DIPARTIMENTO
DI INFORMATICA
Via Salaria 113 - 00198, Roma

SAPIENZA
UNIVERSITÀ DI ROMA

Reti di Elaboratori

Corso di Laurea in Informatica
Università degli Studi di Roma "La Sapienza"
Canale A-L
Prof.ssa Chiara Petrioli

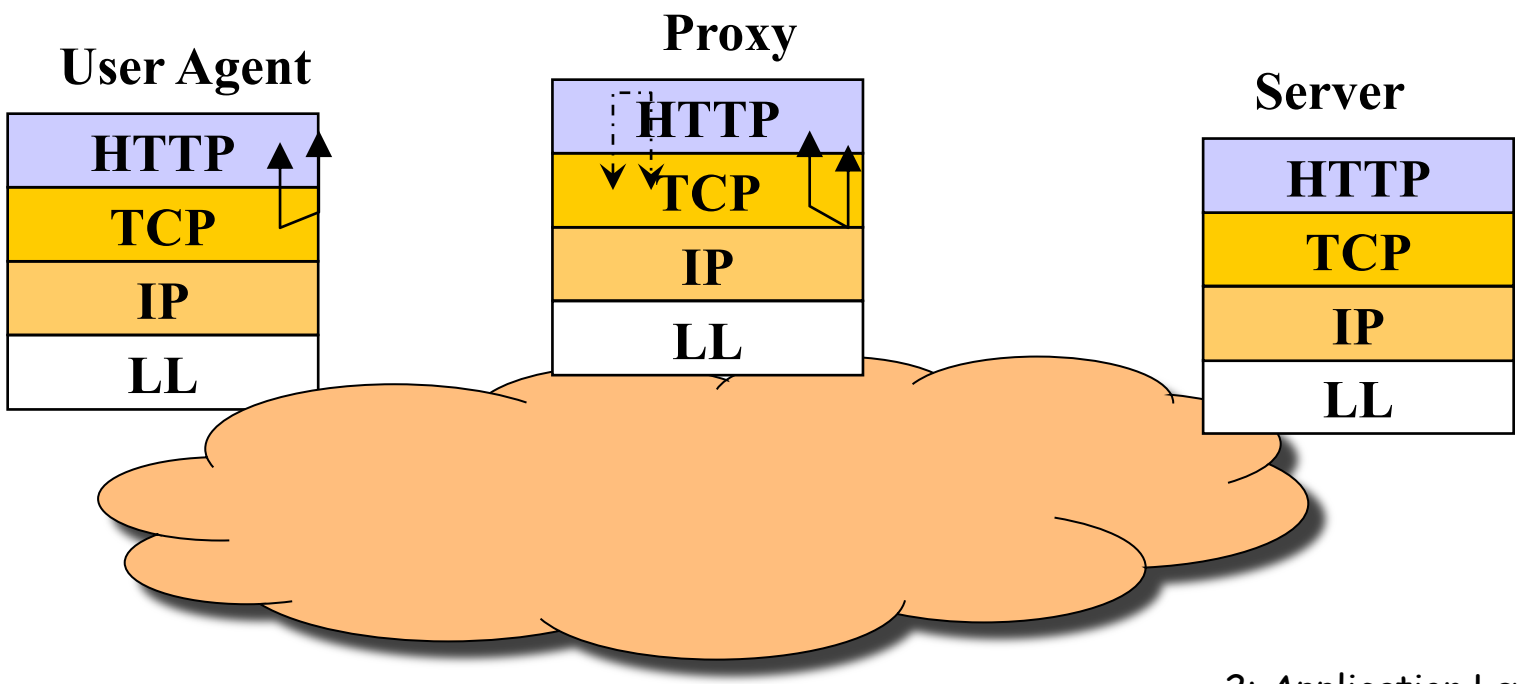
Parte di queste slide sono state prese dal materiale associato al libro *Computer Networking: A Top Down Approach* , 7th edition.

Thanks also to Antonio Capone, Politecnico di Milano, Giuseppe Bianchi and Francesco LoPresti, Un. di Roma Tor Vergata

Proxy

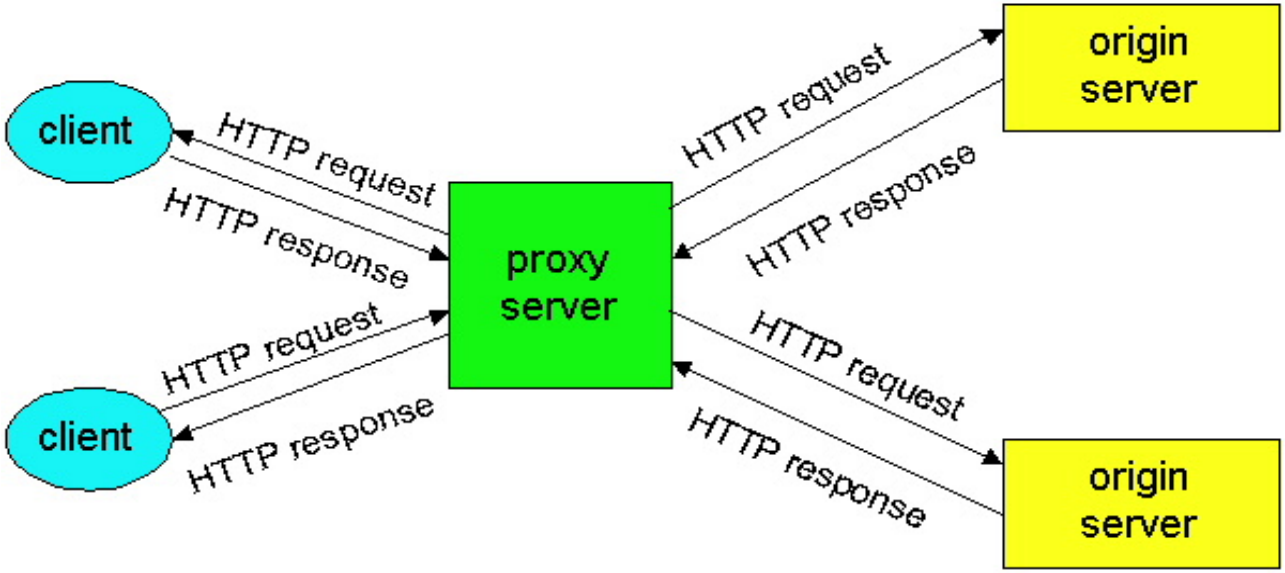
An intermediary program which acts as both a server and a client for the purpose of making requests on behalf of other clients.

- I proxy sono degli instradatori di messaggi di livello applicativo
- Devono essere sia client che server
- Il server vede arrivare tutte le richieste dal proxy (mascheramento degli utenti del proxy)



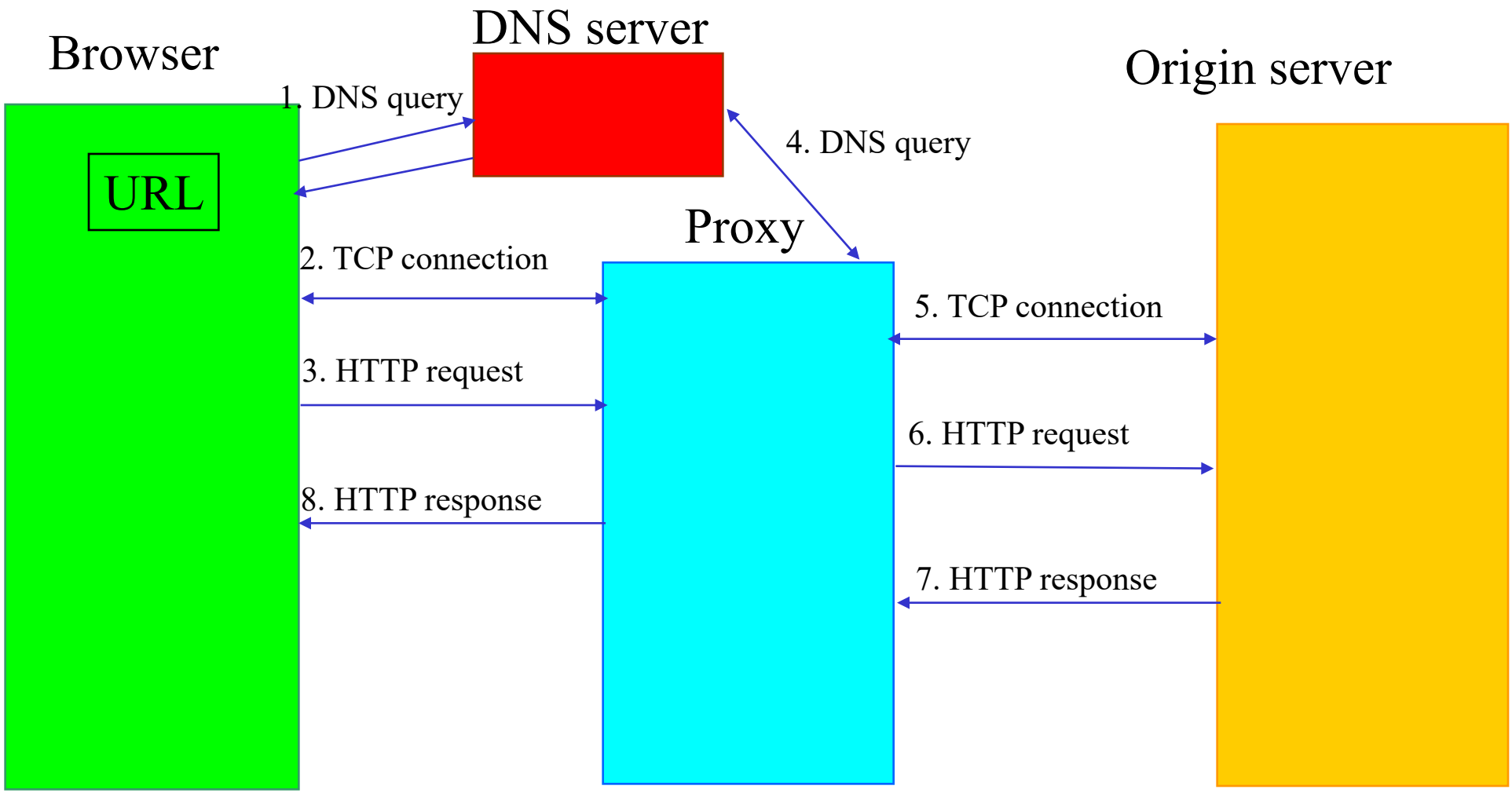
Cache di rete: uso dei proxy

- Compito principale dei proxy è fornire una grande memoria di cache
- Se un documento è contenuto nella cache viene scaricato più velocemente sul client
- Razionale: i server forniscono le stesse info o info fortemente ridondanti a comunita' locali di utenti



- Consistency:cached pages might not be the most recent...
- Ad clickthrough counts:how does Yahoo know how many times you accessed their pages, or ***more importantly***, their ads?

Download atraverso un proxy…



Tipi di proxy

- Regular proxies
  - forwards requests and responses
- Caching proxy
  - has a cache of responses received in the past
  - when the proxy receives a request that can be satisfied by the cache the request message is not forwarded and the response is returned directly by the proxy
- Forwarding behavior
  - Transparent proxy
    - Only superficially changes the request (e.g., adding its ID)
  - Non transparent proxy
    - Can modify request/response
      - Anonymization
      - Media type transformation (e.g. changing image format to reduce size)
      - Language tranformation …
  - Entrambi i transparent/non-transparent proxy possono avere associata una cache

Uso dei Proxy(1)

- Caching: '*storage of a respose obtained earlier for later use, when clients request the same resource'*. Se la pagina richiesta e' nella cache il proxy puo' inviarla immediatamente al client purche' sia fresh (ovvero non sia stata modificata rispetto alla copia in cache)
  - -latency minore MA cache consistency importante. Strong consistency/weak consistency

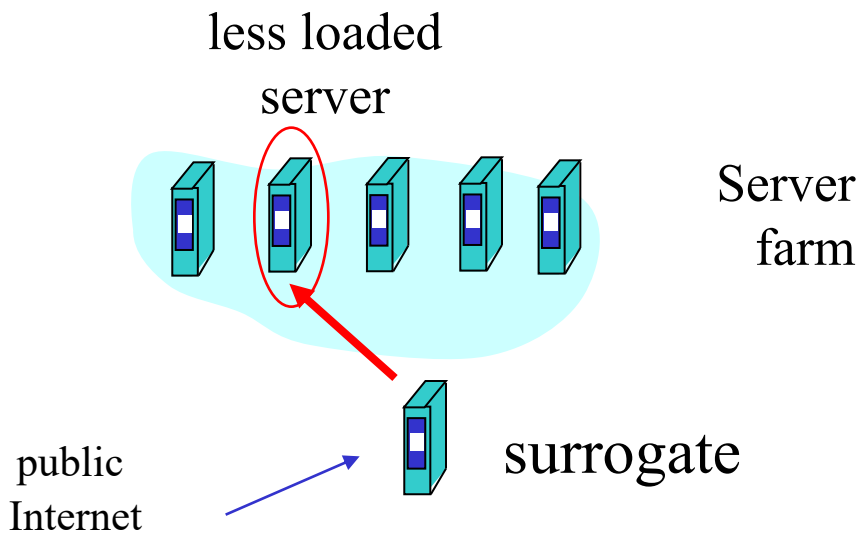
- Agisce come front-end per un gruppo di utenti che condividono l'accesso al web (caching ma anche suddivisione delle risorse— pro: se piu' utenti richiedono la stessa risorsa ad un server un'unica connessione deve essere stabilita; -- con: possibili ritardi dovuti al fatto che piu' utenti condividono le stesse risorse)

Uso dei Proxy(2)

- Mascheramento degli utenti del proxy: Il server vede arrivare tutte le richieste dal proxy e quindi non e' in grado di riconoscere l'utente. Il proxy conosce le informazioni di accesso relative agli utenti ma e' considerata un'entita' trusted.
  - Alcuni proxy non permettono l'anonimizzazione e indicano nell'header informazioni sul client
  - Mascheramento non completo: e.g., user agent e cookies
- Customizzazione delle risposte
- Filtraggio delle richieste e delle risposte
  - per limitare l'accesso a certi siti
  - filtrare richieste a siti non adatti ai minori
  - verificare la presenza di virus prima di inviare al client le risposte
  - togliere alcune informazioni dall'header prima di inviare la richiesta (es. e-mail dell'utente)

Other types of proxies

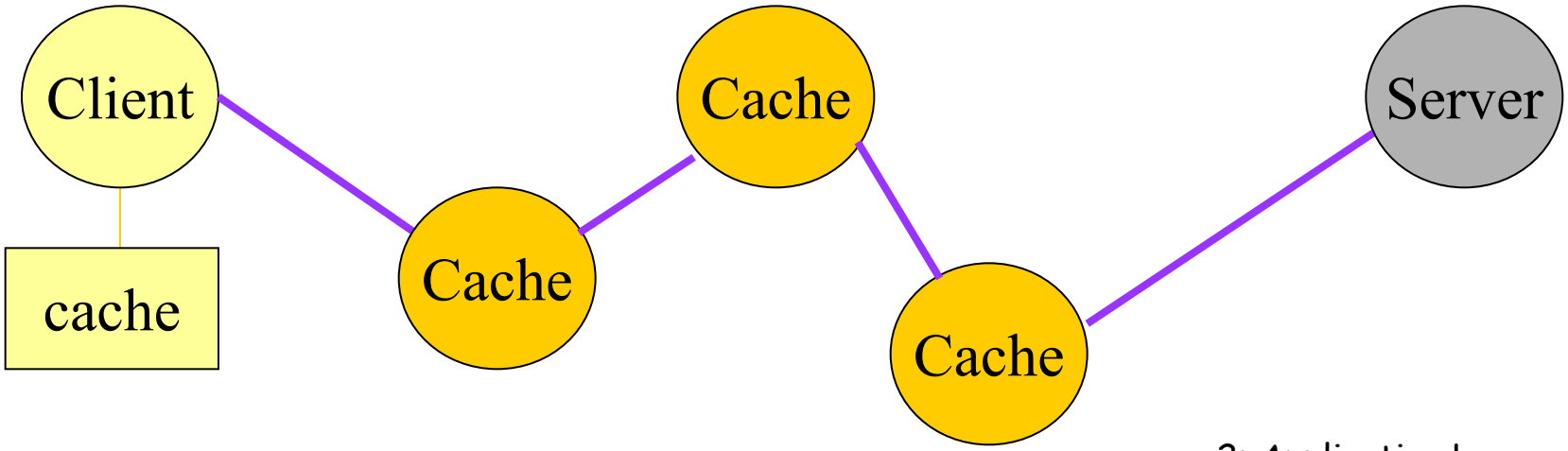
- Reverse proxy (surrogates)
  - Proxy placed close to the origin server to reduce the load on them. Load balancing among servers of a server farm



- Interception proxy. Are invisible to the user. All messages pass through the interception proxy that can intercept them, examine the request, generate a response locally or redirect the request to another place (e.g. to the 'best replica')

Hierarchical Caching

- The message being transported can cross several caches (e.g. at the department, ISP side, in front of the origin server –reverse proxies)
- Decreases communication costs
- Regional/national cache→ high hit ratio



Autenticazione

- HTTP è stateless e quindi non si possono riconoscere richieste successive dello stesso utente
- in HTTP esiste un elementare meccanismo di autenticazione (account e password) che serve a riconoscere gli utenti
- Normalmente il browser memorizza passwd e account in modo da non richiedere la digitazione ogni volta

```
GET /ntw/index.html HTTP/1.1
```

```
401 Authorization Required
WWW-Authenticate:[tipo di autenticazione]
```

```
GET /ntw/index.html HTTP/1.1
Authorization: account, passwd
```

⋮

```
GET image.gif HTTP/1.1
Authorization: account, passwd
```

Cookie

- Esiste anche un altro modo per riconoscere richieste successive di uno stesso utente che non richiede di ricordare password
- Il numero di cookie inviato dal server viene memorizzato in un opportuno file
- Tramite i cookie si può mantenere uno stato "virtuale" per ciascun utente.

```
GET /ntw/index.html HTTP/1.1
```

```
200 OK
Set-cookie:18988466
```

```
GET /ntw/carrello/index.html HTTP/1.1
Cookie: 18988466
```

⋮

```
GET image.gif HTTP/1.1
Cookie: 18988466
```

Cookie

- Permette di identificare gli utenti e mantenere delle info di stato su una transazione che richiede vari scambi di messaggi HTTP
  - Personalizzazione
    - un sito di e-commerce può personalizzare la risposta con il nome dell'utente e suggerimenti di acquisto personalizzati
  - Mantenere informazioni tra sessioni o interazioni
    - esempio: shopping cart

Cookies: keeping "state" (cont.)

Cookie= string of characters

client | server

Cookie file
ebay: 8734

usual http request msg

usual http response +
`Set-cookie: 1678`

server creates ID 1678 for user

entry in backend database

Cookie file
amazon: 1678
ebay: 8734

usual http request msg
`cookie: 1678`

usual http response msg

cookie-specific action

access

access

one week later:

Cookie file
amazon: 1678
ebay: 8734

usual http request msg
`cookie: 1678`

usual http response msg

cookie-spectific action

Identifier to find user info on the back end database

Cookie

- User control on cookies
  - Whether to accept any cookies at all
  - Set a limit on the size and number of cookies
  - Limit the sites/domains from which a cookie can be accepted
  - Limit to a specific session acceptance of cookies
  - Require that cookies must originate from the same server as the current page being viewed
- Privacy concerns
  - Allow to track user behavior
  - Users typically not aware of when a cookie is sent
    - and on the use of the information he/she is providing through use of cookies
      - user profiling, shared between companies
  - Could also be sent from a Web Server different from the one user is aware to being connected to
    - e.g. due to redirection to download some objects referred to in the Main Server web page

Breve storia del Web

- 1945: Vannevar Bush in 'As we may think' proposes Memex to extend human memory via mechanical means. V. Bush wanted to make the existing store of knowledge (fastly growing) more accessible to mankind. 'A memex is a device in which an individual stores all his books, records, and communications, and which is mechanized so that it maybe consulted with exceeding speed and flexibility. It is an enlarged supplement to his memory. …Wholly new forms of encyclopedias will appear ready made with a mesh of **associative trails** running through them'
- 1965 Ted Nelson coins the term 'hypertext' to describe nonsequential writing that presents information as a collection of linked nodes. 'Readers can pursue the information in a variety of ways by navigating from one node to another'
- 1970s scientific community communicate through ARPANET (exchanging e-mails, file transfers..)
- 1989 Tim Berners-Lee creates the World Wide Web (networked application that links users and services distributed across computers around the world. Users can view pages, search for information, send and receive e-mails, initiate business transactions etc. Tim Berners Lee proposed linking info presented on various machines at CERN (European Laboratory for Particle Physics, Geneva). Other systems had been created to download files (FTP)from remote computers, search for information (Gopher), WAIS (Wide Area Information Servers) etc. Why did the Web succed? 1)SIMPLE INTERFACE 2) ENHANCED FEATURES 3) USE OF HYPERTEXT

Breve storia del Web

- 1991 First browser and server introduced
- 1993 The Web consisted of around 50 servers
- 1993 First release of Mosaic browser for windows (written by Marc Andreesen and Eric Bina). The web accounted for 1% of the traffic of the Internet
- Late 1990s → Web responsible for over 75% of the Internet traffic!! Hundreds millions of users; millions of web sites. Reasons for success: graphical user interface, ease of publishing new content
- Web used for e-business, business to business, interactions between users (chatrooms and games). Web increasingly used also to access private or proprietary data.
  - Web traffic: small number of Web pages accounts for the majority of web accesses
  - from text to images, animations, video files (increased storage needed at the Web servers, increased time for downloading, increased load on the network)
  - Dynamically generated content (e.g. search engines generates pointers based on keywords)
  - Web used for banking, e-business;important personal info (e.g. credit card info transferred)→ security, authentication needed

DNS: domain name system

people: many identifiers:

- SSN, name, passport #

Internet hosts, routers:

- IP address (32 bit in IPv4) - used for addressing datagrams
- "name", e.g., www.yahoo.com - used by humans

Q: how to map between IP address and name, and vice versa ?

Domain Name System:

- *distributed database* implemented in hierarchy of many *name servers*
- *application-layer protocol:* hosts, name servers communicate to *resolve* names (address/name translation)
  - note: core Internet function, implemented as application-layer protocol
  - complexity at network's "edge"

DNS: services, structure

DNS services

- hostname to IP address translation
- host aliasing
  - canonical, alias names
- mail server aliasing
- load distribution
  - replicated Web servers: many IP addresses correspond to one name

why not centralize DNS?

DNS: services, structure

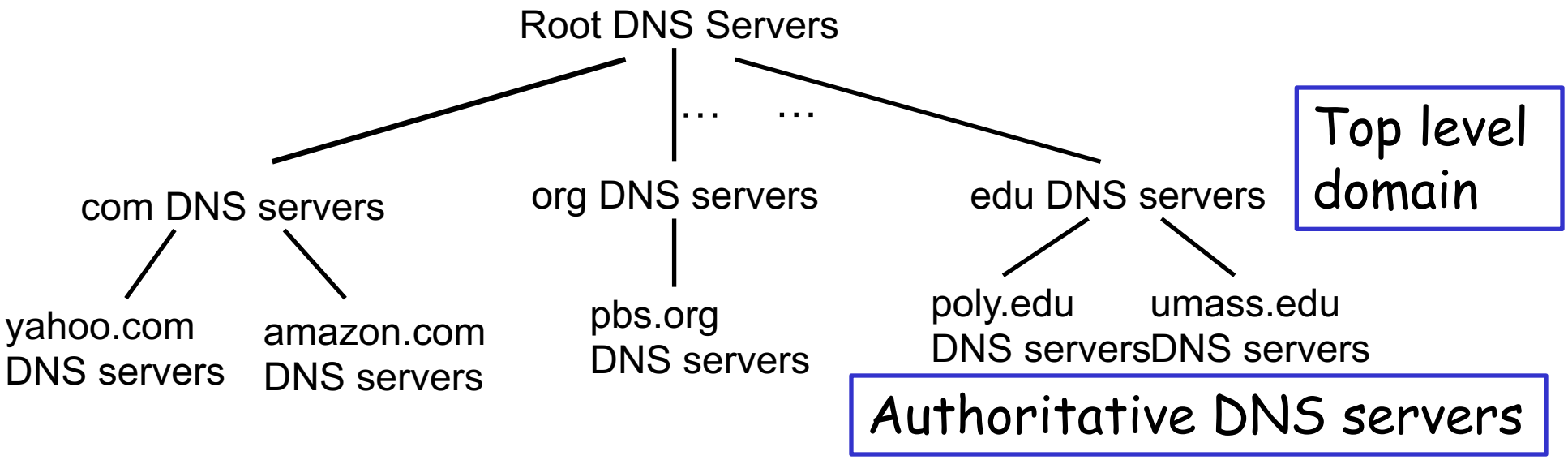
DNS services

- hostname to IP address translation
- host aliasing
  - canonical, alias names
- mail server aliasing
- load distribution
  - replicated Web servers: many IP addresses correspond to one name

why not centralize DNS?

- single point of failure
- traffic volume
- distant centralized database
- maintenance

DNS: a distributed, hierarchical database

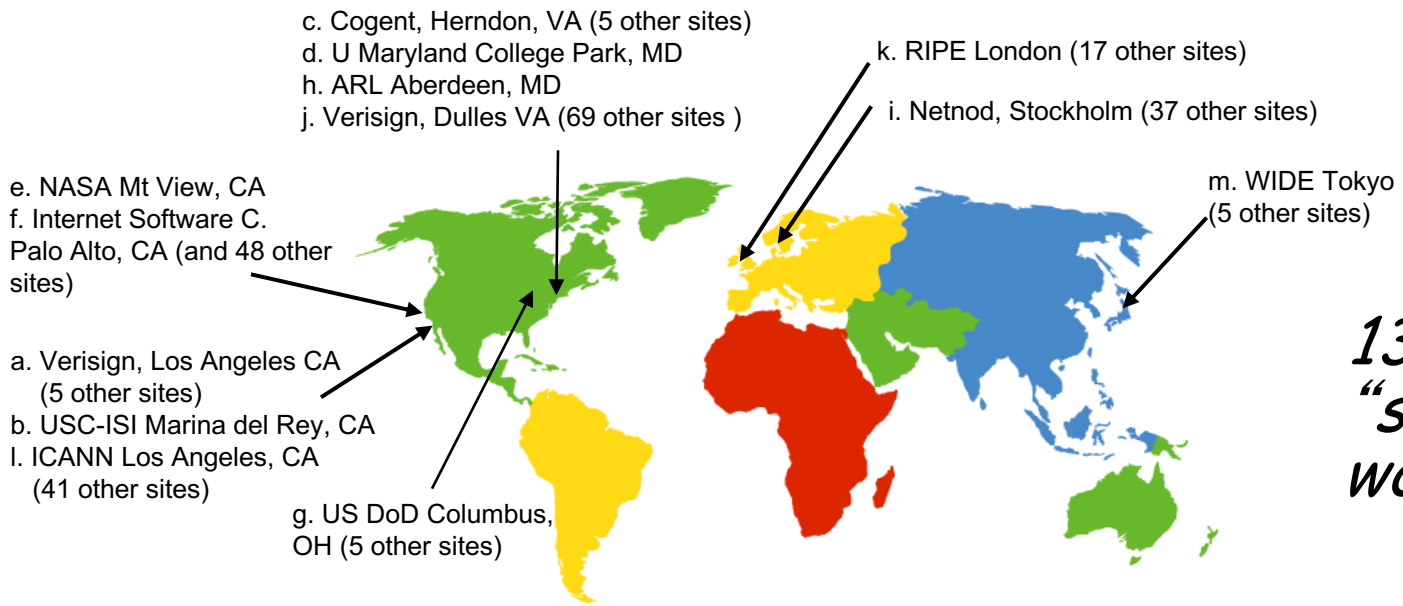


client wants IP for www.amazon.com; 1st approx:

- client queries root server to find com DNS server
- client queries .com DNS server to get amazon.com DNS server
- client queries amazon.com DNS server to get IP address for www.amazon.com

DNS: root name servers

- contacted by local name server that can not resolve name
- root name server:
  - could contacts authoritative name server if name mapping not known (in recursive queries)
  - gets mapping
  - returns mapping to local name server



13 root name "servers" worldwide

TLD, authoritative servers

top-level domain (TLD) servers:

- responsible for com, org, net, edu, aero, jobs, museums, and all top-level country domains, e.g.: uk, fr, ca, jp, eu
- Network Solutions maintains servers for .com TLD
- Educause for .edu TLD

authoritative DNS servers:

- organization's own DNS server(s), providing authoritative hostname to IP mappings for organization's named hosts
- can be maintained by organization or service provider

Local DNS name server

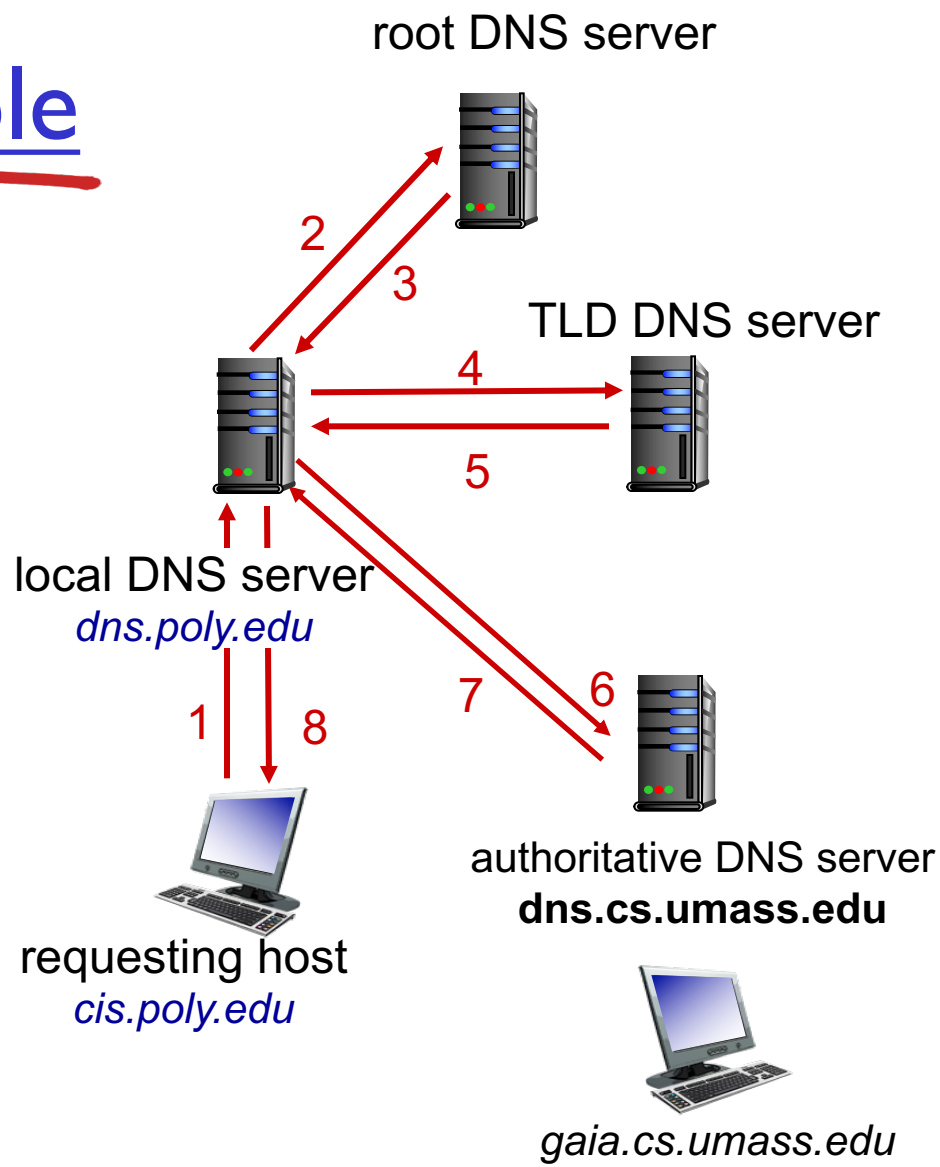
- does not strictly belong to hierarchy
- each ISP (residential ISP, company, university) has one
  - also called "default name server"
- when host makes DNS query, query is sent to its local DNS server
  - has local cache of recent name-to-address translation pairs (but may be out of date!)
  - acts as proxy, forwards query into hierarchy

DNS name resolution example

- host at cis.poly.edu wants IP address for gaia.cs.umass.edu

iterated query:

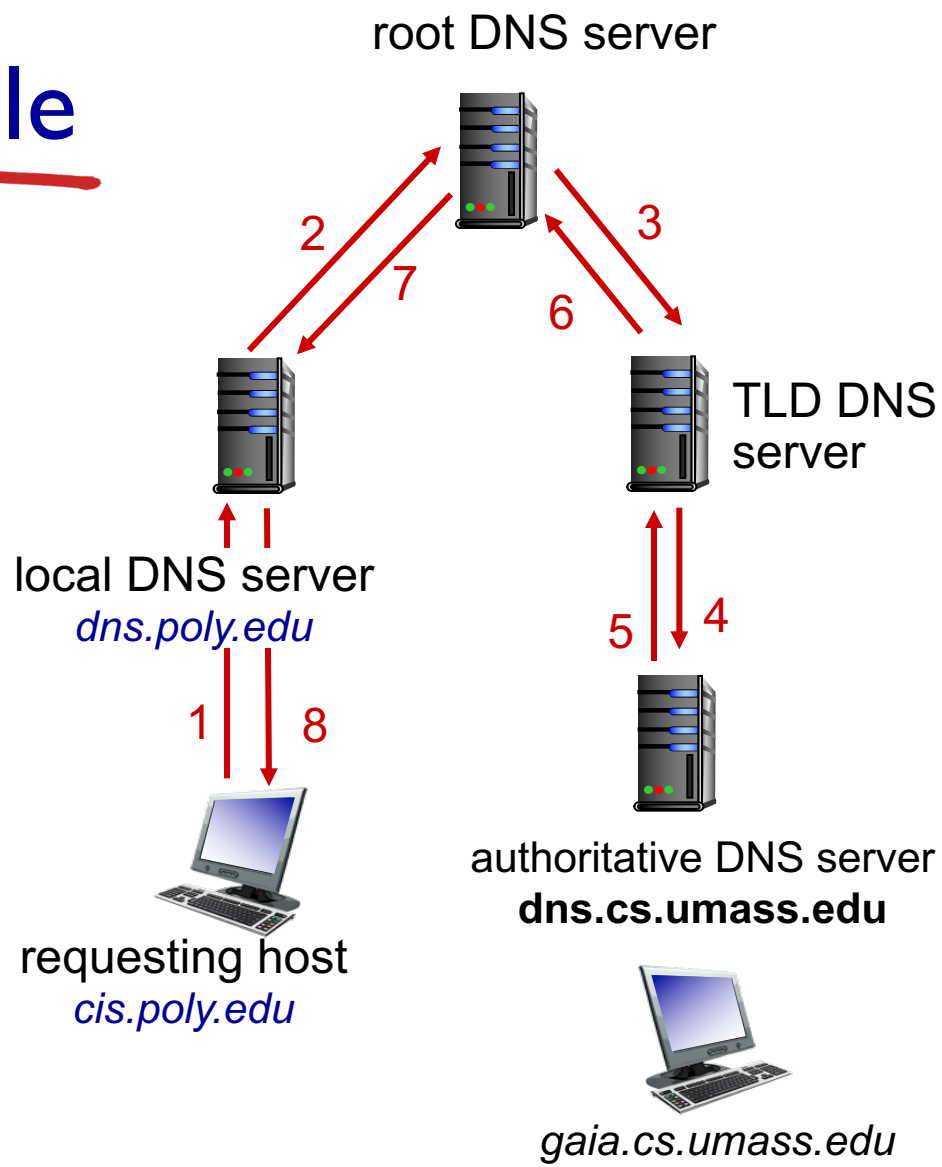
- contacted server replies with name of server to contact
- "I don't know this name, but ask this server"



DNS name resolution example

recursive query:

- puts burden of name resolution on contacted name server
- heavy load at upper levels of hierarchy?



DNS: caching, updating records

- once (any) name server learns mapping, it *caches* mapping
  - cache entries timeout (disappear) after some time (TTL)
  - TLD servers typically cached in local name servers
    - thus root name servers not often visited
- cached entries may be *out-of-date* (best effort name-to-address translation!)
  - if name host changes IP address, may not be known Internet-wide until all TTLs expire
- update/notify mechanisms proposed IETF standard
  - RFC 2136

DNS records

DNS: distributed db storing resource records (RR)

RR format: `(name, value, type, ttl)`

type=A

- **`name`** is hostname
- **`value`** is IP address

(relay.bar.foo.com,145.37.93.126,A)

type=NS

- **`name`** is domain (e.g., foo.com)
- **`value`** is hostname of authoritative name server for this domain

(foo.com,dns.foo.com,NS)

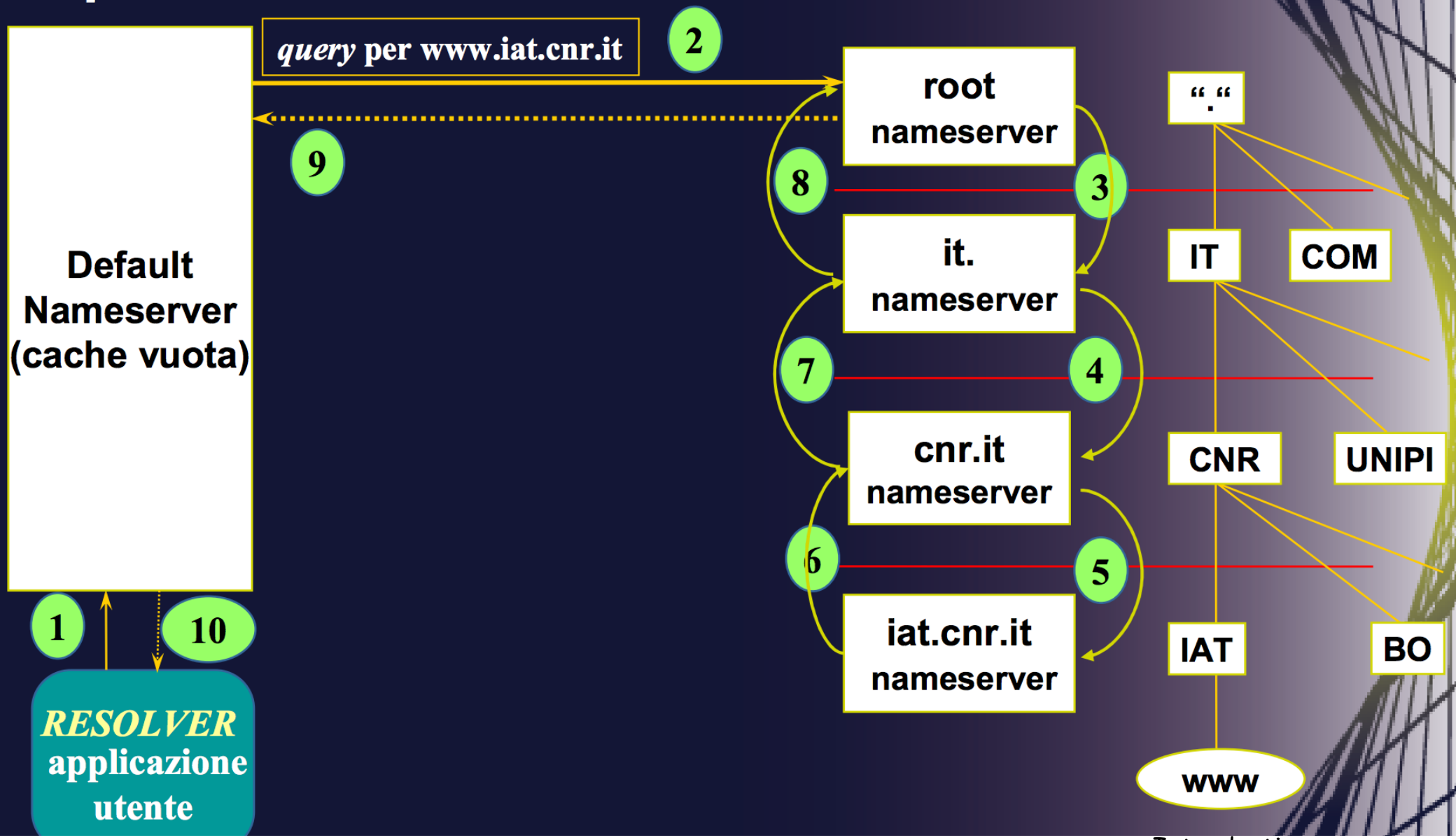
type=CNAME

- **`name`** is alias name for some "canonical" (the real) name
- `www.ibm.com` is really `servereast.backup2.ibm.com`
- **`value`** is canonical name

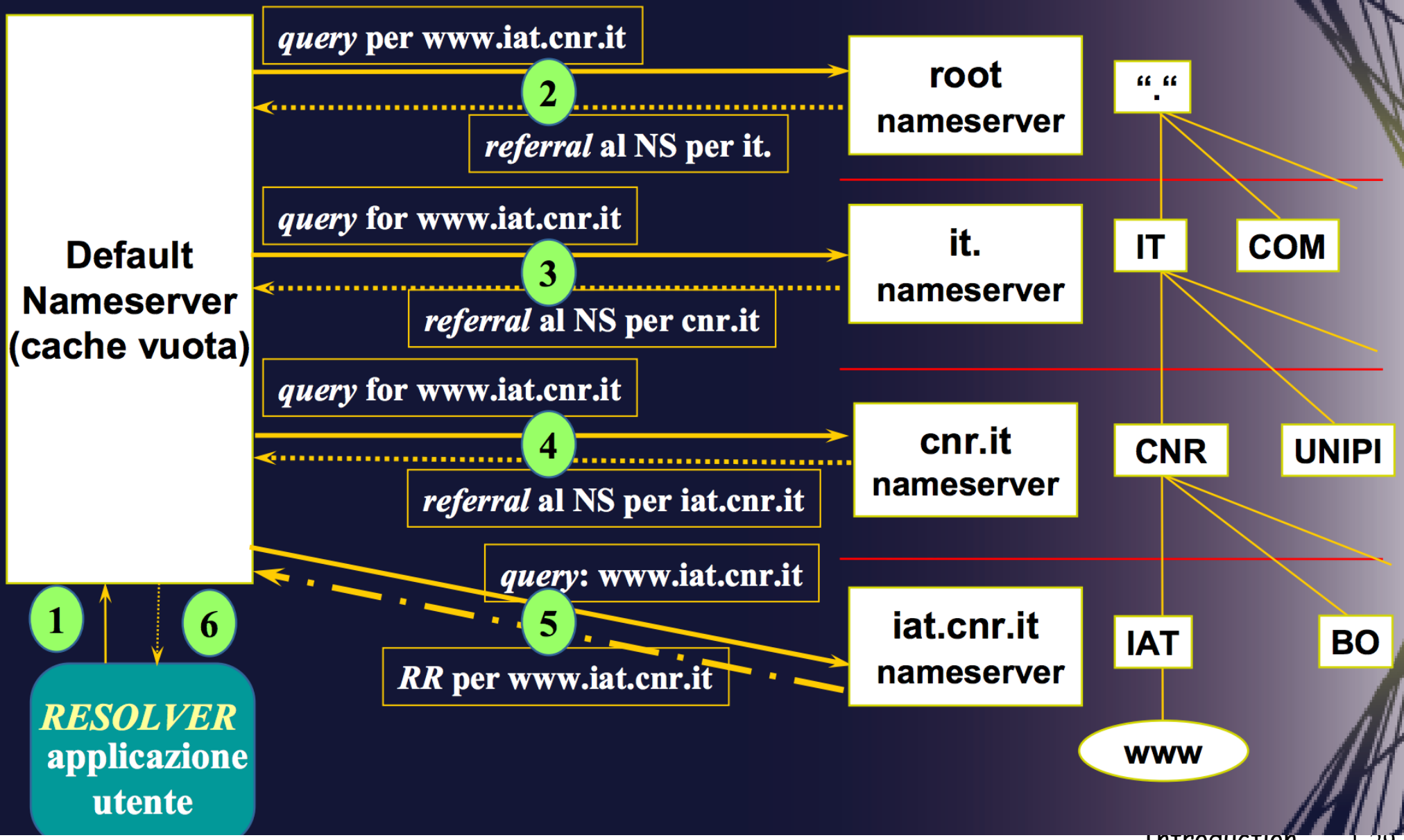
type=MX

- **`value`** is name of mailserver associated with **`name`**

Il processo ricorsivo di risoluzione dei nomi



Il processo iterativo di risoluzione dei nomi

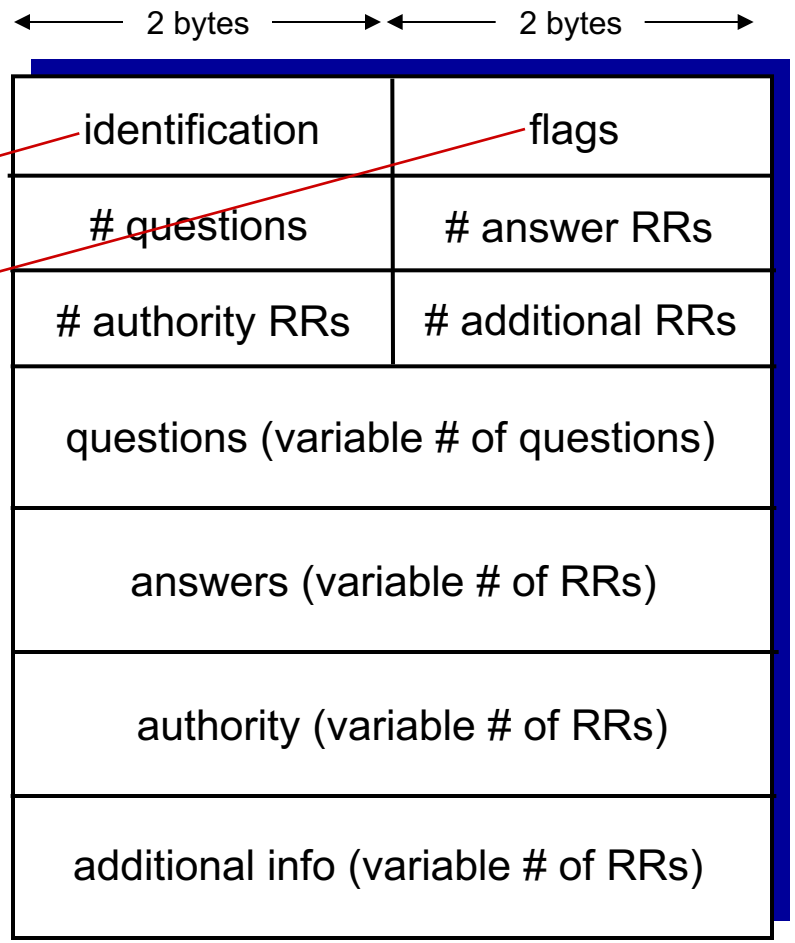


DNS protocol, messages

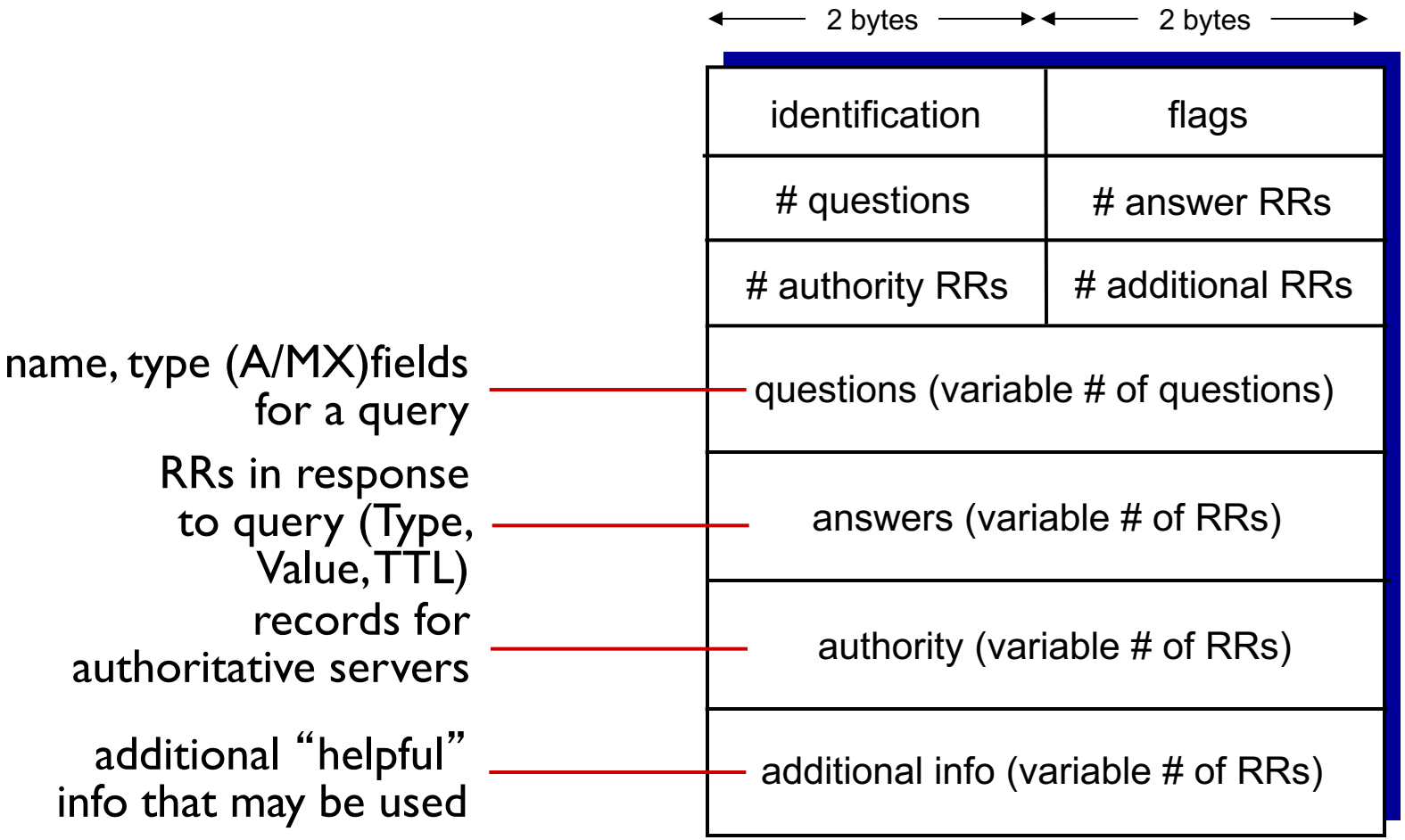
- *query* and *reply* messages, both with same *message format*

msg header

- identification: 16 bit # for query, reply to query uses same #
- flags:
  - query or reply
  - recursion desired
  - recursion available
  - reply is authoritative



DNS protocol, messages



Inserting records into DNS

- example: new startup "Network Utopia"
- register name networkuptopia.com at *DNS registrar* (e.g., Network Solutions)
  - provide names, IP addresses of authoritative name server (primary and secondary)
  - registrar inserts two RRs into .com TLD server:
    `(networkutopia.com, dns1.networkutopia.com, NS)`
    `(dns1.networkutopia.com, 212.212.212.1, A)`
- create authoritative server type A record for www.networkuptopia.com; type MX record for networkutopia.com

Attacking DNS

DDoS attacks

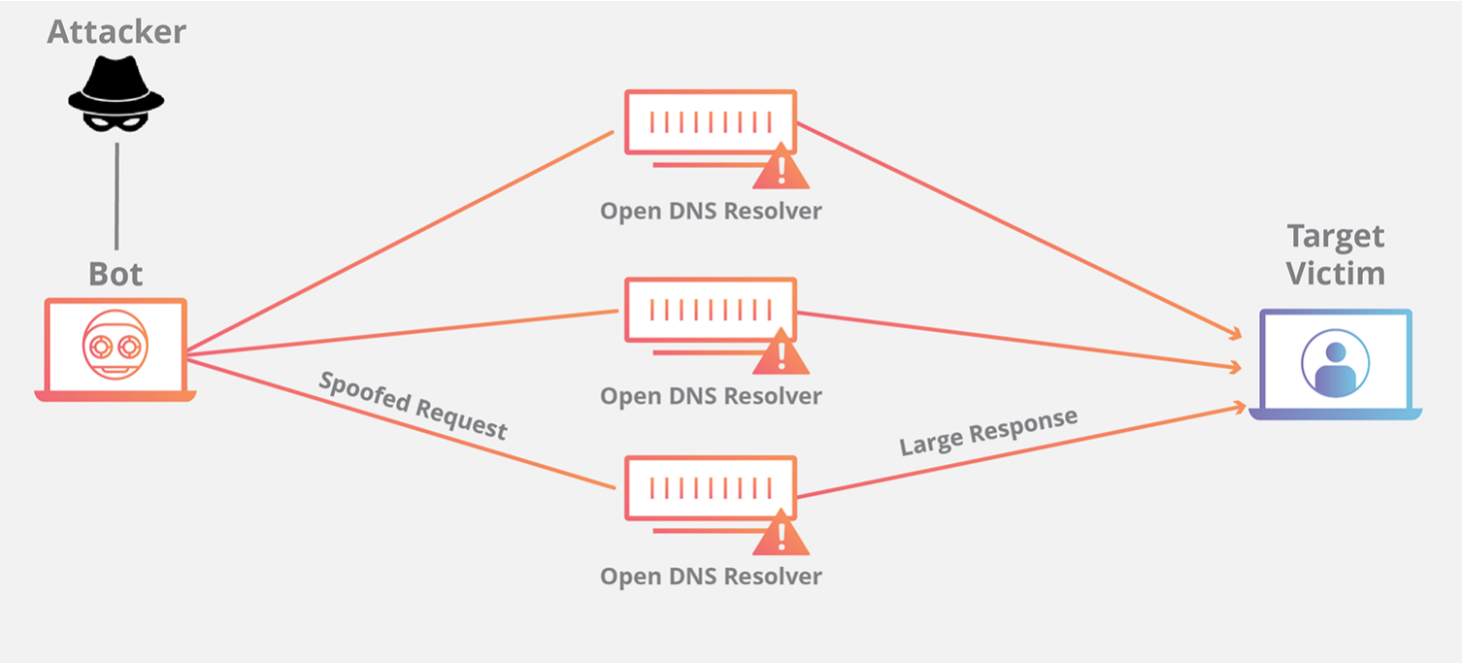
- Bombard root servers with traffic
  - Not successful to date
  - Traffic Filtering
  - Local DNS servers cache IPs of TLD servers, allowing root server bypass
- Bombard TLD servers
  - Potentially more dangerous

Redirect attacks

- Man-in-middle
  - Intercept queries
- DNS poisoning
  - Send bogus replies to DNS server, which caches

Exploit DNS for DDoS

- Send queries with spoofed source address: target IP
- Requires amplification



rtacks

iddle

pt queries

oning

ogus replies

server,

aches

- Traffic Filtering
- Local DNS servers cache IPs of TLD servers, allowing root server bypass

➢ Bombard TLD servers

- Potentially more dangerous

Exploit DNS for DDoS

❖ Send queries with spoofed source address: target IP

❖ Requires amplification

Perche' UDP?

- Less overhead
  - Messaggi corti
  - Tempo per set-up connessione di TCP lungo
  - Un unico messaggio deve essere scambiato tra una coppia di server (nella risoluzione contattati diversi server—se si usasse TCP ogni volta dovremmo mettere su la connessione!!)
- Se un messaggio non ha risposta entro un timeout?
  - Semplicemente viene riinviato dal resolver (problema
  - Risolto dallo strato applicativo)

Porta usata per il DNS: 53!!