TFA 2015
SISTEMI E RETI DI CALCOLATORI PER L'INSEGNAMENTO
Lezione n.2
PROGETTARE UNA UNITA' DIDATTICA: IL DNS, UN SERVIZIO DI NAMING

07/05/2015

Laura Ricci

STRUTTURA DELLA PRESENTAZIONE

Approccio generale: partire da un problema concreto, la risoluzione di nomi simbolici in Internet, per introdurre diversi concetti importanti

- Strutture ad albero
- Meccanismi di caching
- Iterazione verso Ricorsione

Motivare i ragazzi discutendo la struttura di un'applicazione essenziale per il funzionamento di applicazioni su Internet

Associare alla lezione una esperienza di laboratorio strettamente collegata ai temi mostrati nella lezione teorica

INTERNET: LIVELLO APPLICAZIONE

- Applicazioni tradizionali:
  - Email
  - News
  - Remote Login (SSH)
  - File Transfer
- L'applicazione 'killer':
  - World-Wide Web (WWW)
- Nuove applicazioni:
  - Videoconferenza
  - Telefonia (VOIP)
  - P2P applications

INTERNET: LIVELLO APPLICAZIONE

Applicazione	Protocollo a livello applicazioni	Protocollo a livello trasporto
Posta Elettronica	SMTP	TCP
Login Remota	SSH	TCP
Web	HTTP	TCP
Trasferimento File	FTP	TCP
File Server Remoto	NFS	UDP o TCP
Multimedia	Proprietario	UDP o TCP
Telefonia	Proprietario	UDP
DNS (Domain Name Server)	….	…..

IL DOMAIN NAME SYSTEM (DNS)

- La metafora: il DNS è essenzialmente la rubrica telefonica di Internet. Come la rubrica telefonica mappa Mario Rossi nel suo indirizzo telefonico. Ogni device che comunica in Internet possiede un indirizzo IP (analogia con numero telefonico) che viene mappato n un indirizzo simbolico.

gli indirizzi IP:

- hanno lunghezza fissa e possono essere gestiti in modo semplice dai routers.
- hanno struttura gerarchica in modo da favorire l'instradamento nei routers.
- sono poco adatti per essere memorizzati da utenti umani.

Introdotta la possibiltà di associare nomi simbolici agli hosts della rete, ad esempio

fujim3.cli.di.unipi.it

- nomi simbolici a lunghezza variabile adatti per gli utenti

di.unipi.it?

IL DOMAIN NAME SYSTEM (DNS)

Domain Name System: servizio di rete che consente di tradurre nomi simbolici in indirizzi di risorse di rete

qual'è l'indirizzo IP corrispondente a `fujim3.cli.di.unipi.it?`

qual'è l'indirizzo del mail server che gestisce la posta per il dominio

- a livello logico: una grande tabella che registra la corrispondenza tra nomi simbolici ed inidirizzi IP
- In maggior dettaglio: un data base distribuito che memorizza coppie (nome simbolico – indirizzo IP) su un insieme di nodi della rete (name servers)
- un protocollo a livello applicazione che regola la comunicazione tra hosts e name servers basato sul paradigma client/server. Linguaggio standardizzato per effettuare query ai DNS server
- utilizzato da altri protocolli per la risoluzione dei nomi simbolici
  - esempio HTTP estrae dalla URL digitata dall'utente il nome simbolico dell'host e lo traduce in un indirizzo IP

IL DOMAIN NAME SYSTEM

Funzioni principali del DNS

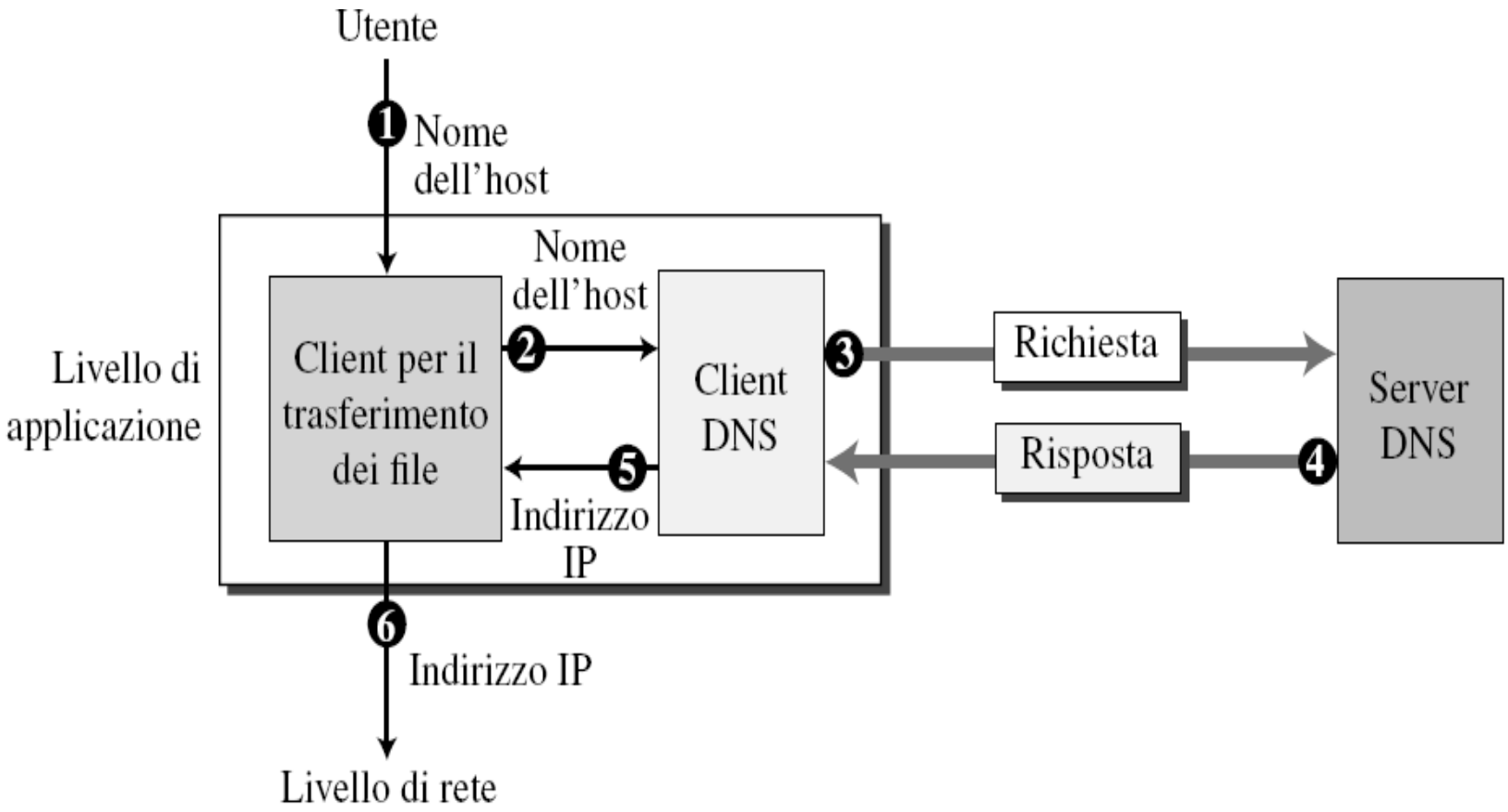
- traduzione nomi indirizzi IP
- gestione degli alias = definire più nomi simbolici per lo stesso host in modo da garantire bilanciamento del carico
  - un web server (es:`www.cnn.com`) destinato a servire una enorme quantità di richieste può essere replicato (più hosts offrono quel servizio)
  - allo stesso nome simbolico viene associato un insieme di indirizzi IP
  - il DNS restituisce gli indirizzi IP associati allo stesso nome simbolico secondo una disciplina circolare

IL DOMAIN NAME SYSTEM

Aspetto motivazionale: far capire l'importanza di un buon servizio di DNS:

- Un DNS che funziona bene è critico per ottenere un buon accesso a Internet
- la banda dell'accesso ad Internet risulta irrilevante se il DNS risulta lento

DNS: ESEMPIO DI UTILIZZO



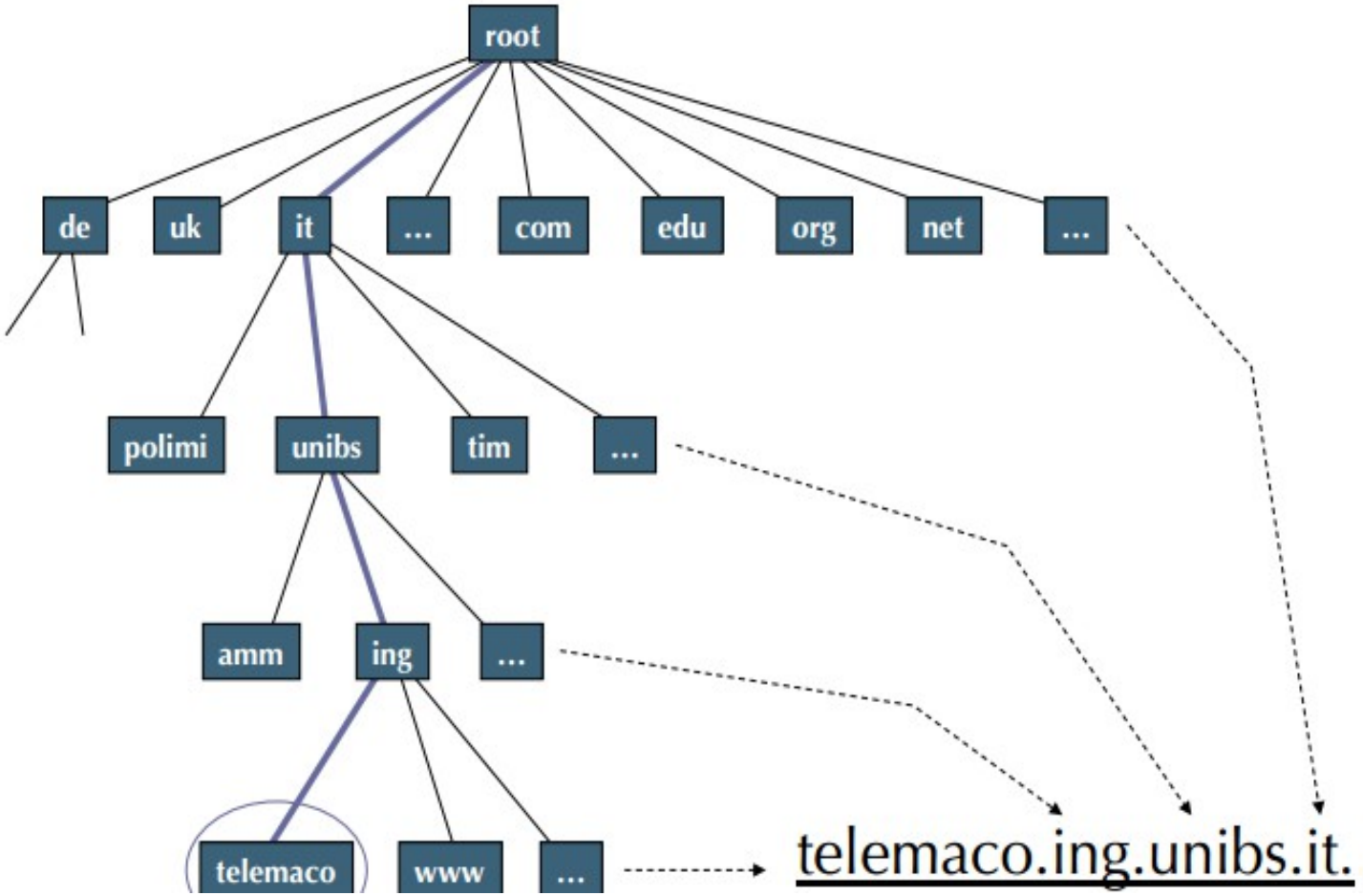
UN PO' DI STORIA: ANNI '80

- la lista degli indirizzi IP degli host interconnessi ad ARPANET veniva mantenuta in un singolo file di testo
- il master file veniva quotidianamente scaricato con FTP da un sito centrale, e poi distribuito a tutti gli host di una data sottorete
- oltre al mappaggio nome/indirizzo IP, il file conteneva anche altre informazioni, come indirizzi di rete e relativi nomi simbolici di sottoreti, indirizzi dei gateway responsabili di una certa sottorete, ecc.
- un singolo server centralizzato pone diversi problemi:
  - namespace piatto: conflitti sui nomi
  - single point of failure
  - volume di traffico eccessivo
  - mantere un database di queste dimensioni risulta problematico
  - scalabilità limitata !!

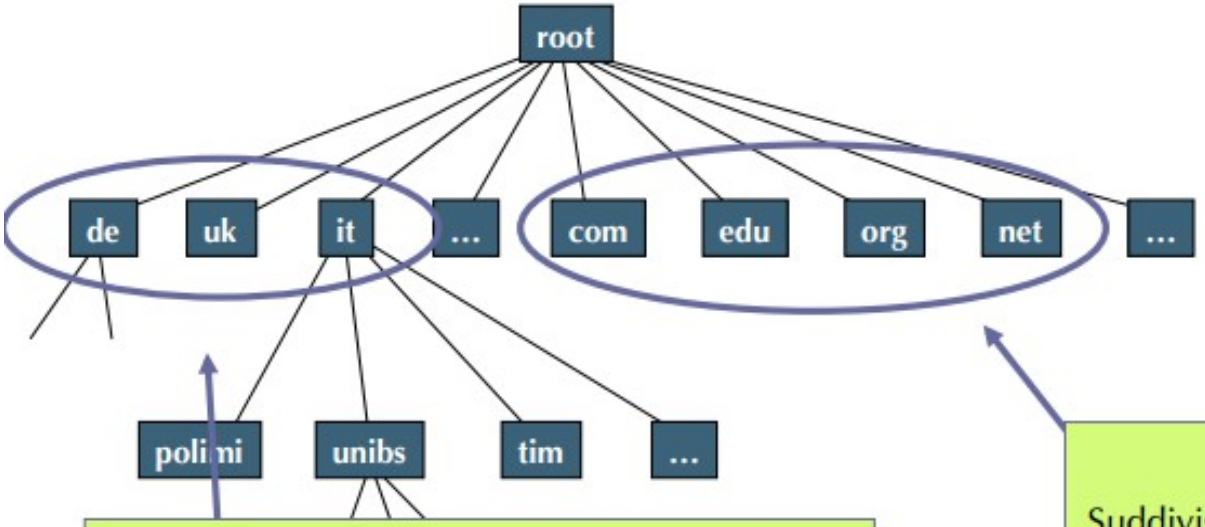
DNS: LA STRUTTURA ATTUALE

- specifiche base in [RFC 1034, RFC 1035]
- sistema facilmente estendibile e scalabile:
  - permette mapping di indirizzi IP end-host e "altri"
- componenti del sistema
  - namespace gerarchico, organizzato ad albero
    - ogni livello dell'albero è gestito da un'entità amministrativa potenzialmente indipendente
    - ogni sotto-albero rappresenta un dominio
  - database che contiene il namespace distribuito su diversi name server
  - possibilità di caching locale
  - protocollo client-server
- namespace gerarchico DNS: parallelo con l'indirizzamento usato in altri campi
  - nella rete telefonica (es: +39-050-221.....)
  - nel sistema postale (Mario Rossi, UniPi, via ....., Pisa, Italy)

DNS: NAMESPACE GERARCHICO



DNS: NOMI DI DOMINIO



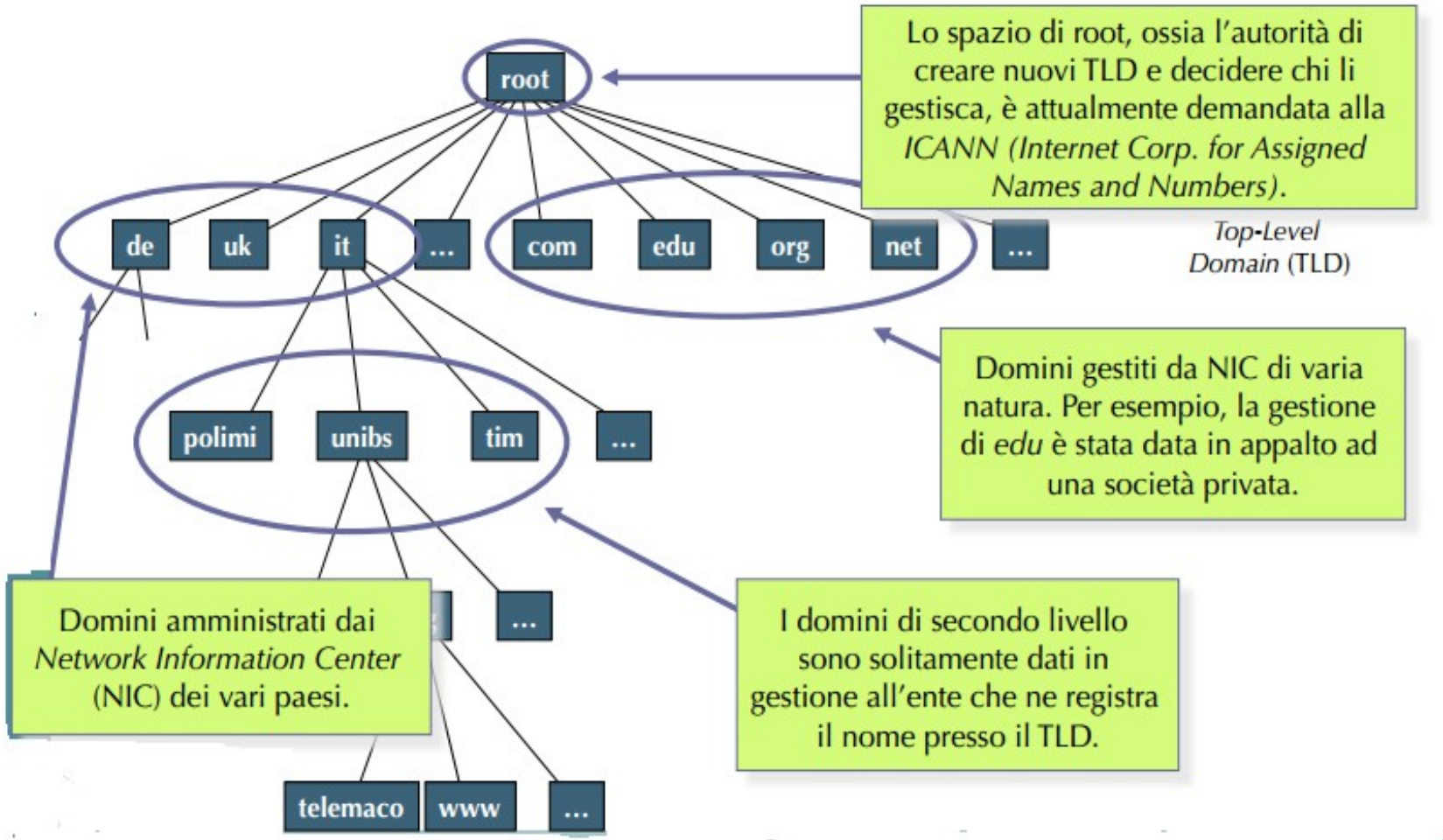
Tutta Internet

Top-Level Domain (TLD)

Suddivisione per **appartenenza geografica**. I domini che fanno capo a questi TLD sono amministrati da enti/società che (*di solito*) geograficamente operano nella specifica area. *NOTA BENE*: questo non significa che gli host appertenenti al dominio risiedano fisicamente in una data area geografica.

Suddivisione per **tipologia di utilizzo** del dominio. Oggi praticamente non più seguita, se non per particolari TLD statunitensi, come *gov, edu, mil*

CHI AMMINISTRA IL NAME SPACE DI UN DOMINIO?



DNS: LA STRUTTURA ATTUALE

- nell'albero gerarchico del namespace DNS, ogni nodo, con tutti i suoi discendenti, corrisponde a un dominio
  - un dominio rappresenta quindi un sotto-albero nell'albero gerarchico di tutti nodi connessi alla rete, includendo tutti i nomi di risorse e nodi di rete che ne fanno parte.

- Esempio
  - il dominio `unipi.it` è un sotto-albero del dominio it, e rappresenta tutti i nomi, le reti e nodi di rete che sono in gestione all'Università di Pisa, il cui nome ha la parte più significativa uguale a `unipi.it`
  - il dominio `it`, a sua volta, è un sotto-albero della root

- casi particolari di dominio
  - root: l'intero albero
  - un singolo nodo di rete (`luna.di.unipi.it`)

DNS: NOMI DI DOMINIO

- composto da label, separate da dot ('.'). Il nome di dominio di un nodo nell'albero gerarchico è l'unione ordinata, separata da dot, dei nomi (label) di tutti i nodi che stanno sul percorso dal nodo in questione alla root
- label
  - composta da lettere, numeri e '-'
  - case-insensitive
  - deve iniziare con una lettera
  - massimo 63 caratteri
- massima lunghezza nome di dominio: 255 caratteri
- nome di dominio assoluto (luna.di.unipi.it)
  - specifica tutte le parti dell'intero percorso, fino alla più significativa
  - ha significato globale e univoco nell'intero albero gerarchico DNS
- nome di dominio relativo
  - ha significato locale, all'interno del dominio di afferenza (luna.di)

DNS: STRUTTURA DEL DATABASE

- I database DNS sono composti da resource record (RR)
- Ciascun dominio (= nodo nell'albero del namespace) è rappresentato nel database da un insieme di RR
- Ogni RR specifica le proprietà di una particolare risorsa del dominio a cui si riferisce. Per esempio:
  - Indirizzo IP di un nome appartenente al dominio
  - Indirizzo IP del Mail Server che è responsabile per il dominio
  - Il database DNS contiene la mappa nomi di dominio ⇔ RR
- Campi del RR:
  - nome del dominio
  - TTL (Time to Live)
  - classe
  - tipo
  - valore

DISTRIBUZIONE DELLO SPAZIO DEI NOMI

- La gerarchia dello spazio dei nomi di dominio serve anche a distribuire le informazioni relative all'associazione fra nomi e indirizzi IP.
- Gerarchia di name server: le informazioni all'interno dei DNS sono memorizzate nei name server.
- A ogni domino della struttura gerarchica corrispondono uno o più name Server
- quindi i name server possono essere organizzati nella struttura gerarchica che corrisponde a quella dei nomi di dominio .

DISTRIBUZIONE DELLO SPAZIO DEI NOMI

- Zone: Un singolo dominio può essere diviso in varie parti chiamate zone. Ogni name server è responsabile solo per una particolare zona.
- Se il name server è responsabile per un intero domino allora la *zona* e il *dominio* fanno riferimento alla stessa cosa.
- quando un name server, invece, divide il dominio in zone che vengono affidate

ad altri name server allora *zona* e *dominio* risultano distinti

- le informazioni riguardanti l'intero domino verranno memorizzate nei

  name server responsabili per i vari sottodomini

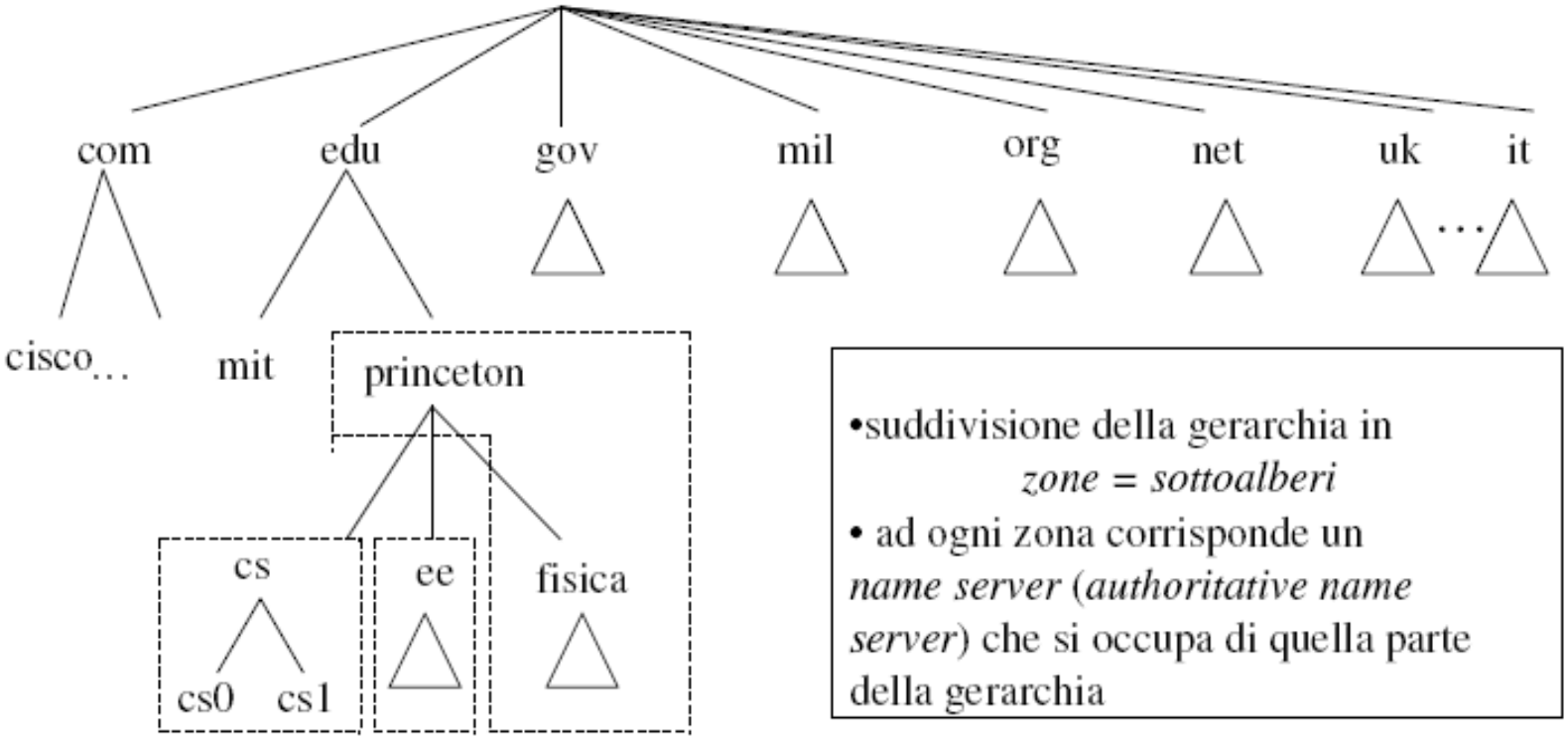
- il server di dominio manterrà solo delle informazioni riguardanti i name

  server dei sottodomini.

- Root name server: è un name server la cui zona consiste nell'intero albero dei

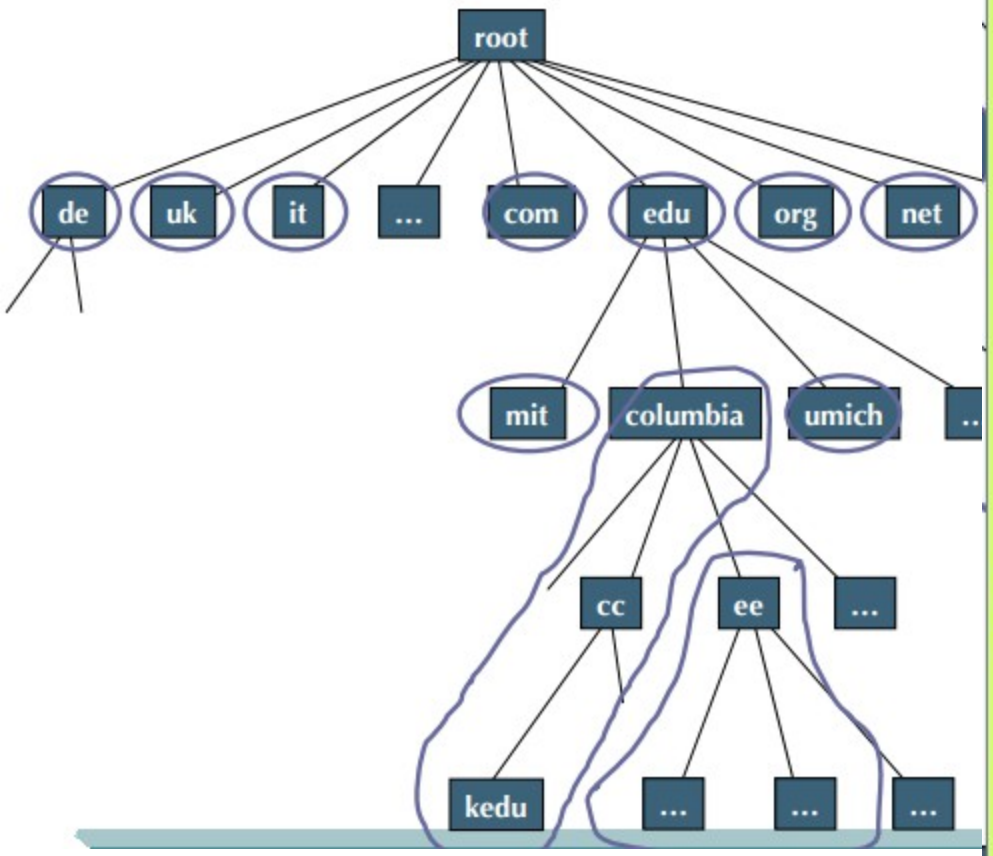
nomi di dominio: per motivi di efficienza e di affidabilità esistono vari root name server. I root name server sono distribuiti un po' dappertutto. Sono funzionalmente equivalenti, basta contattarne uno

DNS: SPAZIO GERARCHICO DEI NOMI



- suddivisione della gerarchia in *zone* = *sottoalberi*
- ad ogni zona corrisponde un *name server* (*authoritative name server*) che si occupa di quella parte della gerarchia

DNS: UN ESEMPIO REALISTICO



Zona: insieme arbitrario di nodi adiacenti nell'albero gerarchico del namespace DNS. Unico requisito: due zone *non possono sovrapporsi*.

Ad ogni zona sono associati due (o più, per backup) ***Name Server***, che gestiscono il database DNS che contiene tutte le RR dei nodi della zona. Uno o più name server di una zona può/possono far parte di un'altra zona, per aggiungere robustezza al sistema. Vi è un NS *primario* più uno o più (ma sempre almeno uno) NS *secondari*.

Authoritative Name Server: è il Name Server che gestisce il database di una zona. Corrispondentemente, i RR contenuti nel database amministrato da quel Name Server sono ***Authoritative RR***. Eccezione: i RR che vengono utilizzati per specificare i NS delle zone "figlie" non sono considerati authoritative.

TIPI DI DNS SERVER

Root Servers.

Ne esistono un numero limitato su tutta la rete (alcune decine). Ciascuno realizzato come un cluster di servers replicati.

Mantengono il database dei RR che contengono gli indirizzi dei NS reponsabili dei TLD

TLD top-level domain server: responsabili dei domini di alto livello, come .com gestiti da enti governativi o privati

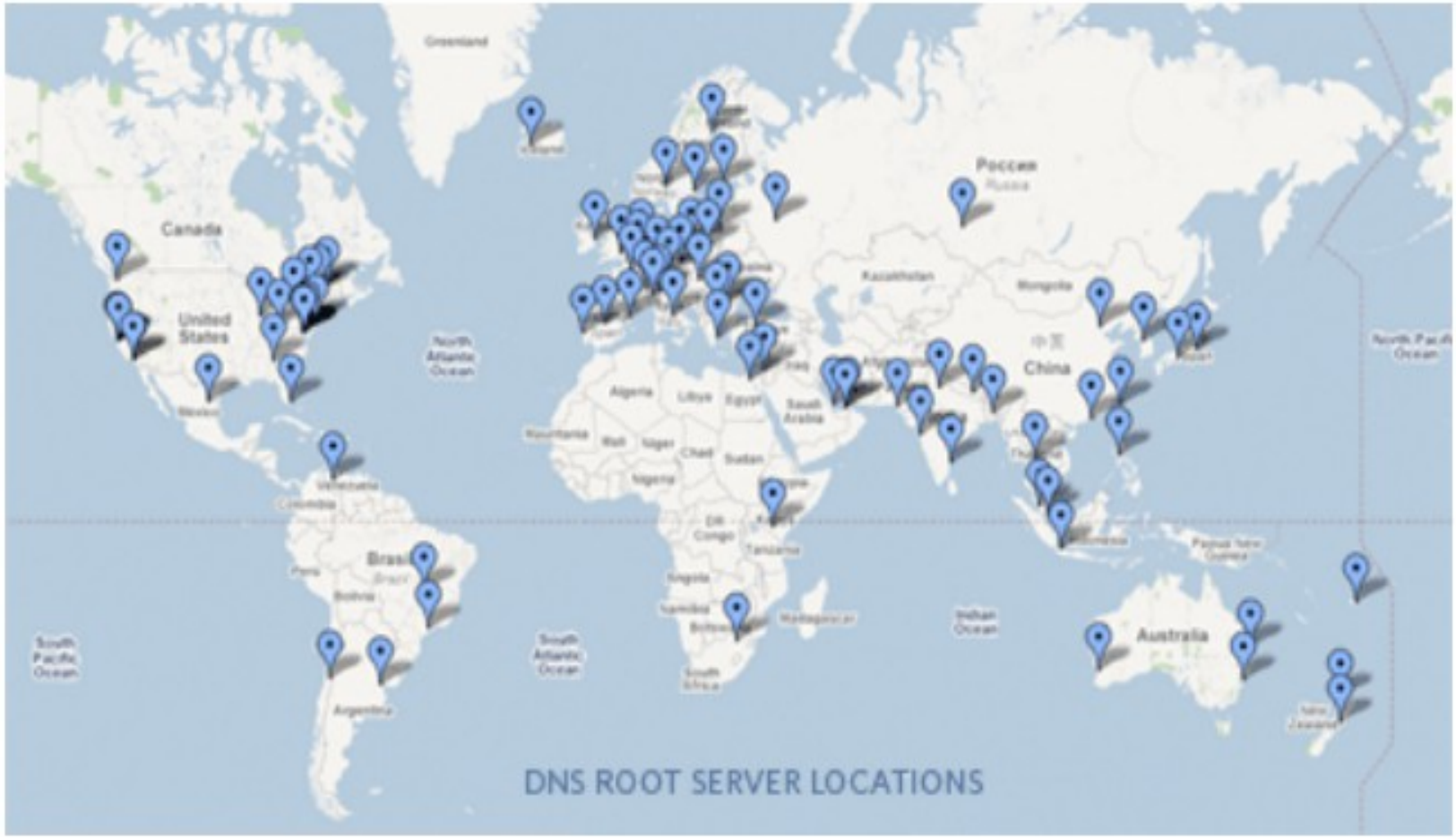
Verisign maintains servers for .com TLD, Educause for .edu TLD

Authoritative Server: Servers associati ad una zona dell'albero dei nomi. Spesso DNS di organizzazioni private

mantenuti dalla organizzazione o dal server privato

devono fornire accesso pubblico per i record RR della organizzazione privata

ROOT NAME SERVER MAP



Oltre 245 servers
Ognuno contiene gli stessi dati

IL DOMAIN NAME SYSTEM

Name Server Locale : Associato ad un ad una organizzazione *O* (università, dipartimento, industria,…).

- contiene le associazioni (nome-indirizzo IP) per tutti gli host di *O*
- la ricerca dell'inizio IP associato ad un nome simbolico inizia sempre dal name server locale.
- l'indirizzo IP del name server locale può essere impostato da ogni host al momento della configurazione dell'host
- se il name server locale non riesce a risolvere il nome, inoltra la richiesta ad un root name server. Quest'ultimo
  - se riesce a risolvere il nome,lo inoltra all'host che lo ha richiesto
  - altrimenti lo invia a un altro name server che possiede il mapping ricercato, oppure conosce l'indirizzo IP di un altro DNS in grado di risolvere il nome

DNS: STRUTTURA DEL DATABASE

- DNS locale: non "strettamente" appartenente alla gerarchia
  - ogni ISP (Internet Service Provider) residential ISP, company, university) ne possiede uno

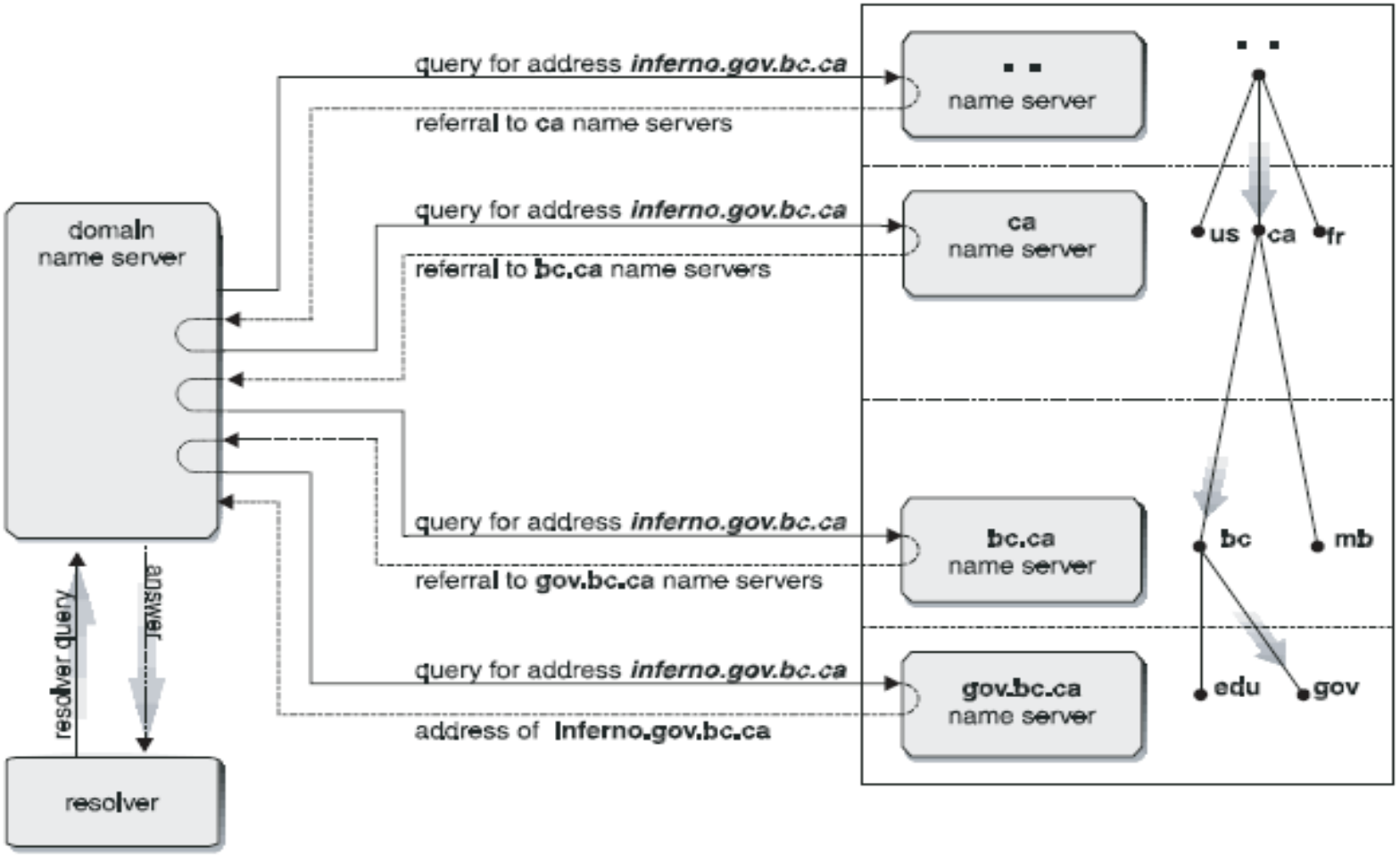
- Quando un host effettua una query al DNS, la query viene inviata al suo DNS locale

- Possiede una cache locale delle traduzioni recenti nome-indirizzo (che però può essere obsoleta)

- Funziona come un proxy forwarda la query nell gerarchia

- Servizio DNS in esecuzione sulla porta 53, usa UDP.

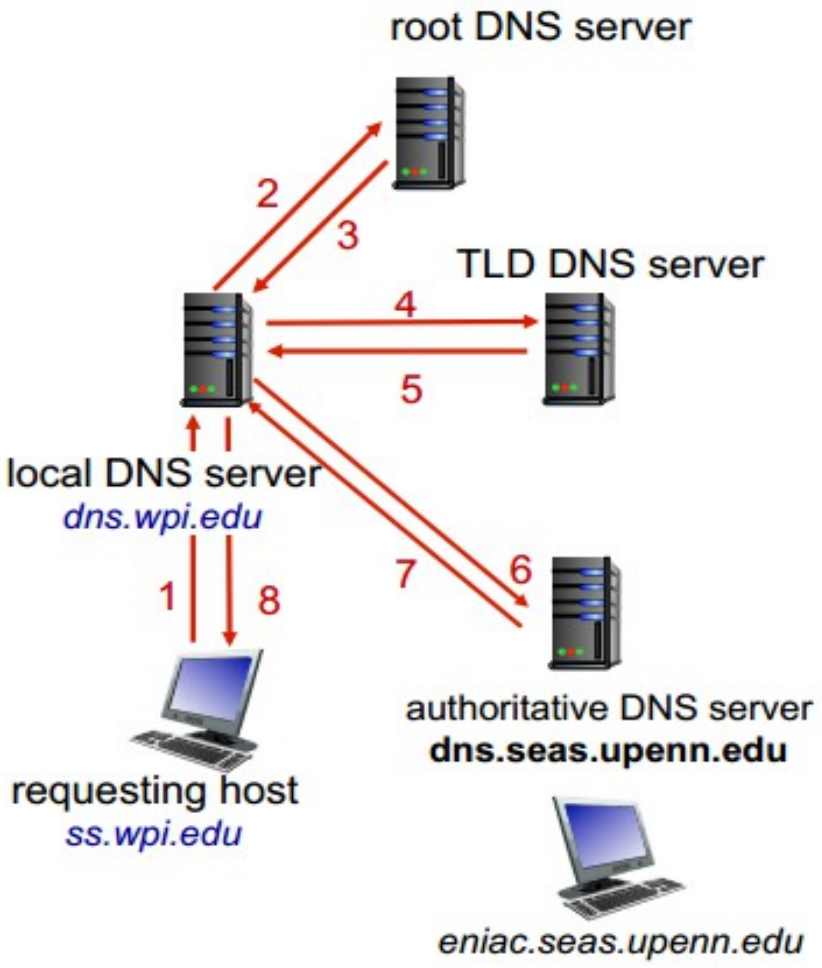
DNS: RISOLUZIONE ITERATIVA



DNS: RISOLUZIONE ITERATIVA DI NOMI

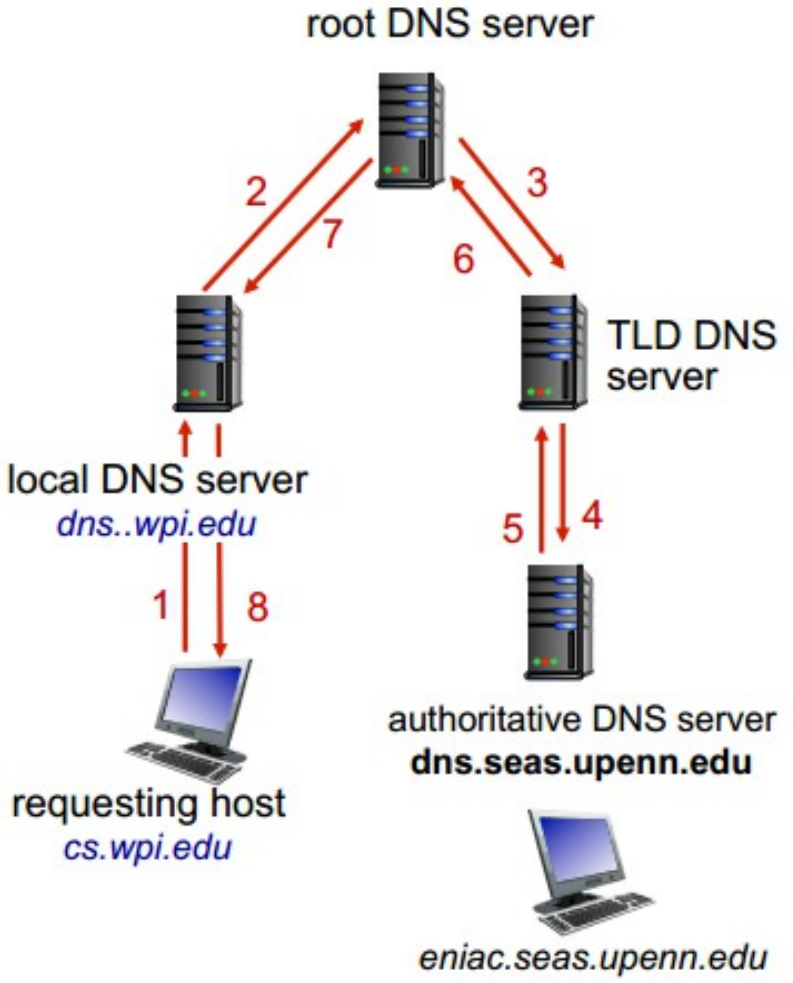
- L'host `ss.wpi.edu` richide l'indirizzo IP address per `eniac.seas.upenn.edu`

- Risoluzione iterativa della query:
  - ogni server contattato risponde con il nome di un altro server da contattare
  - "non conosco questo nome, ma conosco un altro server che deve essere contattato"



DNS: RISOLUZIONE RICORSIVA DI NOMI

- Un altro spunto: processi iterativi verso processi ricorsivi
- L'host `ss.wpi.edu` richide l'indirizzo IP address per `eniac.seas.upenn.edu`
- Risoluzione ricorsiva della query:
  - Carico computazionale per la risoluzione sui name server contattati
  - Alto carico ai livelli più alti della gerarchia



DNS: TECNICHE DI CACHING

- quando un name server riceve un mapping nome simbolico-indirizzo IP,lo memorizza nella sua memoria locale (cache)
- se successivamente riceve la richiesta per lo stesso nome simbolico, può risolvere il nome localmente, senza propagare la query
- Esempio: `fujim3.cli.di.unipi.it` richiede a `dns.di.unipi.it` di risolvere il nome simbolico `google.it`. Dns.di.unipi.it risolve il nome in indirizzo IP e memorizza il mapping nella propria cache. E' molto probabile che un altro host del dominio `cli.di.unipi.it` richieda entro un breve lasso di tempo la risoluzione dello stesso nome simbolico
- un mapping viene scartato dopo un certo intervallo di tempo (tipicamente due giorni)
- i meccanismi di caching sono introdotti per migliorare il tempo di risposta ad una richiesta di risluzione di un nome simbolico e per diminuire il numero di messaggi spediti sulla rete

CACHING: CONCETTI GENERALI

- L'introduzione del DNS può rappresentare un valido spunto per introdurre un altro concetto molto importante in informatica, quello dei meccansimi di caching

- Si può richiedere come prerequisito della lezione, oppure approfittare dello spunto per introdurlo

- Da wikipedia: "Con cache si indica una area di memoria temporanea dove risiede un insieme di dati che possano essere velocemente recuperati quando necessari"

- Una cache è associata ad una memoria "di riferiemto", in cui risiedono i dati. La memoria"di riferimento" può essere:
  - una RAM (memoria principale di un processore)
  - un disco rigido
  - un database distribuito come il DNS.

CACHING: CONCETTI GENERALI

- è tipicamente di capienza inferiore rispetto alla memoria principale, ma il suo utilizzo è più conveniente in termini di tempo di accesso e/o carico sul sistema.
  - tempo di accesso: mediante l'illustrazione del DNS questo concetto viene illustrato facilmente. Richiede meno tempo l'accesso ad una copia locale dei dati, piuttosto che accedere ad un server collocato lontano geograficamente....
  - carico sul sistema: è meglio distribuire gli accessi su tanti server distribuiti, piuttosto che su un unico server centralizzato che diventa un collo di bottiglia

- principio di funzionamento:
  - quando è necessario l'accesso ad un dato, viene prima cercato nella cache
  - se è presente e valido, viene utilizzato
  - altrimenti si recupera il datodalla memoria di riferimento, e si memorizza nella cache, nel caso possa servire successivamente.

UNA ESPERIENZA DI LABORATORIO

Obiettivi

- mostrare come operano le tecniche di caching
- poichè è complesso mostrare le tecniche di caching a livello di server DNS, si possono mostrare a livello della JAVA Virtual Machine
- evidenziare che esistono diversi livelli di caching
- utilizzo dell'API JAVA

UNA ESPERIENZA DI LABORATORIO

- l'accesso al DNS è una operazione potenzialmente molto costosa
- i metodi descritti effettuano caching dei nomi/indirizzi risolti.
  - nella cache vengono memorizzati anche i tentativi di risoluzione non andati a buon fine (di default: per un certo numero di secondi)
- se creo un **`InetAddress`** per lo stesso host, il nome viene risolto con i dati nella cache (di default: per sempre)
- permanenza dati nella cache: per default alcuni secondi se la risoluzione non ha avuto successo, tempo illimitato altrimenti. Problemi: indirizzi dinamici.....
- `java.security.Security.setProperty` consente di impostare il numero di secondi in cui una entrata nella cache rimane valida, tramite le proprietà

  `networkaddress.cache.ttl`

  `networkaddress.cache.negative.ttl`

```
java.security.Security.setProperty("networkaddress.cache.ttl
  ",”0”);
```

UNA ESPERIENZA DI LABORATORIO

```
import java.net.InetAddress; import java.net.UnknownHostException; import
java.security.*;
public class Caching
   public static final String CACHINGTIME="0";
  {public static void main(String [] args) throws InterruptedException
      { Security.setProperty("networkaddress.cache.ttl",CACHINGTIME);
        long time1 = System.currentTimeMillis();
        for (int i=0; i<1000; i++)
          {try { System.out.println(
                  InetAddress.getByName("www.cnn.com").getHostAddress());}
           catch (UnknownHostException uhe)
              { System.out.println("UHE");} }
        long time2 = System.currentTimeMillis();
        long diff=time2-time1;
        System.out.println("tempo trascorso e'"+diff);}}
```

UNA ESPERIENZA DI LABORATORIO

- Output prodotto dal programma:
  - **CACHINGTIME=0**

    tempo trascorso è 545
  - **CACHINGTIME=1000**

    tempo trascorso è 85
- Chiaramente visibili gli effetti del caching sulle prestazioni del programma

IMPLEMENTARE L' UTILITY NSLOOKUP

- implementazione in JAVA dell'utility UNIX **`nslookup`**
- **`nslookup`**
  - consente di tradurre nomi di hosts in indirizzi IP e viceversa
  - i valori da tradurre possono essere forniti in modo interattivo oppure da linea di comando
  - si entra in modalità interattiva se non si forniscono parametri da linea di comando
  - consente anche funzioni più complesse (vedere LINUX)

IMPLEMENTARE L'UTILITY NSLOOKUP

```
import java.net.*;
import java.io.*;

public class HostLookUp {
   public static void main (String [ ] args} {
    if (args.length > 0) {
        for (int i=0; i<args.length; i++) {
              System.out.println (lookup(args[i])) ;
                                                  }
        }
    else {/* modalita' interattiva*/….. }
```

IMPLEMENTARE L'UTILITY NSLOOKUP

```
private static boolean isHostName (String host)

    {char[ ] ca = host.toCharArray();
    for (int i = 0; i<ca.length; i++)
        { if(!Character.isDigit(ca[i])) {
            if (ca[i] != '.') return true; }
        }
    return false;
    }
```

IMPLEMENTARE L'UTILITY NSLOOKUP

```
public class nslookup {
    private static String lookup(String host) {
        InetAddress node;
        try { node =InetAddress.getByName(host);
                System.out.println(node);
                if  (isHostName(host))
                      {return node.getHostAddress( );}
                 else{
                       return node.getHostName ( );}
               }
        catch (UnknownHostException e)
                          {return "non ho trovato l'host";}
                                }
```