Università degli Studi di Roma "Tor Vergata"
Laurea in Informatica

Sistemi Operativi e Reti
(modulo Reti)
a.a. 2024/2025

Livello di applicazione (parte2)

dr. Manuel Fiorelli

manuel.fiorelli@uniroma2.it

https://art.uniroma2.it/fiorelli

Basate sulle slide del libro di testo:
https://gaia.cs.umass.edu/kurose_ross/ppt.php

Livello di applicazione: panoramica

- Principi delle applicazioni di rete
- Web e HTTP
- E-mail, SMTP, IMAP
- DNS: il servizio di directory di Internet
- Applicazioni P2P
- Streaming video e reti di distribuzione di contenti
- Programmazione delle socket programming con UDP e TCP

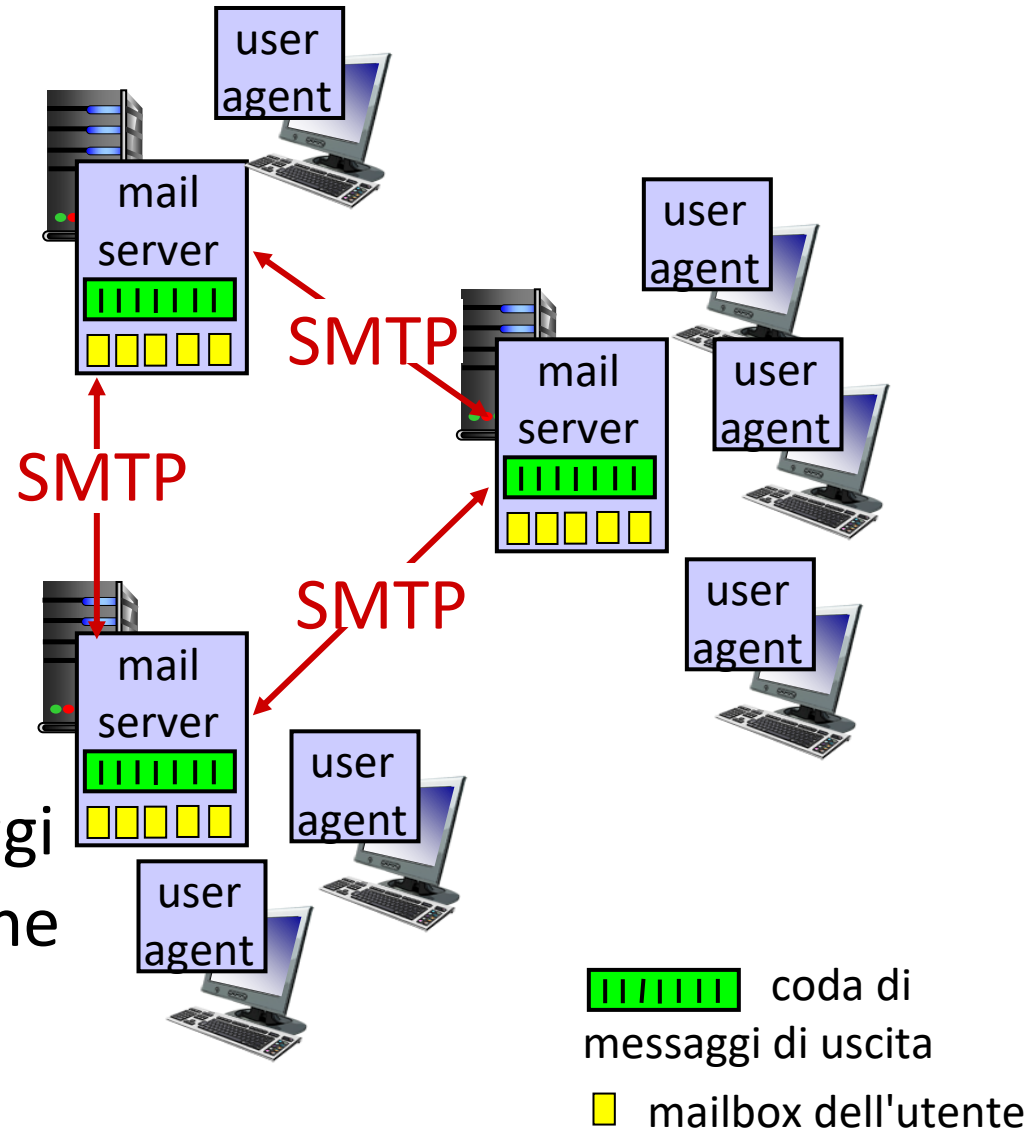
E-mail

Tre componenti principali:

- user agent (o *agenti utenti*)
- mail server (o *server di posta*)
- simple mail transfer protocol: SMTP

User Agent

- detto anche “mail reader”
- composizione, editing, lettura dei messaggi
- esempi: Outlook, client di posta dell'iPhone
- i messaggi in uscita o in arrivo sono memorizzati sul server



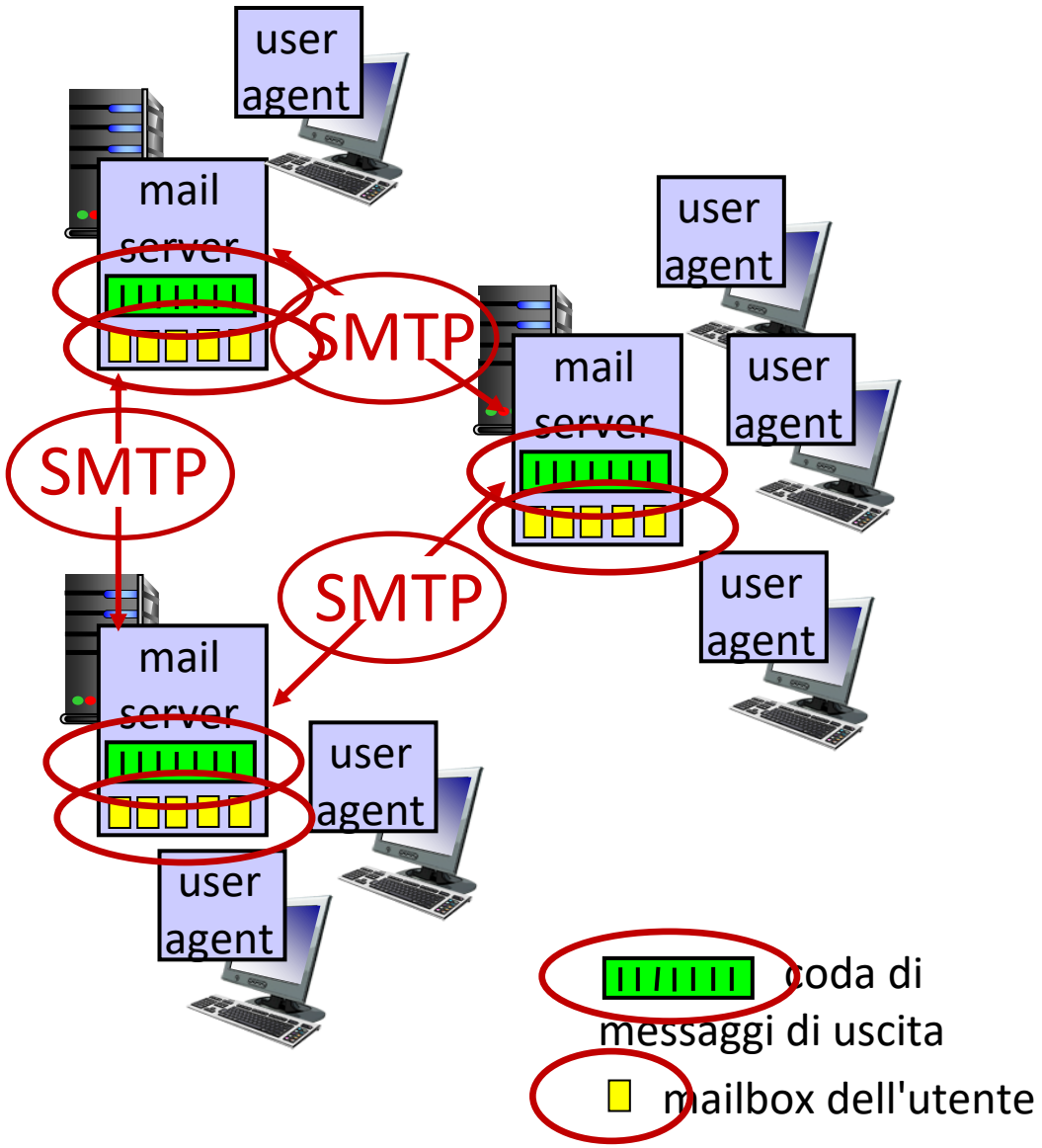
E-mail: mail servers

mail server:

- *mailbox* (casella di posta) contiene i messaggi in arrivo per l'utente
- *coda di messaggi* da trasmettere

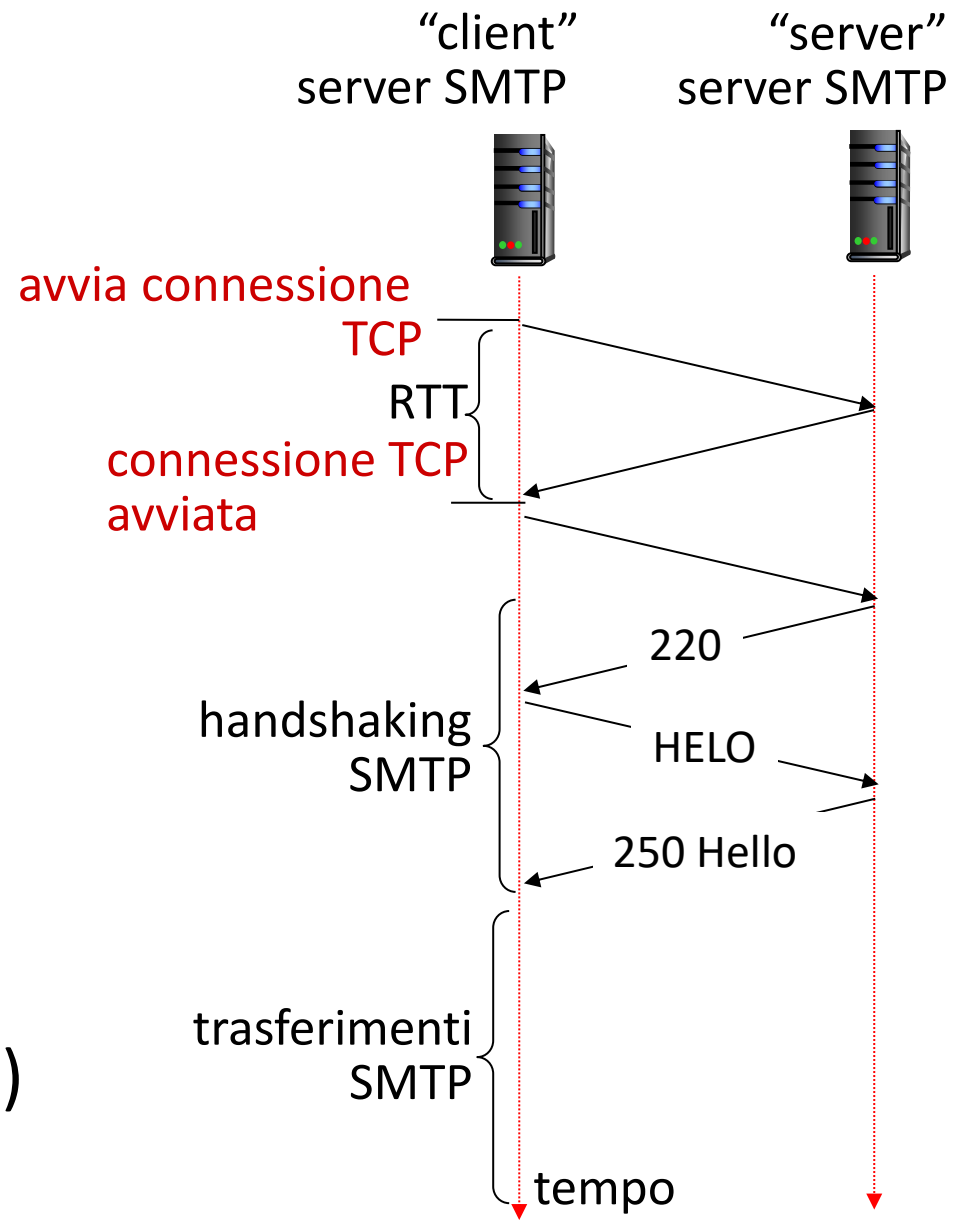
protocollo SMTP tra mail server per inviare messaggi email

- client: mail server trasmittente
- "server": mail server ricevente



SMTP RFC (5321)

- usa TCP per trasferire un modo affidabile i messaggi di posta elettronica dal client (mail server che avvia la connessione) al server, porta 25
  - Trasferimento diretto: dal server di posta del mittente al server di posta del destinatario
- Tre fasi per il trasferimento
  - handshaking (saluto)
  - trasferimento dei messaggi
  - chiusura
- Interazione comando/risposta (come HTTP)
  - comandi: testo ASCII a 7 bit
  - risposta: codice di stato e espressione



Scenario: Alice invia un'e-mail a Bob

1) Alice usa il suo user agent per comporre il messaggio da inviare “a” ("to") bob@someschool.edu

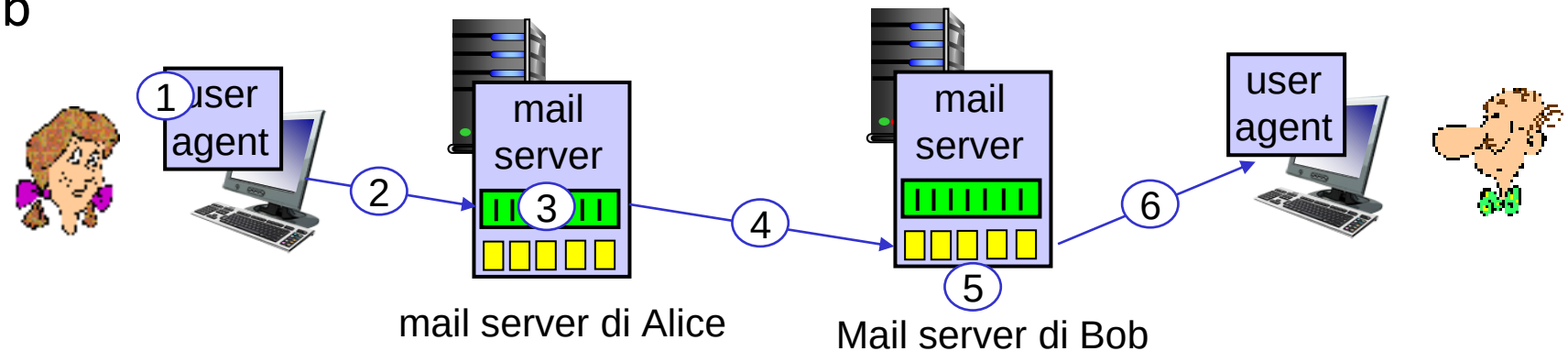
2) lo user agent di Alice invia un messaggio al server di posta di Alice; il messaggio è posto nella coda di messaggi

3) il lato client di SMTP apre una connessione TCP con il mail server di Bob

4) il client SMTP invia il messaggio di Alice sulla connessione TCP

5) il mail server di Bob pone il messaggio nella casella di posta di Bob

6) Bob invoca il suo user agent per leggere il messaggio



Esempio di interazione SMTP

```
S: 220 hamburger.edu
C: HELO crepes.fr
S: 250  Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection
```

SMTP: note finali

confronto con HTTP:

- HTTP: client pull
- SMTP: client push
- Entrambi hanno un'interazione comando/risposta in ASCII, codici di stato
- HTTP: ciascun oggetto è incapsulato nel suo messaggi di risposta
- SMTP: più oggetti vengono trasmessi in un unico messaggio
- SMTP usa connessione persistenti
- SMTP richiede che il messaggio (intestazione e corpo) sia nel formato ASCII a 7 bit
- Il server SMTP usa CRLF.CRLF per determinare la fine del messaggio

SMTP: note finali (cont)

dot-stuffing

Abbiamo visto che una riga contenente soltanto un punto segna la fine di un'email.

Come facciamo a trasmettere un'email nel cui corpo c'è un riga contente soltanto un punto?

SMTP prevede una forma di *escaping*, chiamata *dot-stuffing*:

- Il client invia due punti (cioè `..`) anziché un punto (`.`), quando questo si trova all'inizio di una riga
- Il server sostituisce ogni sequenza di due punti (`..`) all'inizio di una riga con un solo punto (`.`)

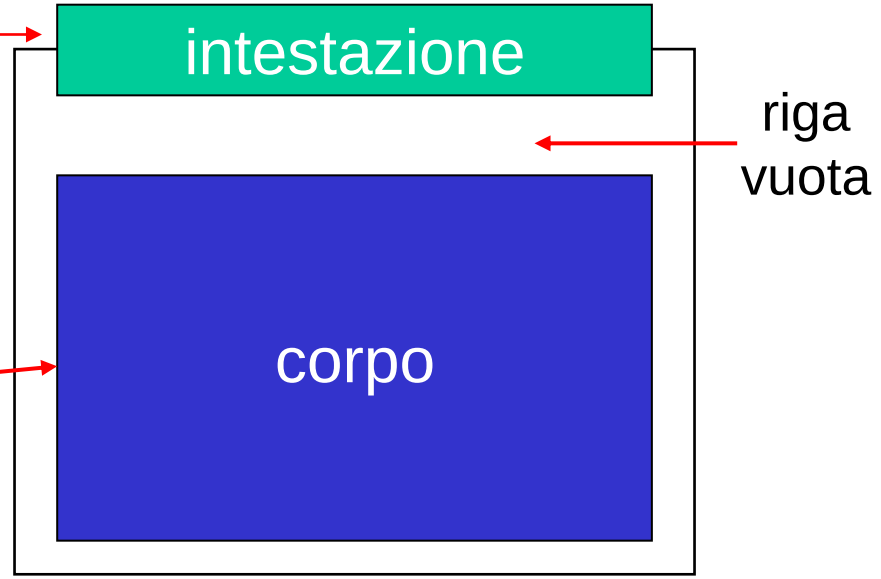
Formato dei messaggi di posta elettronica

SMTP: protocollo per scambiare messaggi di posta elettronica, definito nell'RFC 5321 (come RFC 7231 definisce HTTP)

RFC 2822 definisce la *sintassi* dei messaggi di posta elettronica (come HTML definisce la sintassi per i documenti web)

- Righe di intestazione, per esempio.,
  - To/A:
  - From/Da:
  - Subject/Oggetto:

  differenti da comandi SMTP MAIL FROM:, RCPT TO:!
- corpo: il “messaggio” , soltanto caratteri ASCII



RFC 2045 e RFC 2046 (Multipurpose Internet Mail Extensions (MIME)): inclusione di contenuti non testuali (come immagini, audio, video, documenti) e la suddivisione del messaggio in più parti.

MIME: esempio

```
MIME-Version: 1.0
Date: Fri, 21 Mar 2025 16:44:12 +0100
Message-ID: <CAGDmdGg2LN4R9bn-XT1Y8rGyTUTVEUtWvioM7QzBn7ESdMGBSA@mail.gmail.com>
Subject: Test MIME
From: Manuel Fiorelli <manuel.fiorelli@gmail.com>
To: Manuel Fiorelli <manuel.fiorelli@gmail.com>
Content-Type: multipart/related; boundary="000000000004288980630dc204d"

--000000000004288980630dc204d
Content-Type: multipart/alternative; boundary="000000000004288960630dc204c"

--000000000004288960630dc204c
Content-Type: text/plain; charset="UTF-8"
Content-Transfer-Encoding: quoted-printable

Che bell=E2=80=99immagine!

[image: Senza titolo.png]

--000000000004288960630dc204c
```

Specifica che le parti del messaggio sono "collegate" o correlate, ad esempio un documento HTML e le immagini incorporate in esso

Ogni parte è racchiusa da una coppia di delimitatori iniziale e finale

Prima parte (il messaggio di posta)

Sono disponibili diverse versioni alternative dello stesso contenuto

Messaggio in testo semplice

Codifica il contenuto in modo da rappresentare qualsiasi byte con ASCII a 7bit (=xx dove xx è la rappresentazione esadecimale del byte)

MIME: esempio (cont)

```
--0000000000004288960630dc204c
Content-Type: text/html; charset="UTF-8"
Content-Transfer-Encoding: quoted-printable

<div dir=3D"ltr"><div>Che bell=E2=80=99immagine!</div><div><br></div><div><=
img src=3D"cid:ii_m8iy9q3m0" alt=3D"Senza titolo.png" width=3D"2" height=3D=
"2"><br><br></div></div>

--0000000000004288960630dc204c--
--0000000000004288980630dc204d
Content-Type: image/png; name="Senza titolo.png"
Content-Disposition: attachment; filename="Senza titolo.png"
Content-Transfer-Encoding: base64
X-Attachment-Id: ii_m8iy9q3m0
Content-ID: <ii_m8iy9q3m0>

iVBORw0KGgoAAAANSUhEUgAAAAIAAAACCAYAAABytg0kAAAAAXNSR0IArs4c6QAAAARnQU1BAACx
jwv8YQUAAAAJcEhZcwAADsMAAA7DAcdvqGQAAAAQSURBVBhXY2BgYPgPAA
AElFTkSuQmCC
--0000000000004288980630dc204d--
```

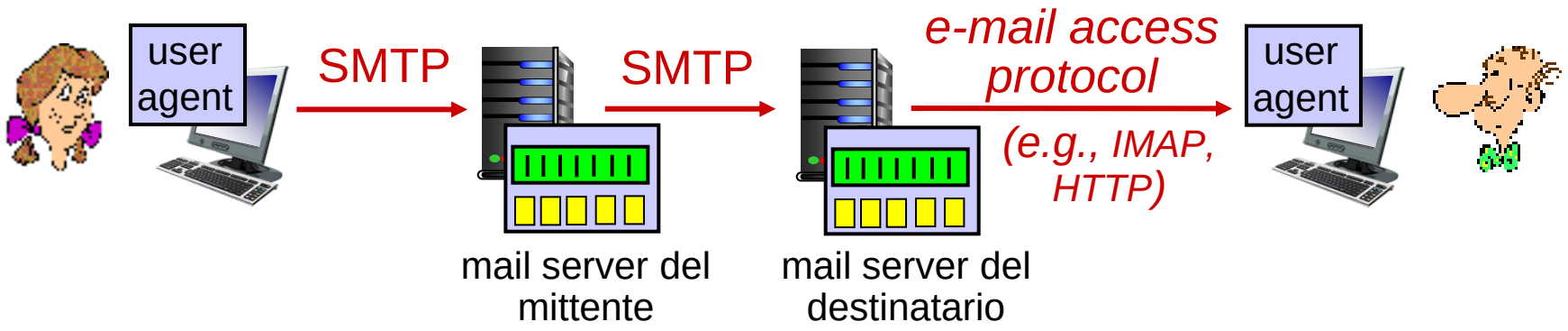
Versione HTML del messaggio

Lo schema cid: fa riferimento a una parte di un messaggio MIME attraverso il content id

=3D?
3D è il codice ASCII del carattere "=" in esadecimale (61 in decimale) [secondo la codifica quoted-printable indicata sopra occorre codificare tutti i caratteri, tranne i caratteri ASCII stampabili o i caratteri di fine riga (ma anche =)]

Codifica base64 di una immagine.,
Originale: 121 byte
Codificato: 168 byte

Protocolli di accesso alla posta



- SMTP: consegna/memorizzazione sul server del destinatario
- protocollo di accesso alla posta: ottenere i messaggi dal server
  - IMAP: Internet Mail Access Protocol [RFC 3501]: messaggi memorizzati sul server, IMAP consente di recuperare, cancellare e archiviare i messaggi memorizzati sul server.
- HTTP: gmail, Hotmail, Yahoo!Mail, etc. consente interfaccia web sopra a STMP (per l'invio) e IMAP (o POP) per il recupero delle email.

Livello di applicazione: panoramica

- Principi delle applicazioni di rete
- Web e HTTP
- E-mail, SMTP, IMAP
- DNS: il servizio di directory di Internet
- Applicazioni P2P
- Streaming video e reti di distribuzione di contenti
- Programmazione delle socket programming con UDP e TCP

Problema: risoluzione dei nomi

persone: molti identificatori:

- Nome, codice fiscale, numero della carta di identità

Host e router di Internet:

- indirizzo IP (32 bit) - usato per indirizzare i datagrammi
- "nome", ad esempio, cs.umass.edu - usato dagli esseri umani

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

Problema: risoluzione dei nomi

persone: molti identificatori:

- nome, codice fiscale, numero della carta di identità

Host e router di Internet:

- indirizzo IP (32 bit) - usato per indirizzare i datagrammi
- "nome", ad esempio, cs.umass.edu - usato dagli esseri umani

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

File *hosts* (/etc/hosts nei sistemi POSIX)

Associa un indirizzo IP a uno o più hostname

```
185.300.10.1   host1
185.300.10.2   host2 merlin
185.300.10.3   host3 arthur king
185.300.10.4   timeserver
```

Locale a un nodo, il suo contenuto non deve necessariamente coincidere con quello di altri nodi (ma meglio evitarlo!)

Problema: risoluzione dei nomi

persone: molti identificatori:

- nome, codice fiscale, numero della carta di identità

Host e router di Internet:

- indirizzo IP (32 bit) - usato per indirizzare i datagrammi
- "nome", ad esempio, cs.umass.edu - usato dagli esseri umani

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

Anni '70: HOSTS.TXT

- *Mantenuto dal Network Information Center (NIC) presso lo SRI*
- *Reso disponibile su un host designato* (attraverso il protocollo FTP)
- *Installato dagli amministratori di sistema sui singoli nodi*

Problemi:

- *Crescita del file*
- *Traffico generato sull'host dove era pubblicato*

Fonte: https://www.oreilly.com/library/view/dns-on-windows/0596005628/ch01s02s01.html

DNS: Domain Name System

persone: molti identificatori:

- nome, codice fiscale, numero della carta di identità

Host e router di Internet:

- indirizzo IP (32 bit) - usato per indirizzare i datagrammi
- "nome", ad esempio, cs.umass.edu - usato dagli esseri umani

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

Domain Name System (DNS):

- *Database distribuito* implementato in una gerarchia di *name server*
- *Protocollo a livello di applicazione* che consente agli host, ai router e ai server DNS di comunicare per *risolvere* i nomi (traduzione nome/indirizzo)
  - *si noti:* funzioni critiche di Internet, implementate come protocollo a livello di applicazione
  - complessità nelle parti periferiche della rete

DNS: servizi, struttura

Servizi DNS

- Traduzione degli hostname in indirizzi IP
- host aliasing
  - nome canonico e alias
- mail server aliasing
- load distribution (*distribuzione del carico di rete*)
  - server Web replicati: più indirizzi IP corrispondono a un solo nome

Q: Perché non centralizzare il DNS?

- un *single point of failure* (*punto di vulnerabilità*)
- volume di traffico
- database centralizzato distante
- manutenzione

R: non scala!

- Solo i server DNS di Comcast: 600B query DNS al giorno
- Solo i server DNS Akamai: 2,2T query DNS al giorno

Pensare al DNS

un enorme database distribuito:

- ~ miliardi di record, ciascuno semplice

gestisce molti *trilioni* di interrogazioni al giorno :

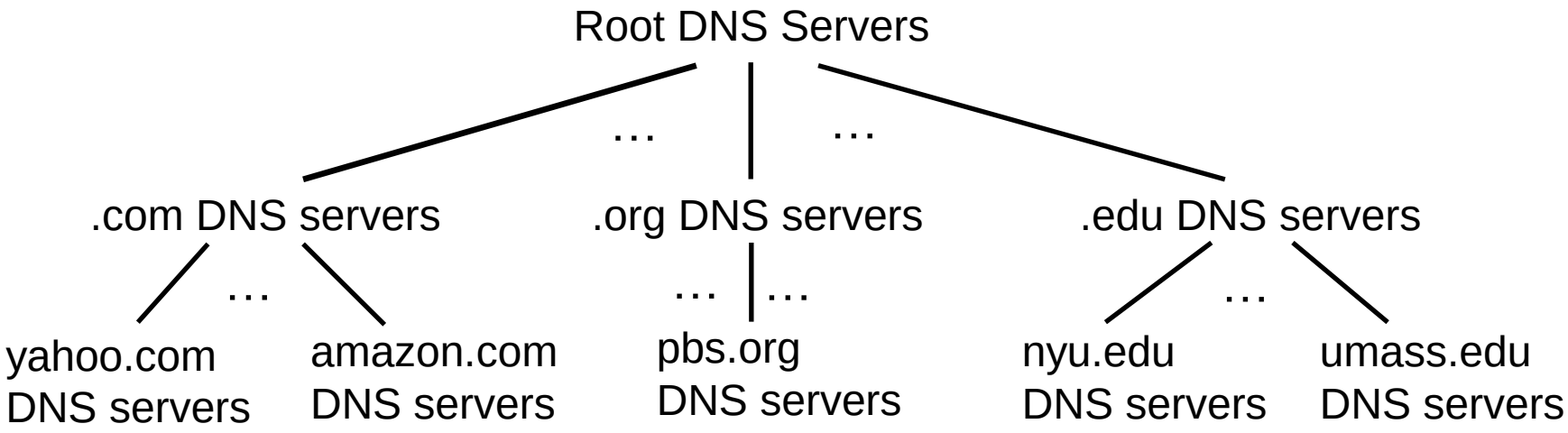
- *molte* più letture che scritture
- *è importante*: quasi tutte le transazioni Internet interagiscono con il DNS - i millisecondi contano!

organizzativamente e fisicamente decentralizzato

- milioni di organizzazioni diverse responsabili d loro *record*

"a prova di proiettile": affidabilità, sicurezza

DNS: un database distribuito e gerarchico



Root
(radice)

Top Level Domain

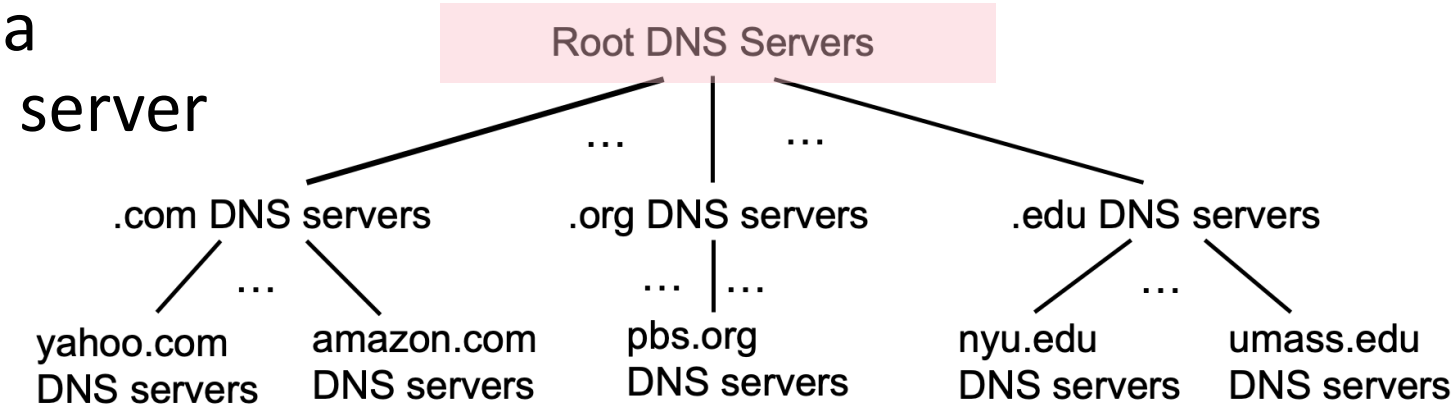
Authoritative
(server autoritativi)

Il client vuole l'indirizzo IP di www.amazon.com; 1ª approssimazione:

- il cliente interroga il root server per trovare il TLD server per .com
- il client interroga il TLD server .com per ottenere il server autoritativo per amazon.com
- il client interroga il server autoritativo per amazon.com per ottenere l'indirizzo IP di www.amazon.com

DNS: root name server

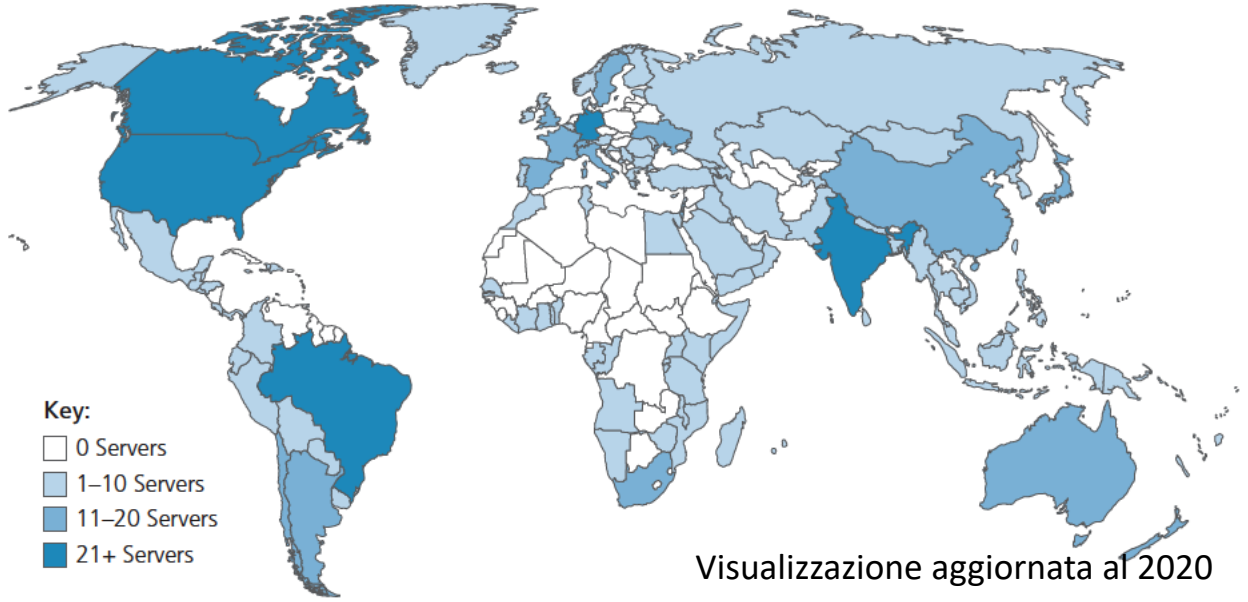
- ufficiale, contatto di ultima istanza da parte dei name server che non sono in grado di risolvere il nome



DNS: root name server

- ufficiale, contatto di ultima istanza da parte dei name server che non sono in grado di risolvere il nome.
  Fornisce gli indirizzi IP dei TLD server
- funzione *incredibilmente importante* di Internet
  - Internet non potrebbe funzionare senza!
  - DNSSEC – offre sicurezza (autenticazione, integrità dei messaggi)
- ICANN (Internet Corporation for Assigned Names and Numbers) gestisce il root DNS domain

13 name "server" logici in tutto il mondo, ogni "server" replicato più volte (~200 server negli USA)
https://www.internic.net/domain/named.root



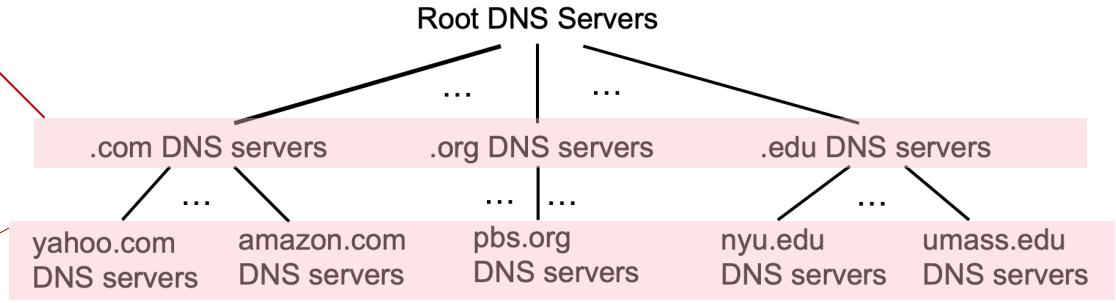
Visualizzazione aggiornata al 2020

Il 20/03/2023 ci sono 1813 istanze gestite da 12 operatori, coordinate dallo IANA (fonte: https://root-servers.org/)

Top-Level Domain, and authoritative servers

Top-Level Domain (TLD) DNS server:

- si occupano dei domini .com, .org, .net, .edu, .aero, .jobs, .museums, e di tutti I domani locali di alto livello, quali .cn, .uk, .fr, .ca, .jp
- Network Solutions: gestisce i server TLD per I domini .com e .net
- Educause: gestisce quelli per .edu



DNS server autoritativo:

- server DNS propri di ciascuna organizzazione, che forniscono i mapping ufficiali da hostname a IP per gli host dell'organizzazione
- possono essere mantenuti dall'organizzazione o dal service provider

Name server DNS locali

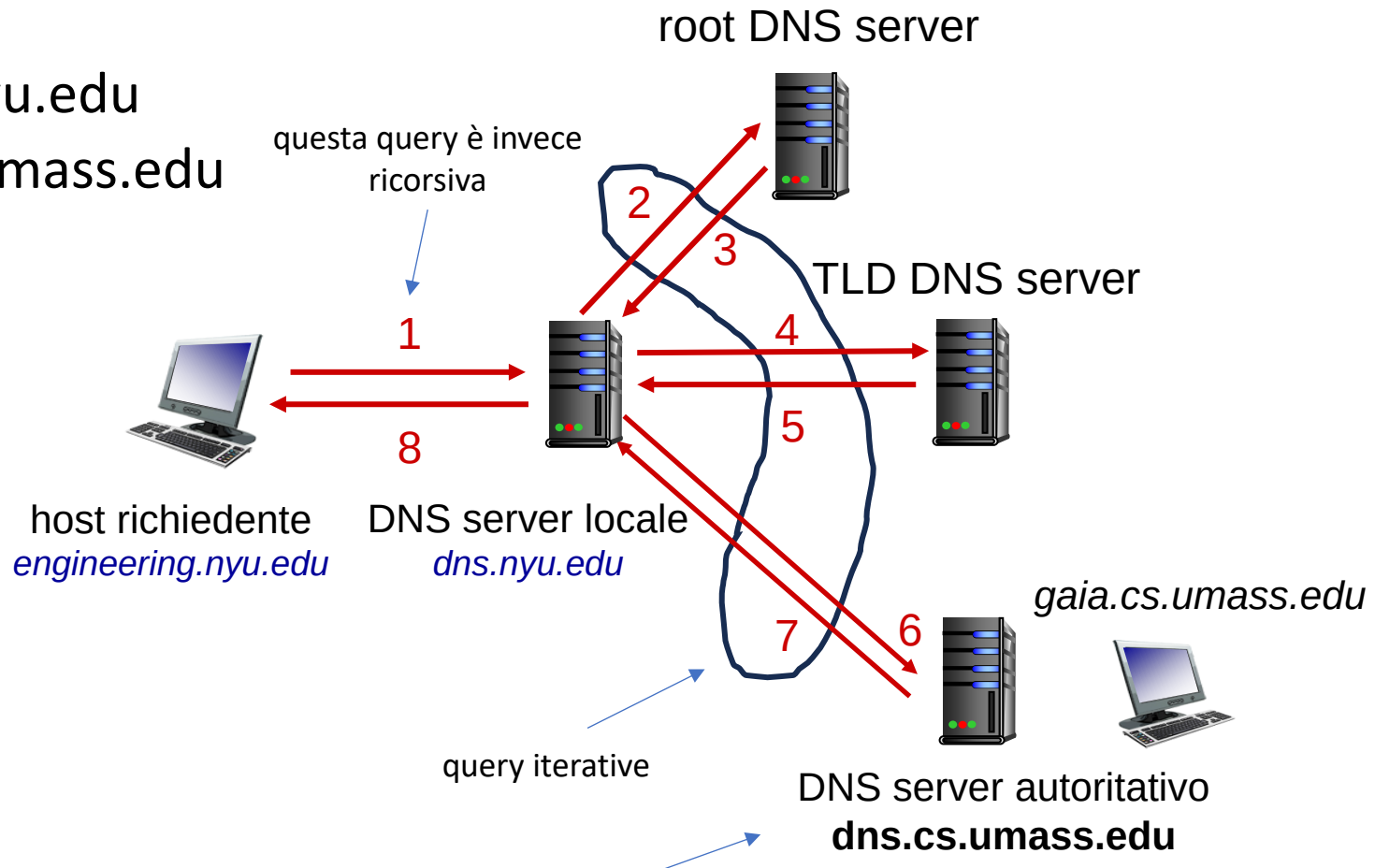
- quando l'host effettua una richiesta DNS, la query viene inviata al suo server DNS *locale* (con funzione di *default name server*)
  - il server DNS locale restituisce una risposta, rispondendo:
    - dalla sua cache locale di coppie nome->indirizzo (possibilmente non aggiornate!)
    - inoltrando la richiesta alla gerarchia DNS per la risoluzione
  - ciascun ISP ha un proprio server DNS locale; per trovare il vostro:
    - MacOS: `% scutil --dns`
    - Windows: `>ipconfig /all`
- il server DNS locale non appartiene strettamente alla gerarchia dei server

DNS: interrogazione iterativa

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

Query iterativa

- Il server contattato risponde con il nome del server da contattare
- "Io non conosco questo nome, ma puoi chiederlo a questo server"



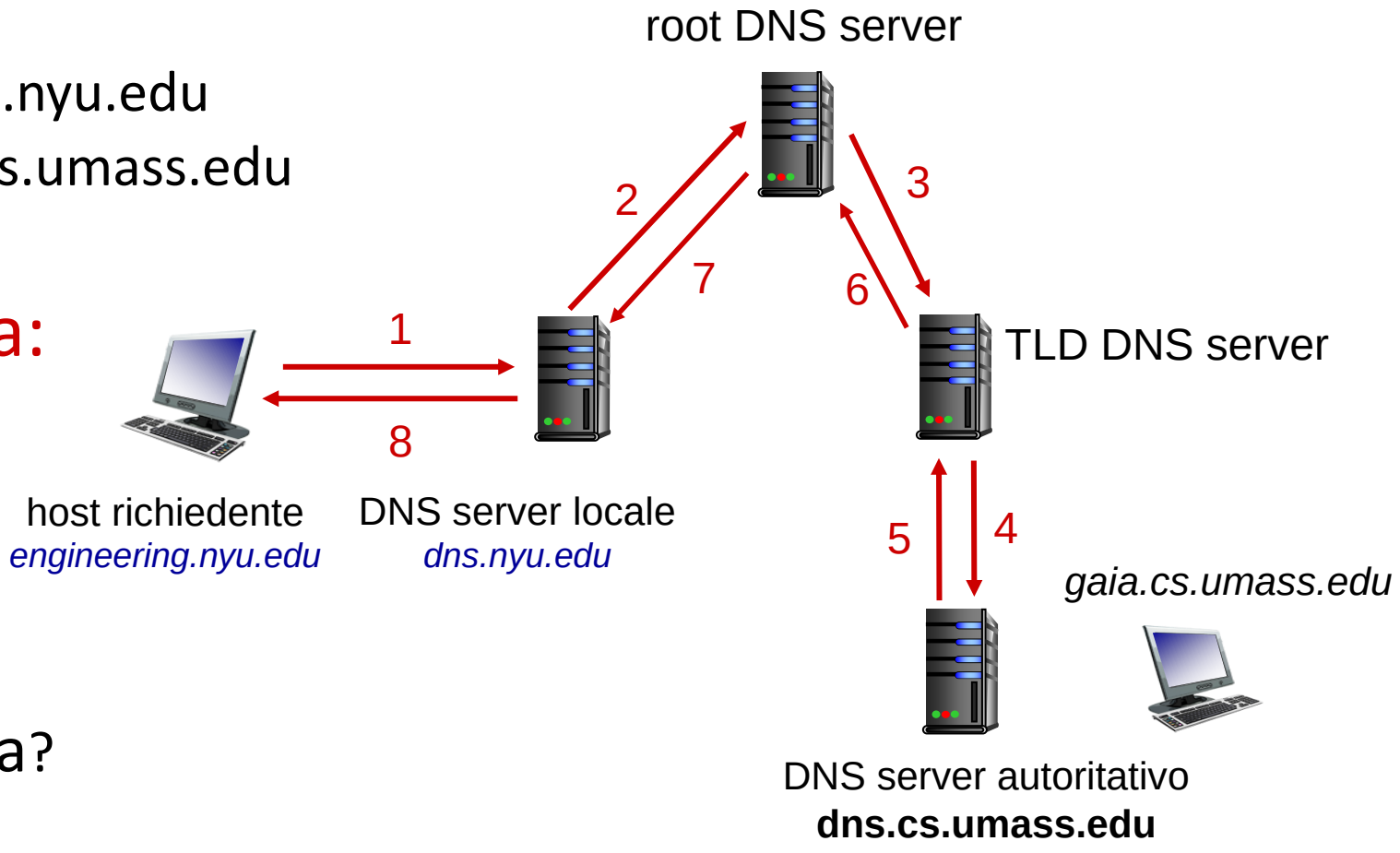
il TLD DNS server potrebbe conoscere in realtà un DNS server intermedio, ad esempio nel caso di domini di terzo livello aventi ciascuno il proprio DNS server autoritativo

DNS: interrogazione ricorsiva

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

Interrogazione ricorsiva:

- Affida il compito di tradurre il nome al server contattato
- carico pesante ai livelli superiori della gerarchia?



DNS: caching e aggiornamento dei record

- una volta che un (qualsiasi) name server impara la mappatura, la mette nella *cache*, e restituisce *immediatamente* il mapping nella cache in risposta a un query
  - il caching migliora i tempi di risposta
  - le voci della cache vanno in timeout (scompaiono) dopo un certo tempo (TTL)
  - i server TLD sono in genere memorizzati nella cache dei server dei nomi locali
- le voci nella cache potrebbero essere *obsolete*
  - se l'host con nome cambia il suo indirizzo IP, potrebbe non essere conosciuto su Internet fino alla scadenza di tutti i TTL!
  - *traduzione nome->indirizzo best-effort!*

Record DNS

DNS: database distribuito che memorizza record di risorsa (RR)

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

type=A

- `name` è l'hostname
- `value` è l'indirizzo IP

type=NS

- `name` è il dominio (ad esempio, foo.com)
- `value` è l'hostname dell'authoritative name server per questo dominio

type=CNAME

- `name` è il nome alias di qualche nome "canonico" (nome vero)
- www.ibm.com è in realtà servereast.backup2.ibm.com
- `value` è il nome canonico

type=MX

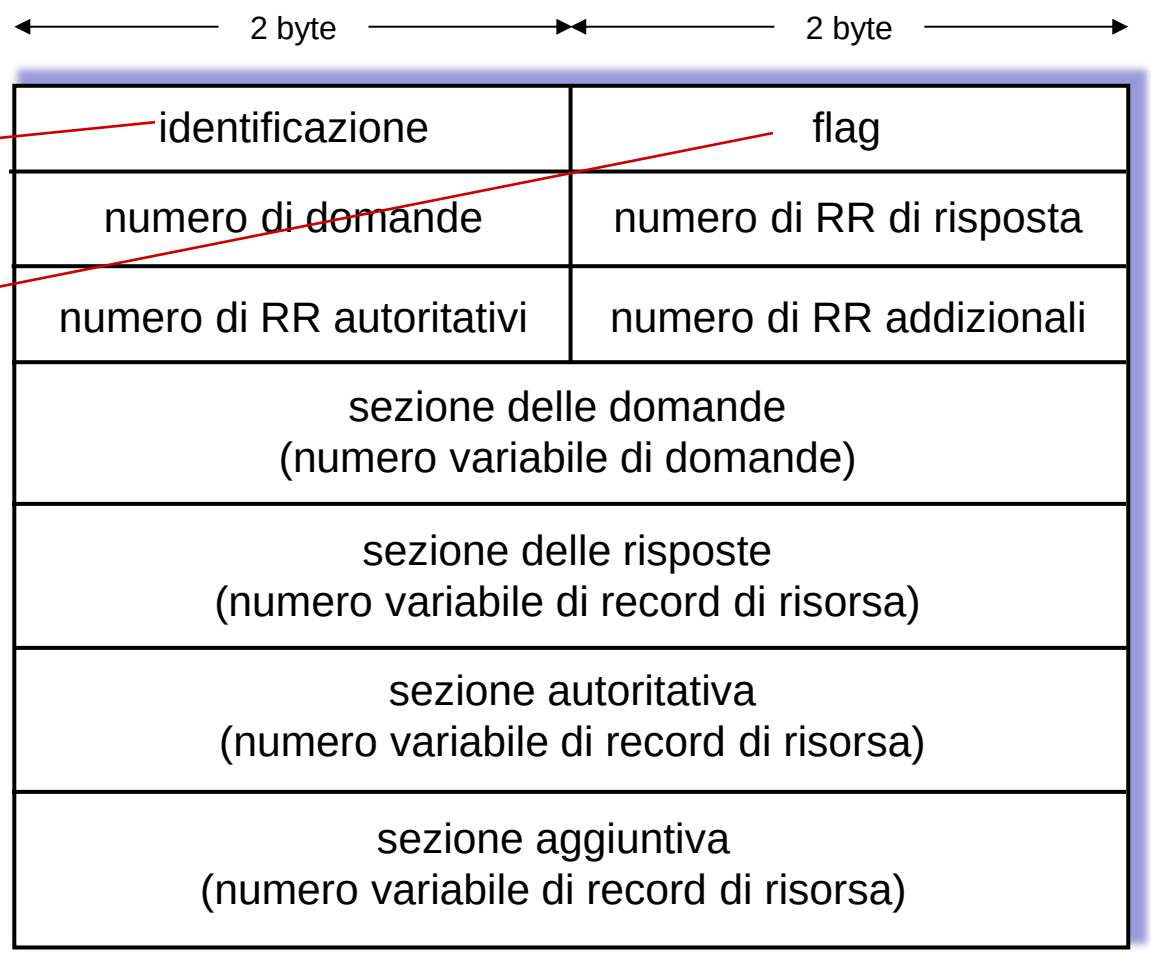
- `value` è il nome del server di posta associato a `name`

Messaggi DNS

domande (query) e messaggi di *risposta* (reply), entrambi con lo stesso formato:

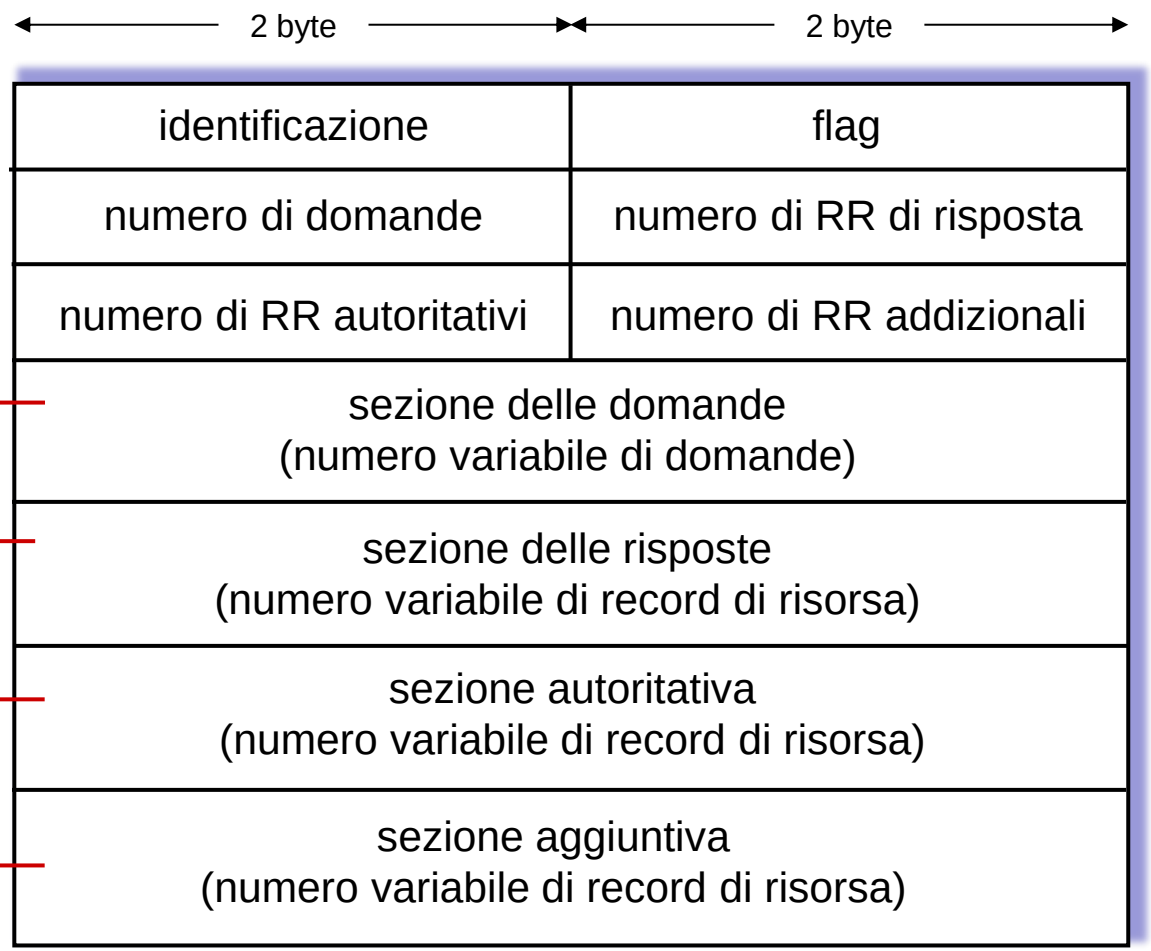
Intestazione del messaggio:

- identificazione: numero di 16 bit per la domanda; la risposta alla domanda usa lo stesso numero
- flag:
  - domanda o risposta
  - richiesta di ricorsione
  - ricorsione disponibile
  - DNS server autoritativo



Messaggi DNS

domande (query) e messaggi di *risposta* (reply), entrambi con lo stesso formato:



campi per il nome richiesto e il tipo di domanda

RR nella risposta alla domanda

record per i server autoritativi (*referral* verso nameserver di livello più basso; vedi RFC 9471: https://datatracker.ietf.org/doc/rfc9471/)

informazioni extra che possono essere usate (es. se la risposta ad una richiesta di tipo MX contiene un hostname, può essere fornita qui la sua traduzione in IP)

Inserire record nel database DNS

Esempio: abbiamo appena avviato la nuova società “Network Utopia”

- Registriamo il nome networkuptopia.com presso il *DNS registrar* (ad esempio, Network Solutions, oppure un altro dei concorrente accreditati dall'ICANN)
  - forniamo al registrar il nome e gli indirizzi IP degli authoritative name server (primario e secondario)
  - il registar inserisce due RR nel TLD server .com:

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

    `(dns1.networkutopia.com, 212.212.212.1, A)`
- Inseriamo localmente nell'authoritative server
  - un record A per www.networkuptopia.com
  - un record MX per networkutopia.com

Sicurezza del DNS

Attacchi DDoS (distributed denial-of-service)

- bombare di traffico di root server
  - finora senza successo
  - filtraggio del traffico
  - I server DNS locali mantengono in cachi gli indirizzi IP dei server TLD, consentendo di aggirare i root server
- bombardare i server TLD
  - potenzialmente più pericoloso

Attacco di spoofing

- intercettare le query DNS, restituendo risposte fasulle
  - DNS cache poisoning
  - RFC 4033: DNSSEC - servizi di autenticazione