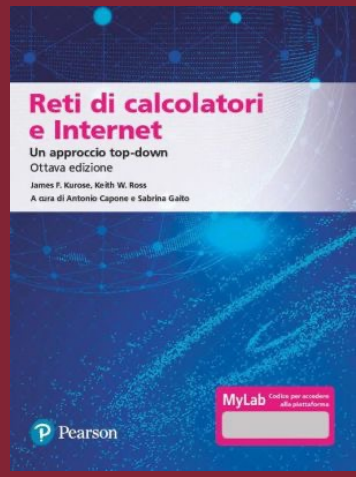
Reti di Elaboratori

Livello di Applicazione, FTP, DNS



SAPIENZA
UNIVERSITÀ DI ROMA

Alessandro Checco
alessandro.checco@uniroma1.it

Capitolo 2

Livello di applicazione: sommario

- Principi delle applicazioni di rete
- Web e HTTP
- Posta elettronica, SMTP, IMAP
- FTP
- Domain Name System: DNS
- Applicazioni P2P
- streaming video e content distribution networks
- programmazione socket con UDP e TCP

File Transfer Protocol (FTP)

- Programma di trasferimento file DA/A un host remoto
- Comando per accedere ed essere autorizzato a scambiare informazioni con l'host remoto

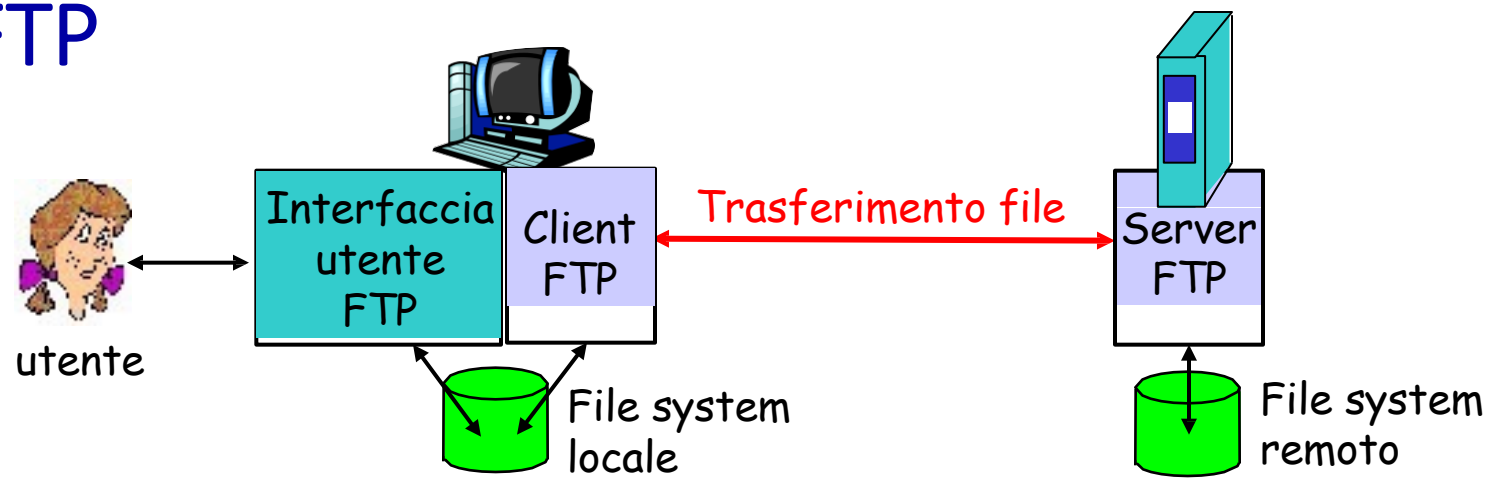
  `ftp NomeHost`

  vengono richiesti nome utente e password
- Trasferimento di un file da un host remoto

  `ftp> get file1.txt`
- Trasferimento di un file a un host remoto

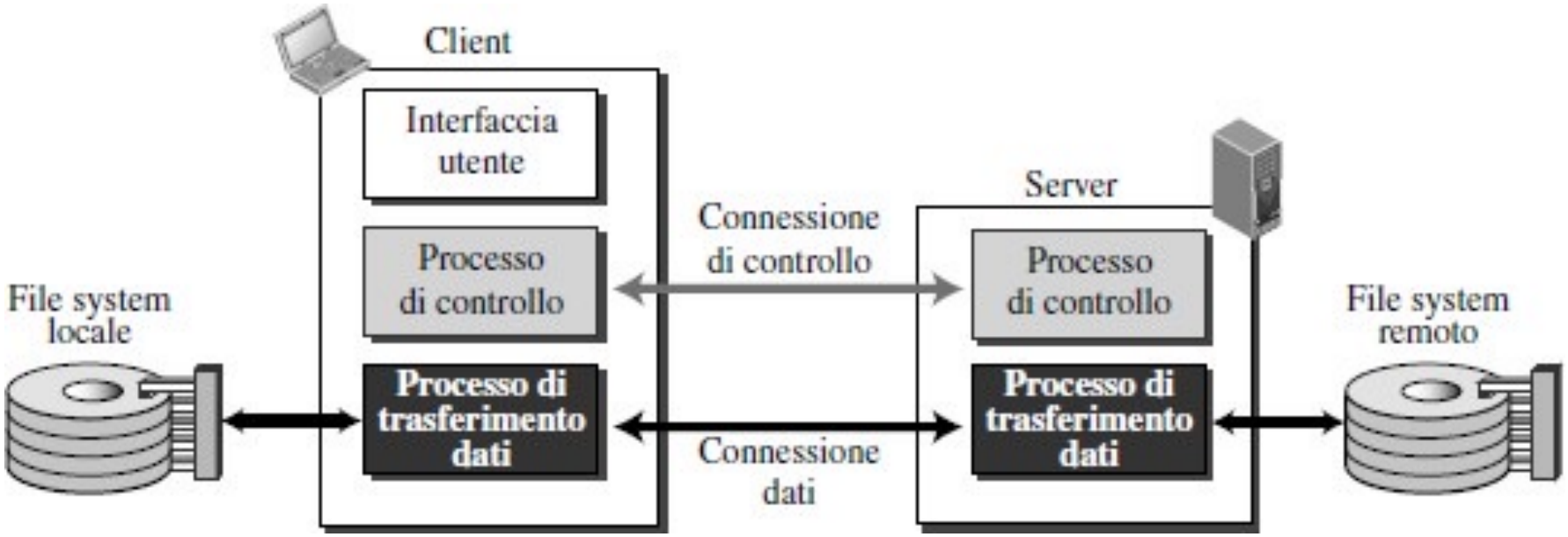
  `ftp> put file2.txt`
- Ci sono comandi per cambiare directory in locale e sull'host remoto, cancellare file, etc.

FTP



- Modello client/server
  - *client:* il lato che inizia il trasferimento (a/da un host remoto)
  - *server:* host remoto
- Quando l'utente fornisce il nome dell'host remoto (ftp NomeHost), il processo client FTP stabilisce una connessione TCP sulla porta 21 con il processo server FTP
- Stabilita la connessione, il client fornisce nome utente e password che vengono inviate sulla connessione TCP come parte dei comandi
- Ottenuta l'autorizzazione del server il client può inviare uno o più file memorizzati nel file system locale verso quello remoto (o viceversa)

FTP client e server



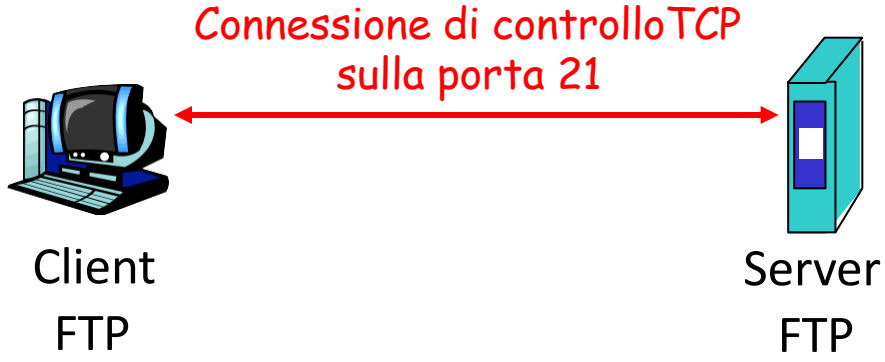
- **Connessione di controllo**: si occupa delle informazioni di controllo del trasferimento e usa regole molto semplici, così che lo scambio di informazioni si riduce allo scambio di una riga di comando (o risposta) per ogni interazione
- **Connessione dati**: si occupa del trasferimento del file

FTP: connessione di controllo

- **Connessione di controllo** (porta 21) viene usata per inviare informazioni di controllo
- L'apertura della connessione di controllo viene richiesta dal client al comando

```
ftp NomeHost
```

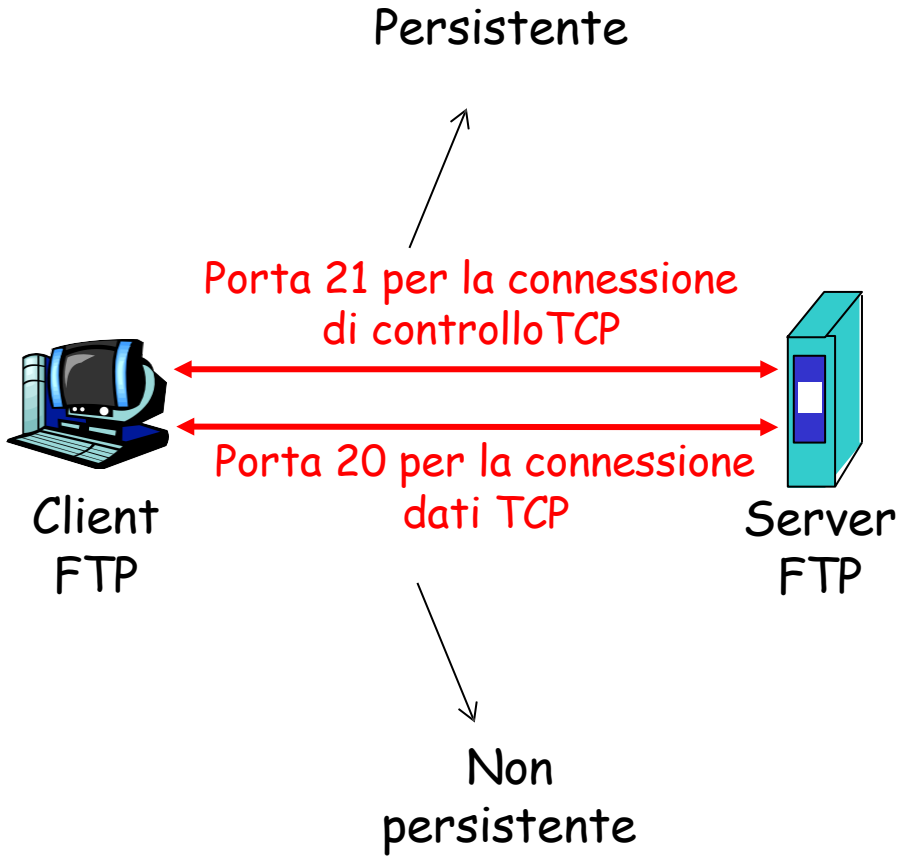
- tutti i comandi eseguiti dall'utente sono trasferiti sulla connessione di controllo
- Esempi di informazioni trasferite sulla connessione di controllo
  - Identificativo utente
  - Password
  - Comandi per cambiare directory
  - Comandi per richiedere invio (put) e ricezione (get) di file



- Connessione di controllo: "fuori banda" (*out of band*)
- HTTP utilizza la stessa connessione per messaggi di richiesta e risposta e file, per cui si dice che invia le informazione di controllo "in banda" (in-band)
- Il server FTP mantiene lo "stato": directory corrente, autenticazione precedente

FTP: connessione dati

- Connessione dati: quando il server riceve un comando per trasferire un file (es. `get`, `put`), apre una connessione dati TCP sulla porta 20 con il client
- Dopo il trasferimento di un file, il server chiude la connessione
- La connessione dati viene aperta dal server e utilizzata per il vero e proprio invio del file.
- Si crea una nuova connessione per ogni file trasferito all'interno della sessione



Comandi e risposte FTP

- Esiste una corrispondenza uno a uno tra il comando immesso dall'utente e quello FTP inviato sulla connessione di controllo
- Ciascun comando è seguito da una risposta spedita dal server al client (codice di ritorno)

Comandi comuni:

- Inviati come testo ASCII sulla connessione di controllo
- **USER *username***
- **PASS *password***
- **LIST**
  elenca i file della directory corrente (`dir`), la lista di file viene inviata dal server su una uova connessione dati
- **RETR *filename*** recupera (`get`) un file dalla directory corrente
- **STOR *filename***
  memorizza (`put`) un file nell'host remoto

Codici di ritorno comuni:

- Codice di stato ed espressione (come in HTTP)
- **331 Username OK, password required**
- **125 data connection already open; transfer starting**
- **425 Can't open data connection**
- **452 Error writing file**

Principali comandi FTP

Codifica standard chiamata NVT ASCII sia per i comandi che per le risposte

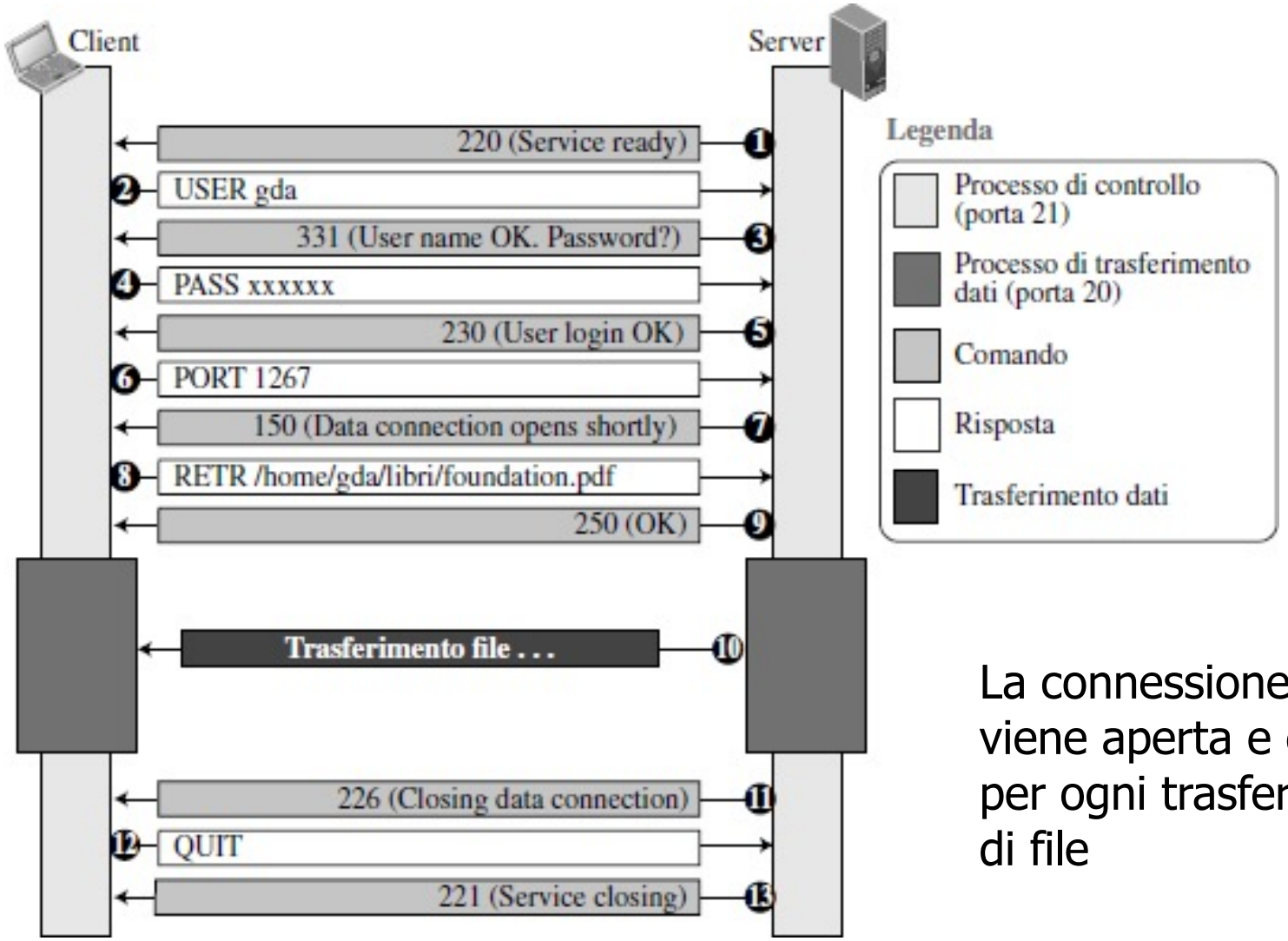
Comando	*Argomenti*	*Descrizione*
ABOR		Interruzione del comando precedente
CDUP		Sale di un livello nell'albero delle directory
CWD	Nome della directory	Cambia la directory corrente
DELE	Nome del file	Cancella il file
LIST	Nome della directory	Elenca il contenuto della directory
MKD	Nome della directory	Crea una nuova directory
PASS	Password utente	Password
PASV		Il server sceglie la porta
PORT	Numero di porta	Il client sceglie la porta
PWD		Mostra il nome della directory corrente
QUIT		Uscita dal sistema
RETR	Nome di uno o più file	Trasferisce uno o più file dal server al client
RMD	Nome della directory	Cancella la directory
RNTO	Nome (del nuovo) file	Cambia il nome del file
STOR	Nome di uno o più file	Trasferisce uno o più file dal client al server
USER	Identificativo	Identificazione dell'utente

Esempi di risposte FTP

- Le risposte sono composte da due parti: un numero di 3 cifre e un testo. La parte numerica costituisce il codice della risposta, quella di testo contiene i parametri necessari o informazioni supplementari
- La tabella riporta alcuni codici (non il testo)

Codice	*Descrizione*	*Codice*	*Descrizione*
125	Connessione dati aperta	250	Azione sul file OK
150	Stato del file OK	331	Nome dell'utente OK; in attesa della password
200	Comando OK	425	Non è possibile aprire la connessione dati
220	Servizio pronto	450	Azione sul file non eseguita; file non disponibile
221	Servizio in chiusura	452	Azione interrotta; spazio insufficiente
225	Connessione dati aperta	500	Errore di sintassi; comando non riconosciuto
226	Connessione dati in chiusura	501	Errore di sintassi nei parametri o negli argomenti
230	Login dell'utente OK	530	Login dell'utente fallito

Esempio



La connessione dati viene aperta e chiusa per ogni trasferimento di file

Livello di applicazione: sommario

- Principi delle applicazioni di rete
- Web e HTTP
- Posta elettronica, SMTP, IMAP
- FTP
- Domain Name System: DNS
- Applicazioni P2P
- streaming video e content distribution networks
- programmazione socket con UDP e TCP

DNS: domain name system

persone: molti identificatori:

- CF, nome, #matricola

Host Internet, router:

- Indirizzo IP (32 bit) - utilizzato per indirizzare i datagrammi
- "nome", ad es. cs.umass.edu - utilizzato dagli esseri umani

D: come mappare tra indirizzo IP e nome e viceversa?

Domain Name System (DNS):

- *database distribuito* implementato come una gerarchia di *name servers*
- *protocollo a livello di applicazione:* host e server DNS comunicano per *risolvere* nomi (traduzione indirizzo/nome)
  - nota: funzione "core" di Internet, *implementata come protocollo a livello di applicazione*
  - complessità sulla periferia della rete

DNS: servizi, struttura

Servizi DNS

- traduzione da nome host a indirizzo IP
- host aliasing
  - canonical names, alias names
- alias del server di posta
- distribuzione del carico
  - server Web replicati: più indirizzi IP possono corrispondere a un unico nome

D: Perché non centralizzare DNS?

- singolo punto di fallimento
- volume del traffico
- distanza dai nodi
- manutenzione sarebbe difficile

R: non scala!

- Solo server DNS Comcast: 600 miliardi di query DNS al giorno

Servizi DNS: aliasing

- Permette di associare un nome più semplice da ricordare a un nome complesso
- **Host aliasing:** un host può avere uno o più sinonimi (alias)
  - Esempio: `relay1.west-coast.enterprise.com` potrebbe avere due sinonimi, quali `enterprise.com` e `www.enterprise.com`
  - `relay1.west-coast.enterprise.com` è un hostname ***canonico***
  - `enterprise.com` e `www.enterprise.com` sono ***alias***
  - Gli alias sono più facili da ricordare
  - l DNS può essere invocato da un'applicazione per conoscere l'hostname canonico di un sinonimo così come l'IP
- Mail server aliasing: spesso i mail server e il web server di una società hanno lo stesso alias, ma nomi canonici diversi (ibm.com per entrambi ma con nomi canonici www.ibm.com e mail.ibm.com)
- Il DNS può essere invocato da un'applicazione per avere il nome canonico di un alias e il suo indirizzo IP

Servizi DNS: distribuzione del carico

- DNS viene utilizzato per distribuire il carico tra server replicati (es. web server)
- I siti con molto traffico (es. cnn.com) vengono replicati su più server, e ciascuno di questi gira su un sistema terminale diverso e presenta un indirizzo IP differente
- Hostname canonico associato a un insieme di indirizzi IP
- Il DNS contiene l'insieme di indirizzi IP
- Quando un client effettua un richiesta DNS per un nome mappato in un insieme di indirizzi, il server risponde con l'insieme di indirizzi ma variando l'ordinamento a ogni risposta
- La rotazione DNS distribuisce il traffico sui server replicati

DNS - Specifiche

- Database decentralizzato di grandissime dimensioni
  - miliardi di record (ma semplici)
- Alto numero di richieste
  - Migliaia di miliardi di richieste al giorno (perlopiù lettura)
  - Performance è importante: < ms
- Decentralizzato fisicamente e dal punto di vista organizzativo
  - Milioni di aziende diverse sono responsabili per il loro record
- Core Internet feature
  - Sicuro
  - Affidabile

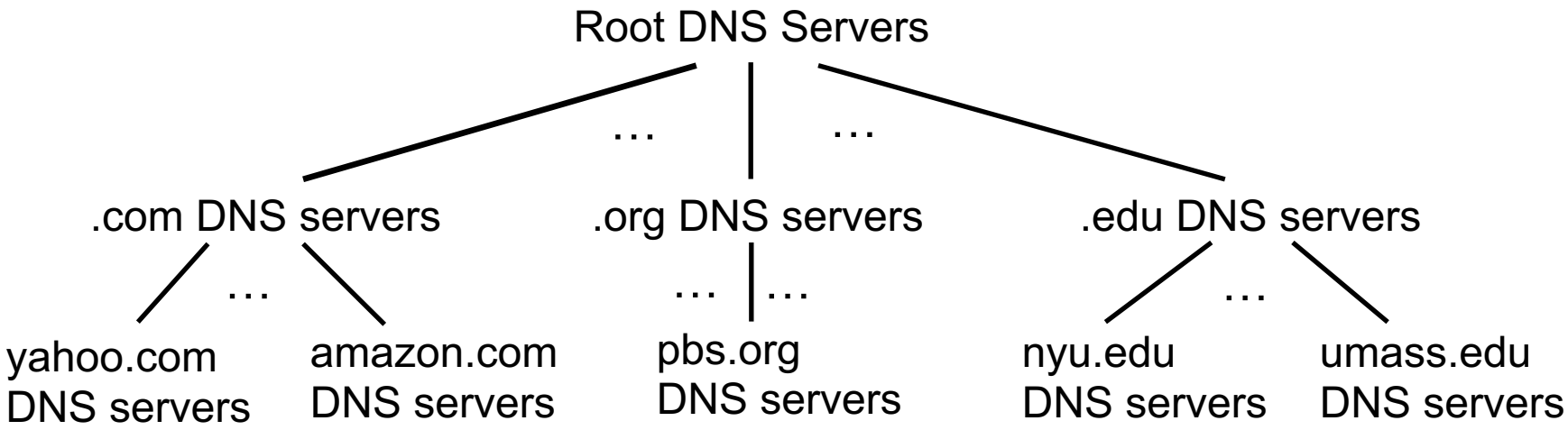
Gerarchia server DNS

- Nessun server DNS mantiene il mapping per tutti gli host in Internet
- Il mapping è distribuito su svariati server DNS
- Come si possono memorizzare le 2^{32} coppie (IP, nome host) su più server in modo da poter poi fare una ricerca in tempo breve?

Gerarchia server DNS

- Consideriamo i seguenti domini (ognuno avrà il proprio IP):
  - www.uniroma1.it
  - www.unipi.it
  - www.google.com
  - www.di.uniroma1.it
  - www.cnn.com
  - www.cnr.it
  - www.umass.edu
  - www.northeastern.edu
- Come potremmo raggrupparli in modo da fare una ricerca in tempi rapidi?

DNS: un database gerarchico distribuito



Root

Top Level Domain

Authoritative

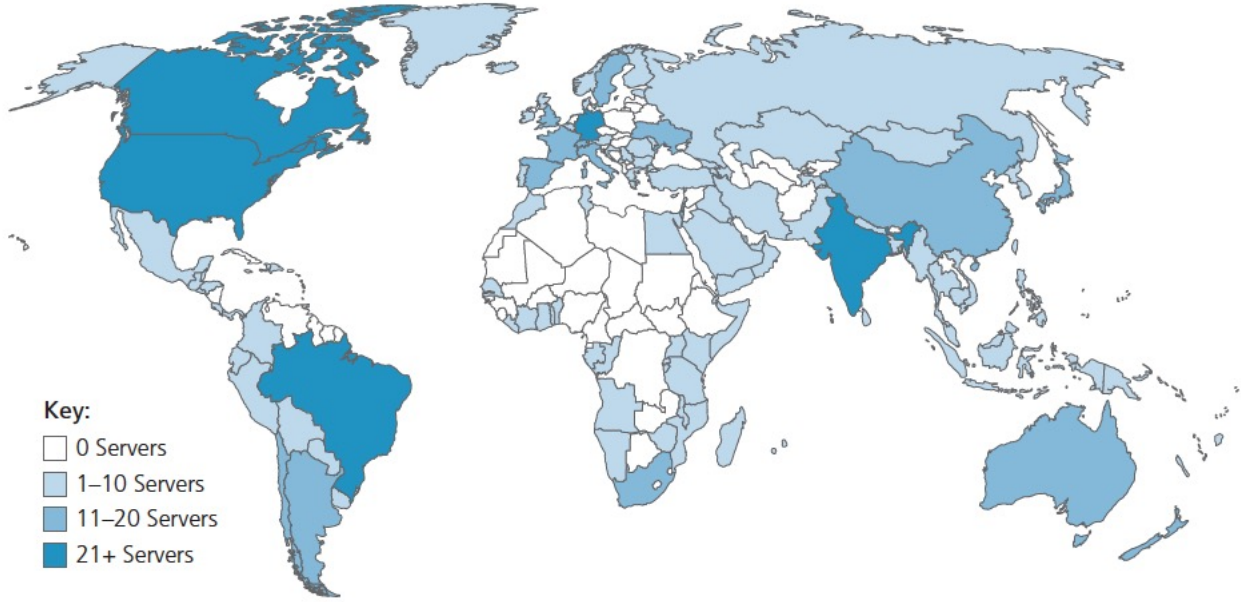
Il cliente vuole l'indirizzo IP per www.amazon.com; semplificando:

- protocollo end-to-end eseguito dal client con paradigma client-server usando UDP
- il client interroga il root server per trovare *il server DNS (TLD)* **.com**
- il client interroga il *server DNS* **.com** per ottenere il Server DNS autoritativo di amazon.com
- il client interroga il Server DNS di amazon.com per ottenere l'indirizzo IP di www.amazon.com

DNS: root

- contatto ufficiale di ultima istanza per server DNS che non riescono a risolvere un nome
- funzione Internet *core*
  - Internet non potrebbe funzionare senza di essa!
  - DNSSEC: fornisce sicurezza (autenticazione e integrità dei messaggi)
- ICANN (Internet Corporation for Assigned Names and Numbers) gestisce il dominio DNS principale

13 root name server principali (logici) distribuiti geograficamente, ognuno replicato molte volte (~200 server fisici negli Stati Uniti)



Top-Level Domain e authoritative server

Server Top-Level Domain (TLD):

- responsabili di .com, .org, .net , . edu , .aero, .jobs, .museums e tutti i domini nazionali di primo livello, ad esempio: .it, .uk, .fr , .ca, .jp
- Network Solutions: registro autoritativo per .com, .net
- Educause: .edu

Server DNS di competenza (authoritative):

- ogni organizzazione con host Internet pubblicamente accessibili (server web/posta) deve fornire i record DNS di pubblico dominio che mappano i nomi di tali host in indirizzi IP
- può essere mantenuto dall'organizzazione o dal fornitore di servizi
- In genere sono due server (primario e secondario)

Server DNS locali

- non appartiene strettamente alla gerarchia
- ogni ISP (ISP residenziale, azienda, università) ne ha uno
  - chiamato anche *default name server*
- quando l'host effettua una query DNS, questa viene inviata al server DNS locale dell'host, il quale:
  - ha una cache locale delle recenti coppie di mappatura nome-indirizzo (ma potrebbe non essere aggiornato!)
  - funge da proxy, inoltra la query nella gerarchia per risolvere la query (inizia il processo di risoluzione)

Qual è il vostro server DNS locale?

- Windows:
  - ipconfig /all
- MacOs
  - scutil --dns
- Linux
  - cat /etc/resolv.conf

Etichette dei domini generici

Tabella 2.10 Etichette dei domini generici.

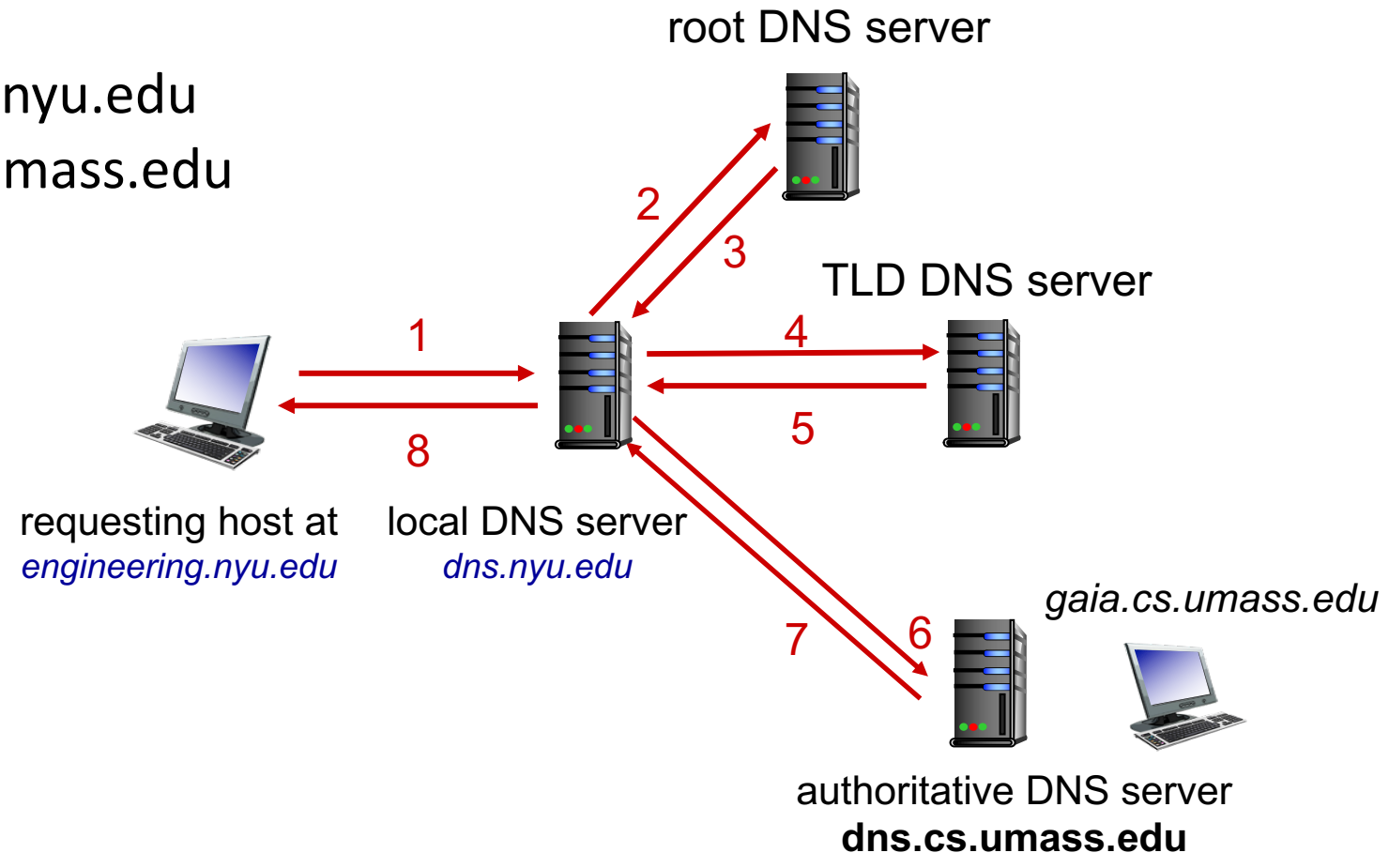
Etichetta	*Descrizione*	*Etichetta*	*Descrizione*
aero	Compagnie aeree e aziende aerospaziali	int	Organizzazioni internazionali
biz	Aziende (simile a com)	mil	Organizzazioni militari
com	Organizzazioni commerciali	museum	Musei
coop	Associazioni di cooperazione	name	Nomi di persone
edu	Istituzioni educative	net	Organizzazioni che si occupano di reti
gov	Istituzioni governative	org	Organizzazioni senza scopo di lucro
info	Fornitori di servizi informativi	pro	Organizzazioni professionali

Risoluzione dei nomi DNS: query iterativa

Esempio: host su engineering.nyu.edu
vuole l'indirizzo IP di gaia.cs.umass.edu

Query iterativa:

- il server contattato risponde con il nome del prossimo server da contattare
- "Non conosco questo nome, ma chiedi a questo server"

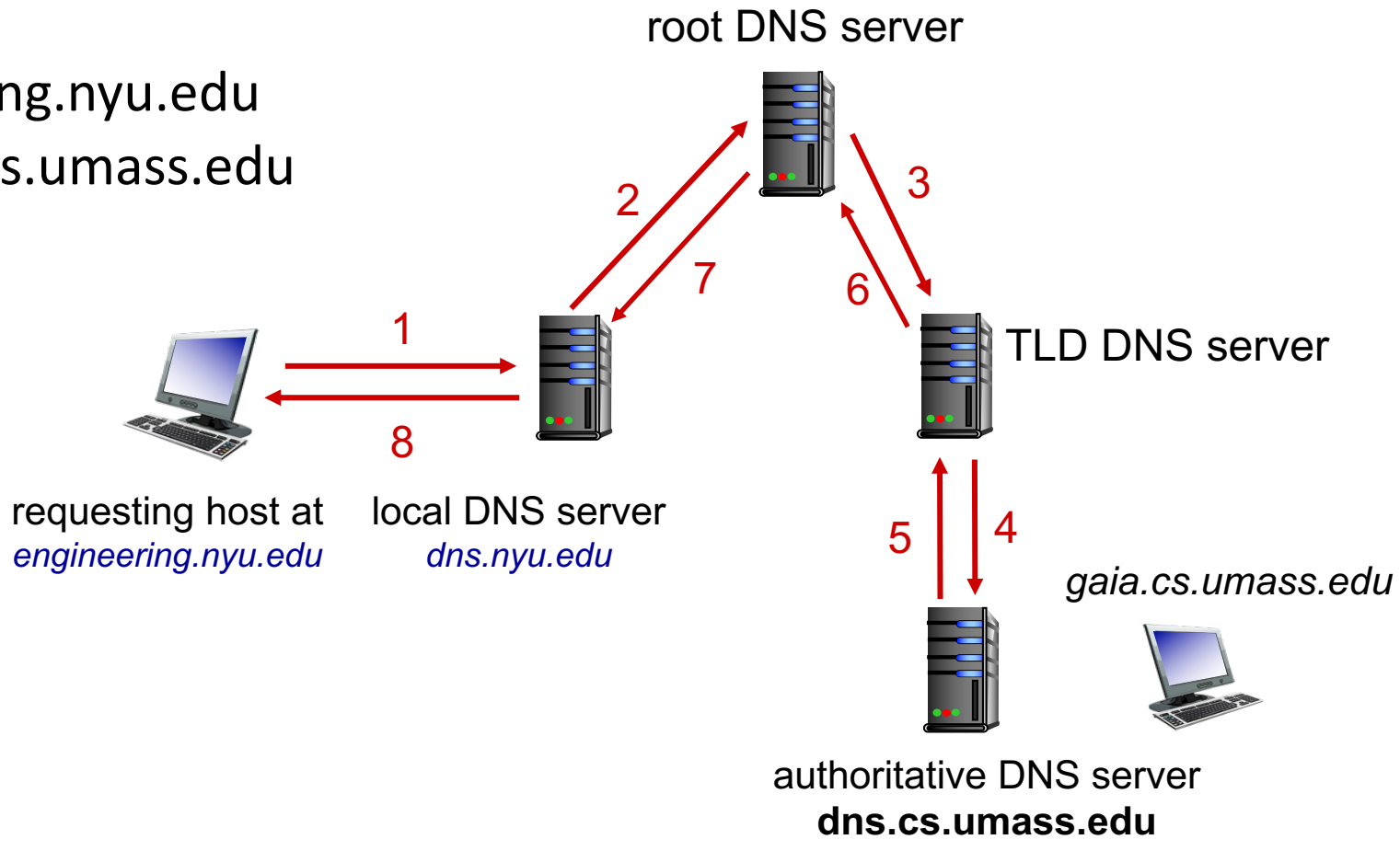


Risoluzione dei nomi DNS: query ricorsiva

Esempio: host su engineering.nyu.edu vuole l'indirizzo IP di gaia.cs.umass.edu

Query ricorsiva:

- pone l'onere della risoluzione sul name server contattato
- carico pesante ai livelli superiori della gerarchia?



Caching, aggiornamento dei record DNS

- una volta che un server DNS apprende una mappatura, la memorizza e la usa per rispondere a query future
  - i record in cache vengono cancellati dopo un TTL (Time-to-live)
  - I server TLD in genere sono nella cache nei server DNS locali
    - quindi i server DNS radice non vengono visitati spesso
- le voci nella cache potrebbero non essere *aggiornate* (il servizio di risoluzione dei nomi è best-effort!)
  - se il nome dell'host cambia l'indirizzo IP, potrebbe non essere noto a livello di Internet fino alla scadenza di tutti i TTL!
- meccanismi di aggiornamento/notifica proposti
  - standard IETF (RFC 2136)

DNS records

DNS: distributed database storing resource records (RR)

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

type=A

- `name`: hostname (relay1.bar.foo.com)
- `value`: IP address (45.37.93.126)

type=NS

- `name`: domain (foo.com)
- `value`: hostname of authoritative name server for this domain (dns.foo.com)

type=CNAME

- `name`: è l'alias nel vero nome (canonico)
  - www.ibm.com in realtà è servereast.backup2.ibm.com
- `value`: canonical name

type=MX

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

Esempio (importante)

- Server di competenza per un hostname
  - Contiene un record di tipo A per l'hostname
  - Es. (`corsi.di.uniroma1.it, 131.111.45.68, A`)
- Server non di competenza per un dato hostname
  - Contiene un record di tipo NS per il dominio che include l'hostname
  - Contiene un record di tipo A che fornisce l'indirizzo IP del server DNS nel campo `value` del record NS
- Es.:
  - Un server TLD it non è competente per l'host `corsi.di.uniroma1.it`
  - Contiene (`uniroma1.it, dns.uniroma1.it, NS`)
    (`dns.uniroma1.it, 128.119.40.111, A`)

Restrizioni ai record DNS

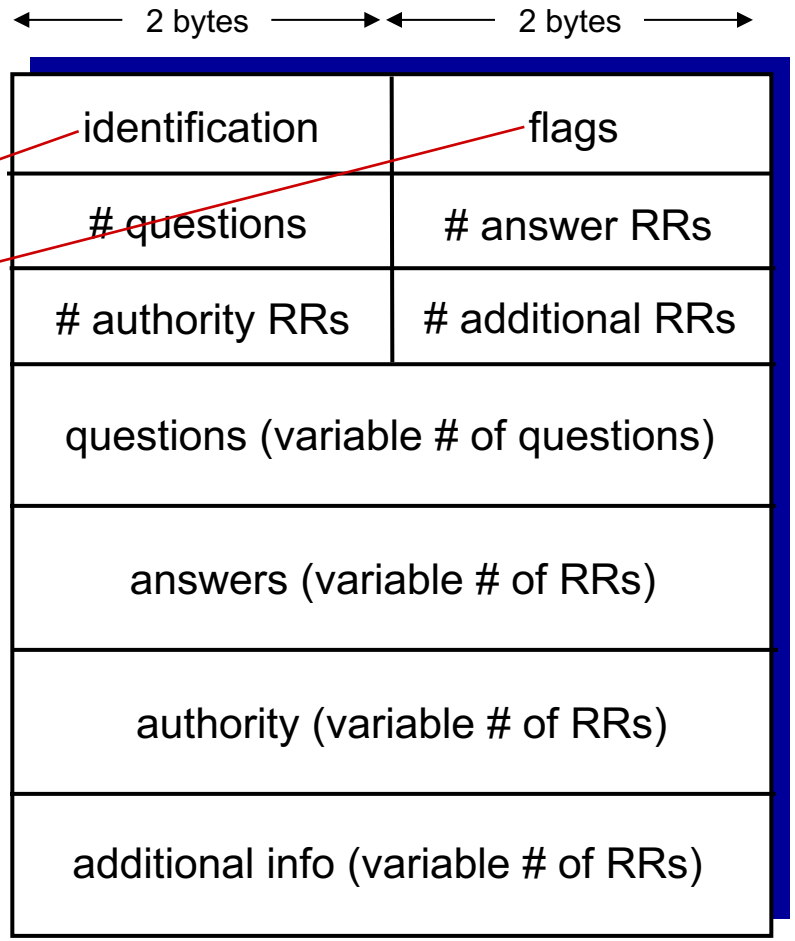
- CNAME non può coesistere con altri record di altro tipo per lo stesso dominio: non posso avere un record A e un record CNAME per lo stesso (sub)dominio nella stessa tabella
- CNAME non può essere usato nei domini di root. Quindi il DNS server di example.com NON può contenere: **`example.com.  CNAME  alias.example.net`**.
- Se si tengono in cache valori di un server non di competenza, bisogna anche fornire il NS autoritativo quando rispondiamo a una query
- Raccomandati
  - MX dovrebbe essere usato su un nome canonico e non un CNAME
  - quando si fornisce il NS di un server autoritativo, bisogna anche fornire il record A (glue) per evitare riferimenti circolari

Messaggi del protocollo DNS

DNS *query* e messaggi di *risposta* hanno lo stesso *formato:*

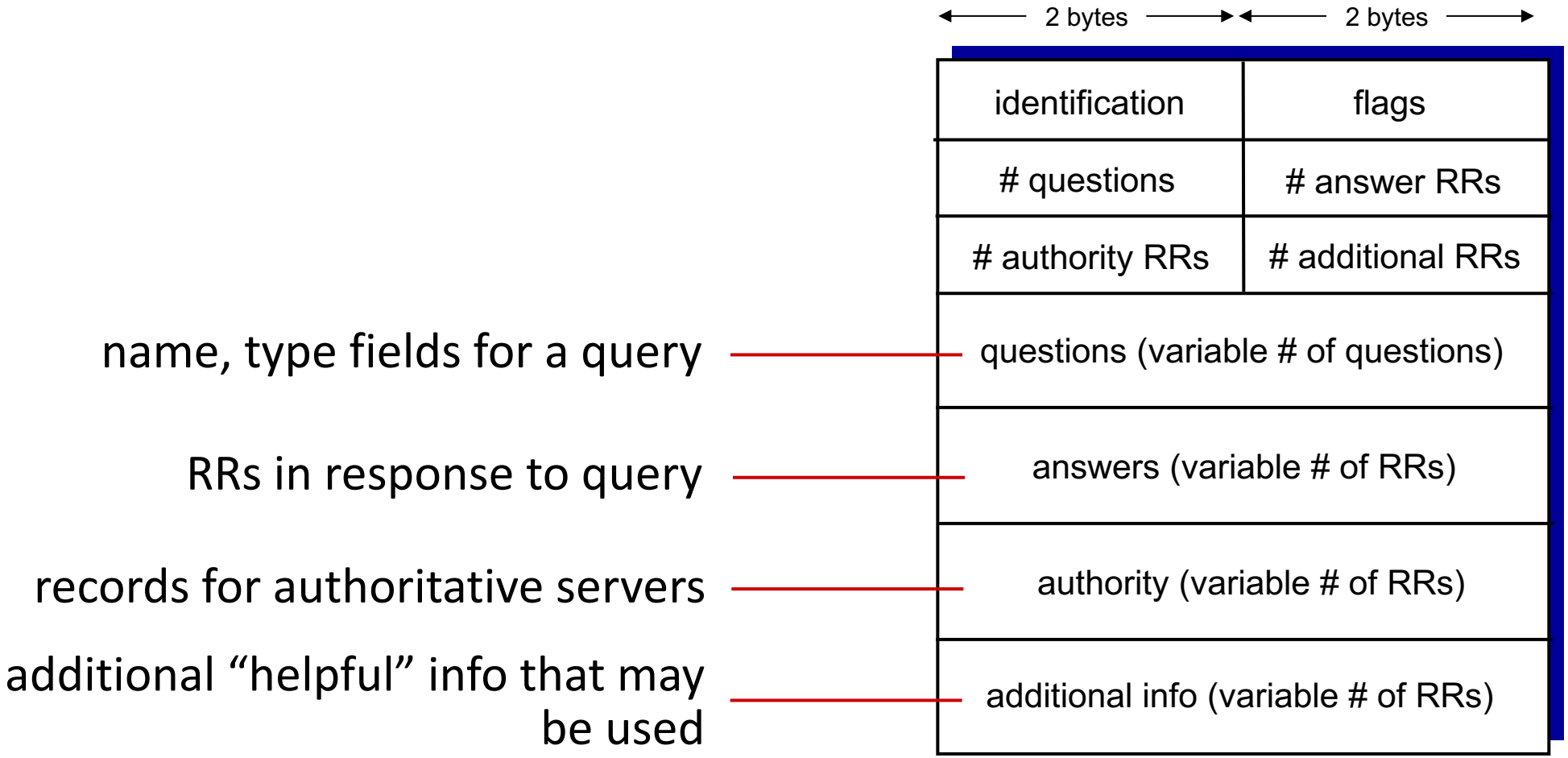
intestazione del messaggio:

- identificazion: 16 bit # per la query, la risposta alla query utilizza lo stesso #
- flags:
  - query/reply
  - recursion desired
  - recursion available
  - reply is authoritative



Messaggi del protocollo DNS

DNS *query* e messaggi di *risposta* hanno lo stesso *formato:*



Ruolo del punto "."

- "." rappresenta il dominio di root
- nei record di configurazione di un DNS server autoritativo, ogni dominio senza il "." alla fine rappresenta un dominio locale
  - nel dominio example.com potremmo avere questo record

```
orange  300   IN    A     1.2.3.4
fruit   300   IN    CNAME orange
```

  - che si traduce con

```
orange.example.com.  300   IN    A     1.2.3.4
fruit.example.com.   300   IN    CNAME orange.example.com.
```

- nslookup -type=ns di da dentro eduroam manderà una query per di.uniroma1.it

Inserimento di record nel DNS

Esempio: nuova startup “Network Utopia”

- nome di registrazione networkuptopia.com presso *il registrar DNS* (ad es. Network Solutions)
  - fornire nomi e indirizzi IP del server DNS di competenza desiderato (primario e secondario) al registrar
  - il registrar inserisce record NS e A nel server TLD di .com:

`(networkutopia.com, dns1.networkutopia.com, NS)`

`(dns1.networkutopia.com, 212.212.212.1, A)`

- La startup crea un server di competenza locale con IP `212.212.212.1`
  - Authoritative DNS server con record per tutti i sottodomini:
    - record di tipo A per www.networkuptopia.com, subdomain.networkutopia.com, etc.
    - record MX per networkutopia.com (email)

Sicurezza DNS

Attacchi DDoS

- bombarda i root server con traffico
  - mai riuscito fino ad oggi
  - filtraggio del traffico
  - i server DNS locali memorizzano nella cache gli IP dei server TLD, consentendo il bypass del server root
- bombardare i server TLD
  - potenzialmente più pericoloso

Attacchi di tipo redirect

- man-in-the-middle
  - Intercetta le query DNS
- DNS poisoning
  - manda risposte sbagliate al server DNS, che le mette in cache

Sfruttare DNS per DDoS

- inviare query con indirizzo di origine falsificato: IP bersaglio (reflection)
- richiede amplificazione (ad es. query with 'ANY')

DNSSEC [RFC 4033]

Attacco DNS amplification

- L'attore malevolo manda pacchetti UDP (query DNS) a un server DNS con IP falsificato, contenente l'IP dell'obbiettivo come mittente
- La query è di tipo ANY: in questo modo la risposta sarà molto grande, contenendo tutti i campi in possesso del server DNS riguardo a un dominio
- Un piccolo pacchetto mandato dall'attore malevolo corriponde a tanto traffico verso il target da parte del server DNS -> amplification
- Questo attacco si può distribuire usando più server DNS e più mittenti malevoli per condurre un Distributed Denial of Service attack

Mitigare questo tipo di attacco

- Non rispondere a richieste ANY
- Cercare di identificare IP spoofing: se l'IP di un pacchetto in uscita da una rete ha un IP mittente facente parte di un'altra rete si può identificare

Perché 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 ri-inviato dal resolver (problema risolto dallo strato applicativo)

Prova pratica

- `Nslookup`: command-line tool to query Internet DNS interactively
- `nslookup` dal prompt dei comandi

- `nslookup www.di.uniroma1.it`
- `nslookup -type=a www.google.com 1.1.1.1`
- `nslookup -type=ns google.com [and then use it for the next]`
- `nslookup -type=a mail.google.com xxxx [IP above]`
- `nslookup -type=mx google.com xxxx`
- `nslookup -debug -type=cname www.ibm.com`
- `interattivo nslookup iterativo: avviare s`
  - `server 8.8.8.8; set type=ns; .; set type=a; [use one ns for root]`
  - `server [ipofns]; set type=ns; com.`
  - `server [ipofns]; google.com.; server [ipofns]`
  - `set type=a; www.google.com.`
  - `set debug e riprova`

Chiarimento sul TTL

- Il TTL è il tempo in cui viene richiesto di mantenere il valore in cache da parte del server di competenza
- Non è detto che venga onorato
  - non c'è spazio
  - altre policies (ad es. 5 minuti fisso)
- Alcuni server DNS permettono il flush manuale del proprio RR
- Anche se un TTL molto corto potrebbe creare problemi all'intera rete, in pratica i sistemi riescono a gestire questo traffico

Esercizio d'esame

- Quali dei seguenti resource record non sono corretti (TTL non è mostrato)? Motivare la risposta (l'ordine è domain, value, type)

1. `< it nameserver.cnr.it NS >`

2. `< it nameserver.cnr.it A >` nameserver.cnr.it non è un indirizzo IP!

3. `< it nameserver.cnr.it CNAME >` CNAME non può essere usato alla ROOT di un dominio

4. `< it 151.100.27.38 NS >` 151.100.27.38 non è un hostname/domain name

5. `< nameserver.cnr.it 151.100.27.38 A>`