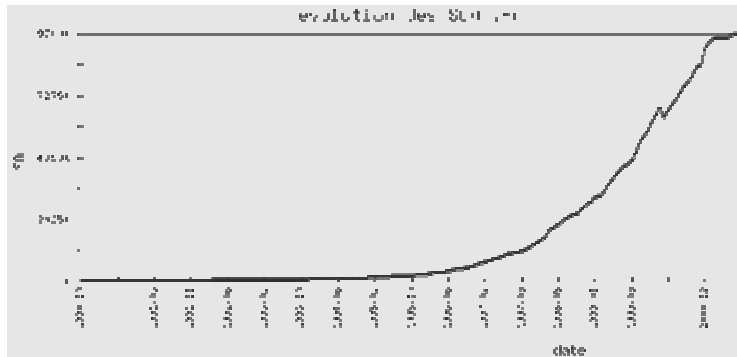
- **TTL – Time To Live** - Determina per quanto tempo il record sarà valido sul server, senza richiedere un refresh
- **Serial** – identificatore seriale di aggiornamento; server per verificare che un server secondario abbia l'ultimo record con i dati più aggiornati
- **Refresh** – indica ad un server secondario quanto frequentemente deve richiedere un aggiornamento al server primario
- **Expire** – Tempo limite che indica per quanto tempo un file di ZONA può essere servito; utilizzato solo nel caso in cui il server primario non risponde per un lungo periodo di tempo
- **Retry** – Se il server secondario richiede un refresh ed il primario è irraggiungibile, il valore Retry indica quanto tempo attendere prima di provare nuovamente

Esempio

```
@ IN SOA mcs.vuw.ac.uk mark.comp.vuw.ac.uk (  
    199610140 ; Serial number  
    28800     ; Refresh 8 hours  
    7200      ; Retry 2 hours  
    604800    ; Expire 7 days  
    86400 )   ; Minimum 24 hours
```

Start Of Authority (SOA)

- Crescita del numero degli SOA ovvero delle Zone gestite in modo autonomo e delegato nell'ambito del TLD .fr



Esempio RR: *Name Server (NS)*

- Specifica i server che contengono dati autoritativi relativi ad una Zona
- In particolare, indica il server primario e le informazioni sui server secondari che vengono utilizzati nel caso in cui il primario è irraggiungibile.
- Quando si aggiungono nuovi server alla **RootZone** per il dominio, i relativi hostname e indirizzi devono essere aggiunti manualmente al file della **LocalZone**

Esempio

@ IN NS downstage.mcs.vuw.ac.uk

Altri esempi di RR

- **A -Address**

Il tipo di record più comune. Usato per il riferimento da un nome (es., WWW) all'indirizzo IP di un host (es., 66.26.153.214). Le implementazioni più vecchie non consentivano che due record A contenessero lo stesso indirizzo IP. Questo vincolo è stato rimosso a causa della diminuzione di disponibilità di indirizzi IP.

```
embassy IN A 130.195.6.15  
proto IN A 130.195.5.12
```

- **CNAME - Canonical Name**

Tipicamente utilizzato per unificare i record e limitare le modifiche da effettuare quando un indirizzo IP cambia. Un CNAME agisce sia come record MX sia come record A.

Un CNAME non può contenere un indirizzo IP. Deve essere un alias di un record A già esistente.

```
www IN CNAME proto
```

- **MX – Mail Exchanger**

L'unico record che consente di specificare una "priorità". Specifica a quale server inviare una e-mail in arrivo. E' possibile specificare fino a 128 server, ciascuno con una priorità differente. La priorità più comune è pari a 10.

- **AAAA – IPv6 Address**

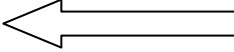
Identico ad un tipico record A -Address, con l'unica differenza che gestisce i nuovi indirizzi IP del protocollo IP versione 6. Ci si aspetta che i record AAAA prevalgano sui record A entro il 2006

Esempio di file DNS

```
laser.net. IN NS ns1.granitecanyon.com.  
laser.net. IN NS ns2.granitecanyon.com.  
laser.net. IN NS ns1.secondary.com.  
laser.net. IN NS ns2.secondary.com.  
laser.net. IN TXT "LASER lmtd., NIC Handle: CIK72"  
laser.net. IN A 64.39.26.186  
laser.net. IN MX 10 mail.laser.net.  
www IN CNAME laser.net.  
mail IN A 63.116.218.133  
imap IN CNAME mail.laser.net.  
smtp IN CNAME mail.laser.net.  
webmail IN CNAME laser.net.  
localhost IN A 127.0.0.1  
ftp IN CNAME CIK.NO-IP.COM
```

Uso di NSLOOKUP

Componenti del DNS - INDICE

- Domain Name Space e Resource Records
→ Ovvero i dati: zone, descrittori, informazioni
- Name Servers 
→ Ovvero i possessori e gestori dell'informazione, con le funzionalità di **server** abilitati a rispondere alle query dei client
- Resolvers
→ Ovvero i (primi) **client** del sistema DNS che sottomettono query per informazioni su hostname e indirizzi IP per conto delle applicazioni di rete

Gerarchia dei server

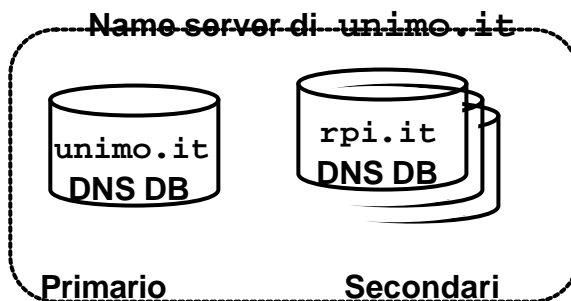
- I name server non hanno i dati di tutti i nomi
 - I name server devono conoscere quali altri server sono responsabili di altre zone
 - Tutti i server devono conoscere i root name server
 - I root name server devono conoscere i server dei TLD
 - Ciascun name server deve conoscere almeno il name server della zona immediatamente superiore (p.es., il name server della zona **ing** deve conoscere il name server della zona **unimo**) e viceversa
 - Tuttavia, ciascun amministratore di una zona, può inserire tra i propri RR-NS, altri name server
 - Ne risulta una gerarchia di name server differente e molto più irregolare rispetto alla gerarchia dei nomi di dominio
- NOTA
- Un singolo livello della gerarchia può essere partizionato tra multipli server
 - Un singolo server può servire più zone

Server primari e secondari

- Gli amministratori di sistema inseriscono i dati relativi ad una Zona in un **master file**, che è l'unica sorgente di dati autoritativi per quella Zona
- Ci sono due tipi di name server che possono fornire dati autoritativi:
 - **Primary o master server**, che leggono i dati su di una Zona direttamente dal master file
 - **Secondary server**, che scaricano i dati di una Zona dal rispettivo server primario

Server primari e secondari (cont.)

- Per ciascuna Zona, c'è sempre un **server primario** e vi può essere un certo numero di **server secondari**, che contengono una copia del database hostname-indirizzi IP



Server primari e secondari (cont.)

- I **server secondari** comunicano periodicamente con il **server primario** per verificare se i propri dati sono consistenti rispetto a quelli contenuti nel primario
- Se una copia di dati di un **server secondario** non è aggiornata, il **server primario** invia l'ultima versione
- La frequenza delle verifiche da parte del **server secondario** è stabilita dall'**amministratore del sistema**, ed il suo valore è tipicamente di uno/due volte al giorno

Gestione del caching dei dati

- Ogni name server del DNS è libero di effettuare il caching dei dati relativi ad altri server ed altre zone in modo da evitare di contattarli quando una risoluzione viene richiesta più volte
- I client che ricevono dati dalle cache dei name server sono informati che ciascun dato è fornito "as it is" e non è da considerare autoritativo
- A ciascun dato in una zona si assegna un valore **time-to-live (TTL)**
- Quando un name server non-autoritativo ottiene un dato da un server autoritativo prende nota del TTL associato
- Il name server fornirà un dato nella cache al client che ne fa richiesta solo se il relativo TTL non è scaduto
- Se invece il TTL è scaduto, il name server contatta il name server autoritativo per controllare se il dato è valido o meno

Valore del TTL

- Il caching è uno strumento potente per ridurre il traffico di Internet relativo alla risoluzione indirizzi
- La scelta del valore del TTL deve seguire tale scopo ed è a carico dell'amministratore della zona:
 - Quando ci si aspetta che i dati di una zona cambino con poca frequenza, l'amministratore dovrà utilizzare TTL elevati
 - Al contrario, in zone soggette a frequenti cambiamenti, è opportuno che il TTL abbia valori bassi

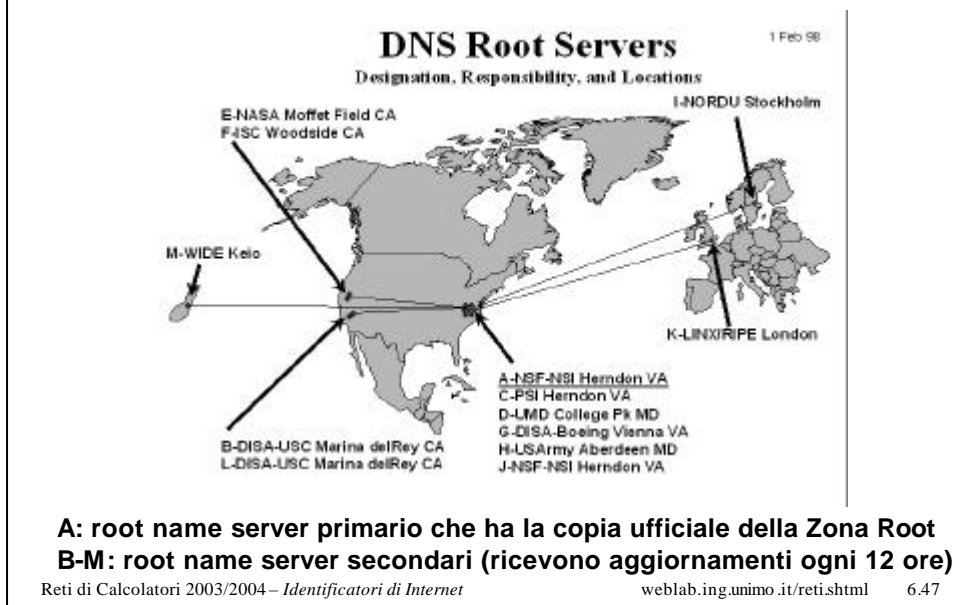
Classi di name server

- Root name server (.) – la radice dell'albero dei domini
- TLD name server (relativi ai domini top-level:
.com, .edu, .org, .it, .uk, ...)
- Intermediate name server
- Local name server (ogni ISP e organizzazione gestisce uno o più name server locali, foglie dell'albero)

Altri attributi relativi ai name server

- Name server primario (mantiene i dati della zona)
- Name server secondario (mantiene copie dei dati della zona)
- Authoritative name server (autoritativo relativamente ai dati di una zona)

I 13 Root name servers (1998)



Nuova organizzazione dei root name server

Nome	Organizzazione	Luogo	URL
A	Network Solutions, Inc	Herndon, VA, USA (<i>segreto</i>)	http://www.netsol.com
B	Information Sciences Institute		
	University of Southern California	Marina Del Rey, CA, USA	http://www.isi.edu
C	PSINet	Herndon, VA, USA	http://www.psi.net
D	University of Maryland	College Park, MD, USA	http://www.umd.edu
E	NASA	Mountain View, CA, USA	http://www.nasa.gov
F	Internet Software Consortium	Palo Alto, CA, USA	http://www.isc.org
G	Defense Inform. Systems Agency	Vienna, VA, USA	http://nic.mil
H	Army Research Laboratory	Aberdeen, MD, USA	http://www.arl.mil
I	NORDUNet	Stockholm, Sweden	http://www.nordu.net
J	(TBD)	Herndon, VA, USA	N/A
K	RIPE-NCC	London, UK	http://www.ripe.net
L	(TBD)	Marina Del Rey, CA	USA N/A
M	WIDE	Tokyo, Japan	http://www.wide.ad.jp

Piattaforme dei root name server

- I **root name server** sono le macchine che forniscono accesso al file della **“root zone”** per le necessarie operazioni di **DNS resolution**
- A causa dei limiti del protocollo DNS utilizzato (basato su UDP), il numero di macchine è attualmente limitato a 13, sebbene siano in corso dei tentativi per rimuovere questo limite
- Dopo lunghi dibattiti, si è arrivati a diversificare l'amministrazione dei 13 *root name server*, che attualmente sono gestiti da enti militari statunitensi, organizzazioni commerciali e non-profit, Internet service providers, università, e istituti di ricerca. In particolare, 3 dei 13 server sono localizzati al di fuori degli Stati Uniti (uno a Londra, ma amministrato dall'Olanda, uno in Giappone, uno in Svezia)
- **“Internet continuerà a funzionare, anche se 2/3 dei root name server dovessero risultare irraggiungibili”** (RFC-2870)

Piattaforme dei root name server (cont.)

- Tutti e 13 i name server sono posti in ambienti controllati e protetti anche da contingenze ambientali, che includono limitazioni e controlli sugli accessi fisici, protezioni contro incendi, allagamenti, e black-out (con generatori autonomi), diverse connessioni ad Internet (dal livello 1 al livello 3)
- I root name server B-M sono costituiti da più macchine, anche se ciascun name server è “logicamente uno”. Il root name server A è fisicamente costituito da una macchina, anche se in caso di emergenza le sue funzionalità possono essere rimpiazzate dal server Z
- Tutti i root name server utilizzano qualche variante del sistema operativo Unix (*assolutamente non Windows...*). Tuttavia, sia l'hardware sia il sistema operativo su cui si basano i name server è estremamente eterogeneo: tra i 13 name server, si contano 7 diverse piattaforme hardware con 8 diverse versioni di sistema operativo forniti da 5 diversi rivenditori

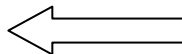
Funzionamento dei Root name server

File della “root zone”

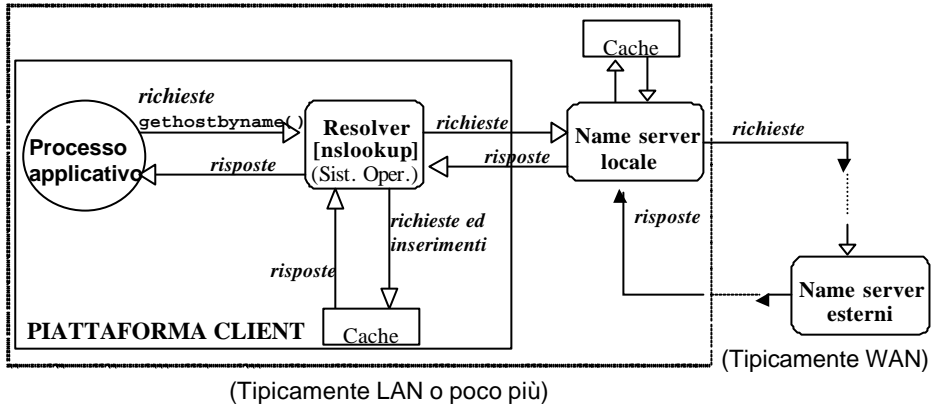
- Attualmente, questo file è gestito dall'ente Network Solutions Incorporated of Herndon, Virginia (USA) ed è reso disponibile ai 12 root name server secondari dal server primario **a.root-servers.net**
- Il controllo sui cambiamenti in questo file è di competenza di IANA. I cambiamenti, tipicamente modifiche dei name server che gestiscono i top level domains, sono effettuati uno, due volte a settimana.
- Il file della “root zone” è trasmesso ai root name servers o in modalità in-band mediante il protocollo DNS (“zone transfer” come descritto in RFC 1034) o in modalità out-of-band mediante il protocollo FTP (come descritto in RFC 952). Data la dimensione relativamente piccola del file della “root zone”, la maggior parte degli aggiornamenti sono propagati mediante “DNS zone transfers”

Componenti del DNS - INDICE

- Domain Name Space e Resource Records
→ Ovvero i dati: zone, descrittori, informazioni
- Name Servers
→ Ovvero i possessori e gestori dell'informazione, con le funzionalità di **server** abilitati a rispondere alle query dei client
- Resolvers
→ Ovvero i (primi) **client** del sistema DNS che sottomettono query per informazioni su hostname e indirizzi IP per conto delle applicazioni di rete



Resolver (lato client)



La maggior parte dei sistemi Linux/Unix hanno il file **/etc/resolv.conf** che contiene il dominio locale e gli indirizzi dei name server per quel dominio

Dati client

- Ogni resolver deve conoscere il riferimento ad almeno un name server locale
- La maggior parte dei sistemi Linux/Unix hanno il file **/etc/resolv.conf** che contiene informazione sulla Zona Locale e gli indirizzi del/i name server per quella Zona

/etc/resolv.conf

```
domain mit.edu
128.113.1.5
128.113.1.3
```

Formato richiesta messaggio DNS

HEADER Section (id., num. richieste, num. risposte,...)

QUERY Section (Domain Name, Type, Class)

***Response* RESOURCE RECORDS**

***Response* AUTHORITY RECORDS**

***Response* ADDITIONAL INFORMATION**

Flag del messaggio (16 bit)

- **QR** - Operation: *Query*=0, *Response*=1
- **QT** - Type: *Standard*=0, *Inverse*=1, *Obsolete*=2,3
- **AA** – Set if Authoritative Answer
- **TC** – Set if Truncated Response (> 512 bytes)
- **RD** – Set if Recursion Desired
- **RA** – Set if Recursion Available
- **Rcode** - Return code: *No_error*=0, *Format_err*=1,
 Server failure=2, *Nome non esistente* =3

Informazioni del RR di risposta

- Domain Name
- Response type
- Classe (IP)
- Time-To-Live (in secondi)
- Dimensione dei dati del RR di risposta
- Dati del Resource Record di risposta

Utilizzo di protocolli TCP e UDP

- Nell'ambito del DNS, si utilizzano sia il protocollo **UDP** sia il protocollo **TCP**:
 - Il protocollo **TCP** per il trasferimento di interi database da server primari a server secondari (replica)
 - Il protocollo **UDP** per il lookup (di singoli nomi)
 - Se nel lookup, la risposta necessita di più di 512 byte (nel caso di interi gruppi di nomi), il richiedente risottomette la richiesta utilizzando il protocollo **TCP**

Libreria di funzioni del DNS

- C'è una libreria di funzioni che agisce come DNS client (**resolver**), per cui non c'è bisogno di scrivere il codice del client per usare il DNS
- Con qualche Sistema Operativo, è necessario esplicitare il link con la libreria del DNS resolver:
 - 1 **ns1** (**ns1** è la "Name Server Library")

gethostbyname : da *hostname* a indirizzo IP

gethostbyaddr : da indirizzo IP a *hostname*

gethostbyname2 (IPv6)

gethostbyname ()

- La funzione **gethostbyname** prende in input una stringa ASCII che contiene l'hostname di un host. Ritorna l'indirizzo di una struttura **hostent** che contiene, tra le altre cose, l'indirizzo IP dell'host in binario:

```
struct hostent *gethostbyname(const char
                               *hostname);
```

La **struct hostent** è definita nel file da includere **netdb.h**:

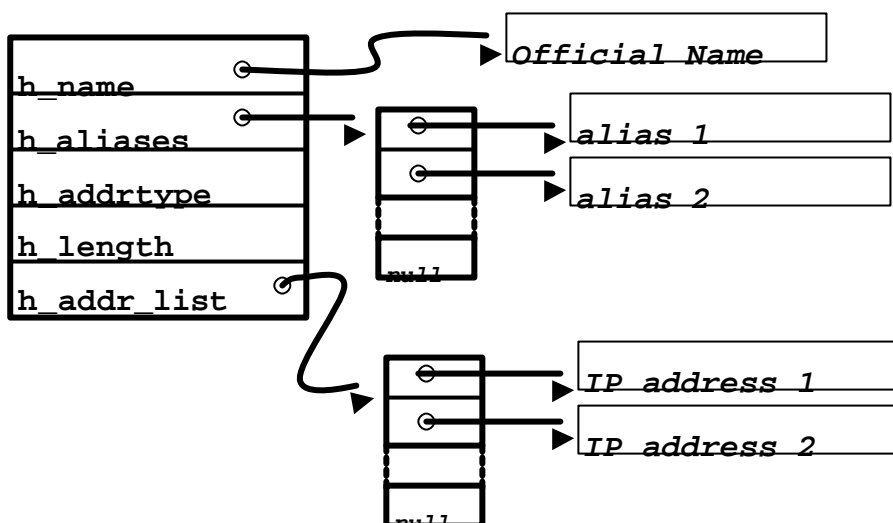
```
#include <netdb.h>
```

struct hostent

```
struct hostent {  
    char *h_name;           nome ufficiale (canonical name)  
    char **h_aliases;       array di altri nomi (aliases)  
    int h_addrtype;         tipo indirizzo: AF_INET o AF_INET6  
    int h_length;           lunghezza dell'indirizzo (4 or 16)  
    char **h_addr_list;     array di puntatori a indirizzi  
};  
  
#define h_addr h_addr_list[0] identificatore contenente la prima  
                                locazione della lista indirizzi
```

I campi che contengono nomi e indirizzi devono essere liste perché host che hanno interfacce multiple, hanno anche nomi e indirizzi multipli

Rappresentazione struct hostent



Risultati della chiamata

Se ha successo, `gethostbyname` restituisce l'indirizzo di un `hostent` che è stato creato in precedenza.

- Contiene un array di puntatori a indirizzi IP
- Tipicamente, viene utilizzato il primo indirizzo IP della lista, contenuto anche nella variabile:

```
#define h_addr h_addr_list[0]
```

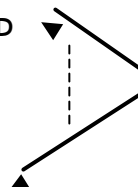
In caso di fallimento, `gethostbyname` restituisce 0 (ovvero null) per cui il client può gestire l'errore verificando il contenuto della variabile `h_errno`

Gestione errori di `gethostbyname`

- In caso di fallimento `gethostbyname` ritorna 0
- `gethostbyname` utilizza la variabile globale `h_errno` per indicare l'errore esatto. Es.,

- `HOST_NOT_FOUND`
- `TRY_AGAIN`
- `NO_RECOVERY`
- `NO_DATA`
- `NO_ADDRESS`

Tutti definiti in `netdb.h`



Esempio

```
struct hostent *hptr;
char *examplename = "weblab.ing.unimo.it";

if (hptr = gethostbyname(examplename))
{ /* l'indirizzo IP cercato si trova in
   hptr->h_addr */ };

else
{ /* si è verificato un errore */
  switch(h_errno)
  {...}
};
```

Abbreviazioni

- I name server rispondono al `gethostbyname()` che ha come parametro il nome completo, detto **Fully Qualified Name**, es. `weblab.ing.unimo.it`
- Comunque, è anche possibile specificare un nome parziale (`weblab.ing`) il cui suffisso (`weblab.ing.unimo.it`) può essere completato dal Resolver
- Ciascun Resolver ha una lista di suffissi da provare

gethostbyaddr

```
struct hostent *gethostbyaddr(  
    const char *addr  
    size_t len, ← sizeof(struct in_addr)  
    int family);
```

↖ AF_INET (could be AF_INET6)

Altre funzioni

uname: per ottenere l'hostname di un host locale

getservbyname: per ottenere il numero di porta relativo ad un certo servizio applicativo di rete.
Per esempio, HTTP -> 80

getservbyaddr: per ottenere il nome di un servizio, dato un numero di porta

Ulteriori informazioni

Informazioni

- **RFC 1034**: concetti del DNS
- **RFC 1035**: implementazione del DNS e specifiche del protocollo

Pratica

- Prendere pratica con il comando `nslookup`
- Dare uno sguardo al codice per server DNS della distribuzione più diffusa **BIND (Berkeley Internet Name Domain)**

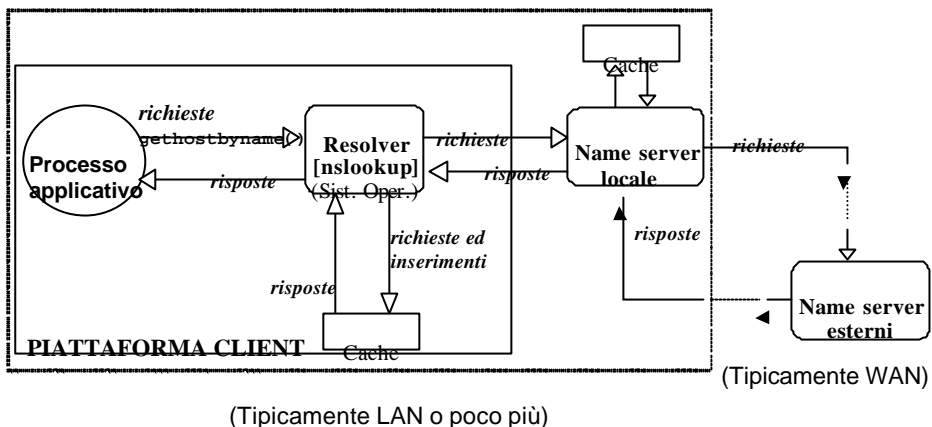
Parte 6

Modulo 4: Meccanismo distribuito di risoluzione dei nomi

Sistema DNS: *Meccanismo di risoluzione*

- Nessun name server ha tutte le corrispondenze tra *hostname* e *indirizzo IP*
- Gli applicativi di rete utilizzano un meccanismo distribuito (client/server) per la risoluzione dei nomi (*Lookup phase*) attivato dalla componente *Resolver* del client

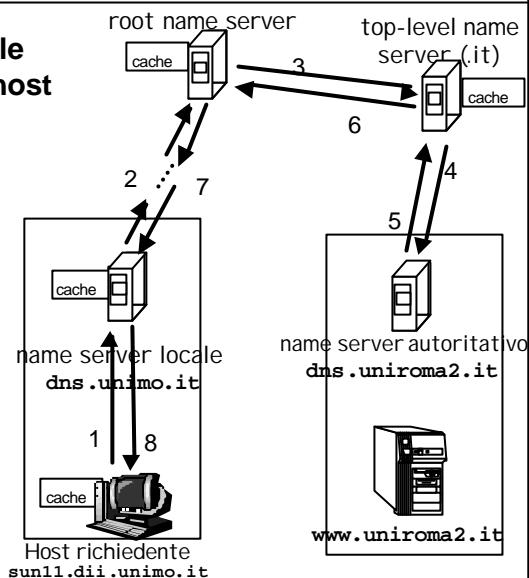
Resolver (*lato client*)



Sistema DNS: *address resolution*

L'host `sun11.dii.unimo.it` vuole conoscere l'indirizzo IP dell'host `www.uniroma2.it`

- 1) Contatta il suo *DNS locale*
- 2) Se necessario, il *DNS locale* può contattare altri DNS intermedi, ed eventualmente uno dei *root DNS*
- 3) Se necessario, il *root DNS* contatta il *DNS autoritativo* per quell'indirizzo (o un *top-level DNS* nel caso in cui non conosca un *DNS autoritativo* per quell'indirizzo)



Tipi di query

Ciascun name server può effettuare due tipi di query nella risoluzione di un nome:

- **Query ricorsiva**
 - Il server, contattato e non in grado di risolvere il nome richiesto, assume un ruolo di client nei confronti di un altro name server.
 - Più costosa in termini computazionali.
- **Query iterativa**
 - Il server, contattato e non in grado di risolvere il nome richiesto, risponde con i nomi di uno o più server da contattare.

I root name server usano sempre la versione iterativa.

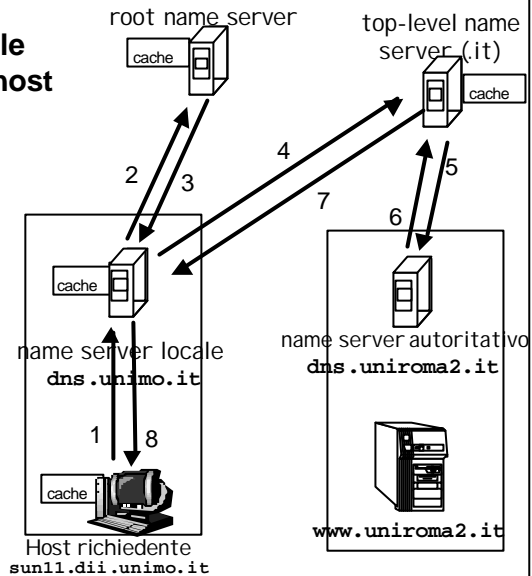
Perché?

Sistema DNS: *address resolution* (2)

L'host `sun11.dii.unimo.it` vuole conoscere l'indirizzo IP dell'host `www.uniroma2.it`.

Poiché i root name server ricevono molte richieste, per limitare il sovraccarico, tipicamente utilizzano la risoluzione iterativa.

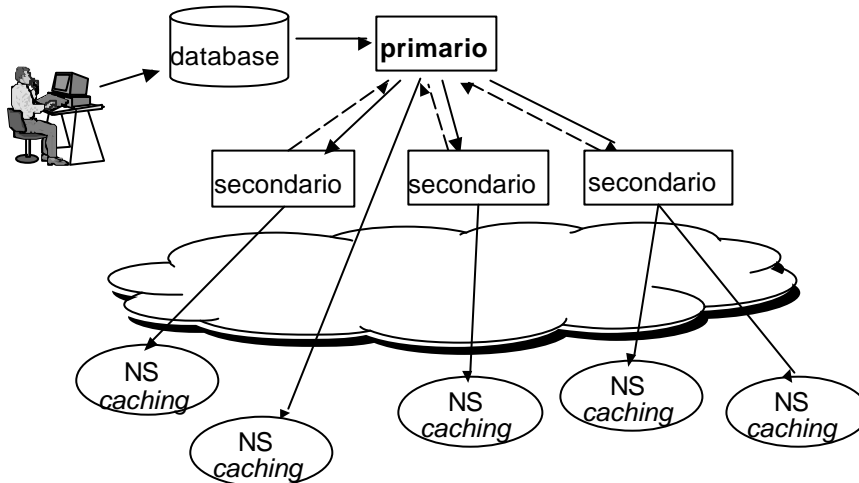
Gli altri name server, tipicamente utilizzano la risoluzione ricorsiva.



Parte 6

Modulo 5: Consistenza ed efficienza del DNS

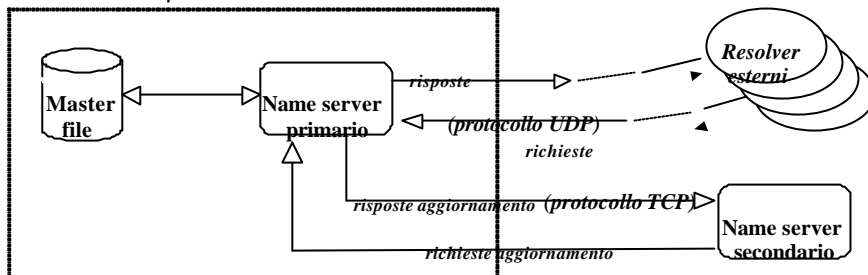
Consistenza dell'informazione distribuita



Attività di aggiornamento e consistenza

I name server che hanno informazioni su di una stessa Zona vengono distinti in **name server primari** (o *master*) e **name server secondari** (o *slave*).

Nel name server primario possono essere effettuate modifiche sui nomi della zona. Il mantenimento della consistenza del database è effettuato dai name server secondari che periodicamente interpellano il server master per controllare se vi sono stati cambiamenti.



Esempio attività aggiornamento

- Gli ISP aggiornano i contenuti dei loro name server locali ogni 12-24 ore
- Gli ISP si collegano ad uno dei server B-M (non A!) e chiedono di effettuare una **“ZONE transfer”**
- Le identità vengono tipicamente garantite da Verisign
- Alcune organizzazioni e università non si affidano ad un ISP, ma hanno propri name server che devono essere continuamente aggiornati (secondo parametri dipendenti dal gestore)
- Se la ROOT ZONE dovesse fallire completamente, Internet non “morirebbe” immediatamente, ma rimarrebbe “ibernata” (impossibilità di aggiungere/modificare domini e zone).
Infatti, l’attuale funzionamento del sistema DNS non tiene conto del TTL quando una nuova copia non risulta disponibile. Tuttavia, una disattivazione dei Root Name Server per più di 60 giorni “ucciderebbe” Internet perché tutti i **record SOA** dei name server perderebbero di validità (si supererebbe il **maximum expiration time** consentito)

Soluzioni per la tolleranza ai guasti

- Un name server può possedere **dati autoritativi** per zero o più Zone
- Per garantire che i dati su hostname e indirizzi IP di una Zona siano disponibili anche quando un name server fallisce, le specifiche dell’architettura del DNS richiedono che ogni Zona debba essere replicata in almeno due server autoritativi che siano tra di loro *failure-independent*

Soluzioni per la scalabilità

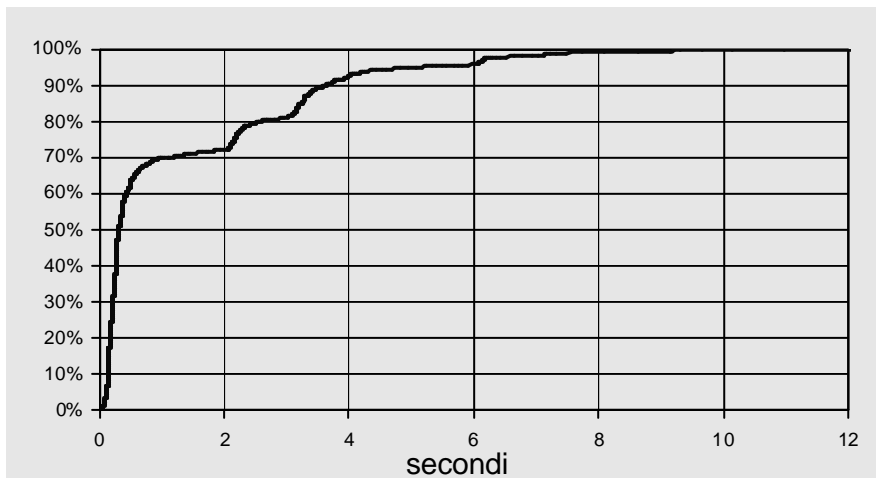
- Autorità e competenze delegate per i differenti Domini
- Database dei dati partizionati in Zone
- Replica dei dati delle Zone su più name server
- Ampio uso di caching delle informazioni

Ricordarsi dei meccanismi di caching

- Tutti i nomi risolti vengono mantenuti nelle cache dei name server -anche intermedi- per un periodo di validità, detto Time-To-Live (TTL)
- Il name server autoritativo, infatti, nel momento in cui risolve un nome, oltre all'indirizzo, ritorna anche il valore del TTL
- Recenti analisi hanno verificato che il valore di TTL più utilizzato è pari ad un giorno (84600 secondi). Comunque, ciascun name server può utilizzare valori differenti

Tempi per la risoluzione mediante DNS

(domini random: 70% entro 2 sec., 90% entro 4 sec.)



I nomi di domini popolari hanno tempi inferiori (*perché?*)