

Reti di Calcolatori:
Internet, Intranet e Mobile Computing
a.a. 2007/2008

<http://www.di.uniba.it/~lisi/courses/reti/reti0708.htm>

dott.ssa Francesca A. Lisi
lisi@di.uniba.it

Orario di ricevimento: mercoledì ore 10-12

Sommario della lezione di oggi: Lo strato di applicazione (3/3)

- ❑ Principi dei protocolli dello strato di applicazione
- ❑ World Wide Web & HTTP
- ❑ Trasferimento di file & il protocollo FTP
- ❑ Posta elettronica & SMTP
- ❑ DNS: il servizio directory di Internet
- ❑ Condivisione di file

Domain Name System (DNS)

Persone: molti
identificatori:

- codice fiscale, nome, nro passaporto

Host e router Internet:

- indirizzo IP (32 bit) - usato per indirizzare i datagrammi
- nome simbolico, p.es., gaia.cs.umass.edu - usato dagli umani

D: corrispondenza fra
indirizzi IP e nomi?

Domain Name System:

- *database distribuito* implementato in una gerarchia di molti *server*
- *protocollo* per la risoluzione (cioè traduzione) dei nomi di dominio (funzione core di Internet)
 - complessità ai "confini" della rete
 - Poggia su UDP
 - Server su porta 53

DNS

Servizi DNS

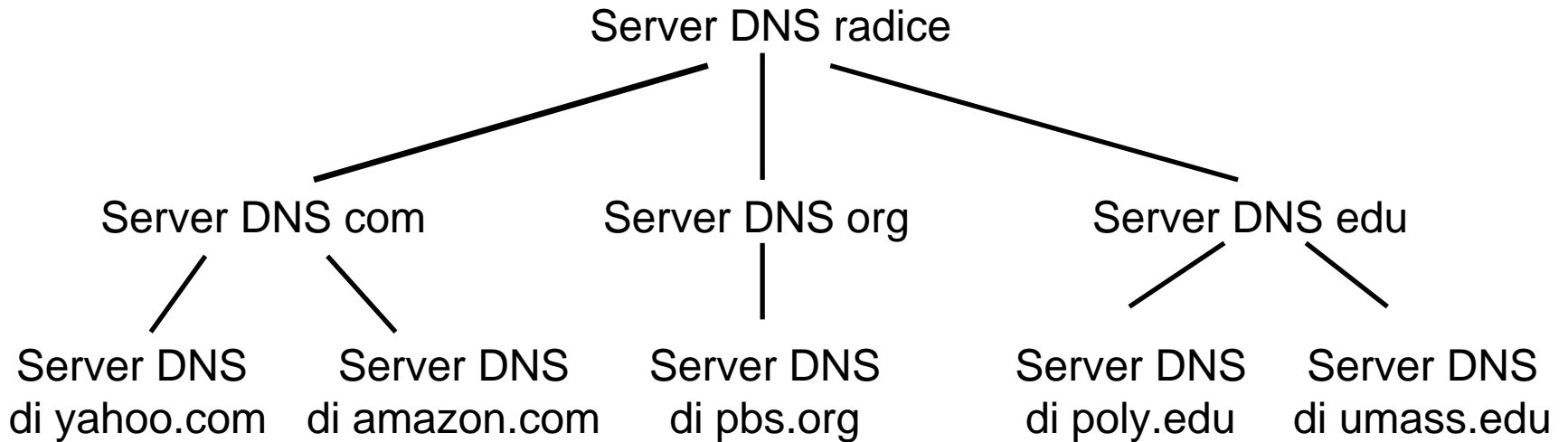
- ❑ Traduzione degli hostname in indirizzi IP
- ❑ Host aliasing
 - un host può avere più nomi
- ❑ Mail server aliasing
- ❑ Distribuzione locale
 - server web replicati: insieme di indirizzi IP per un nome canonico

Perché non centralizzare il DNS?

- ❑ singolo punto di guasto
- ❑ volume di traffico
- ❑ database centralizzato distante
- ❑ manutenzione

Un database centralizzato su un singolo server DNS non è *scalabile*!

Database distribuiti e gerarchici

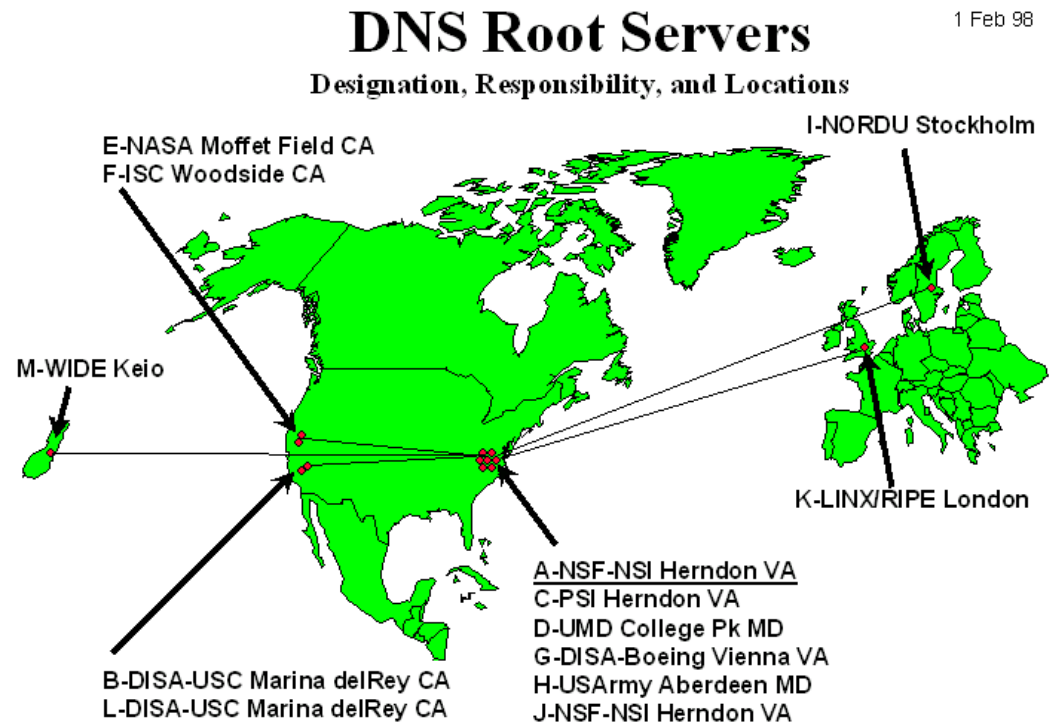


Il client vuole l'IP di www.amazon.com; 1ª approssimazione:

- ❑ Il client interroga il server radice per trovare il server DNS com
- ❑ Il client interroga il server DNS com per ottenere il server DNS amazon.com
- ❑ Il client interroga il server DNS amazon.com per ottenere l'indirizzo IP di www.amazon.com

DNS: i server radice

- Il server DNS locale contatta un server DNS radice se non riesce a risolvere un nome
- Il server DNS radice:
 - contatta un server DNS autorizzato se non conosce il mapping del nome
 - preleva il mapping
 - restituisce il mapping al server DNS locale
- 13 server DNS radice nel mondo



DNS:

server TLD e server assoluti

- **Server TLD (top-level domain):** si occupano dei domini com, org, net, edu, ecc. e di tutti i domini locali di alto livello, quali uk, fr, ca e jp.
 - Network Solutions gestisce i server TLD per il dominio com
 - Educause gestisce quelli per il dominio edu
- **Server di competenza (authoritative server):** ogni organizzazione dotata di host Internet pubblicamente accessibili (quali i server web e i server di posta) deve fornire i record DNS di pubblico dominio che mappano i nomi di tali host in indirizzi IP.
 - possono essere mantenuti dall'organizzazione o dal service provider

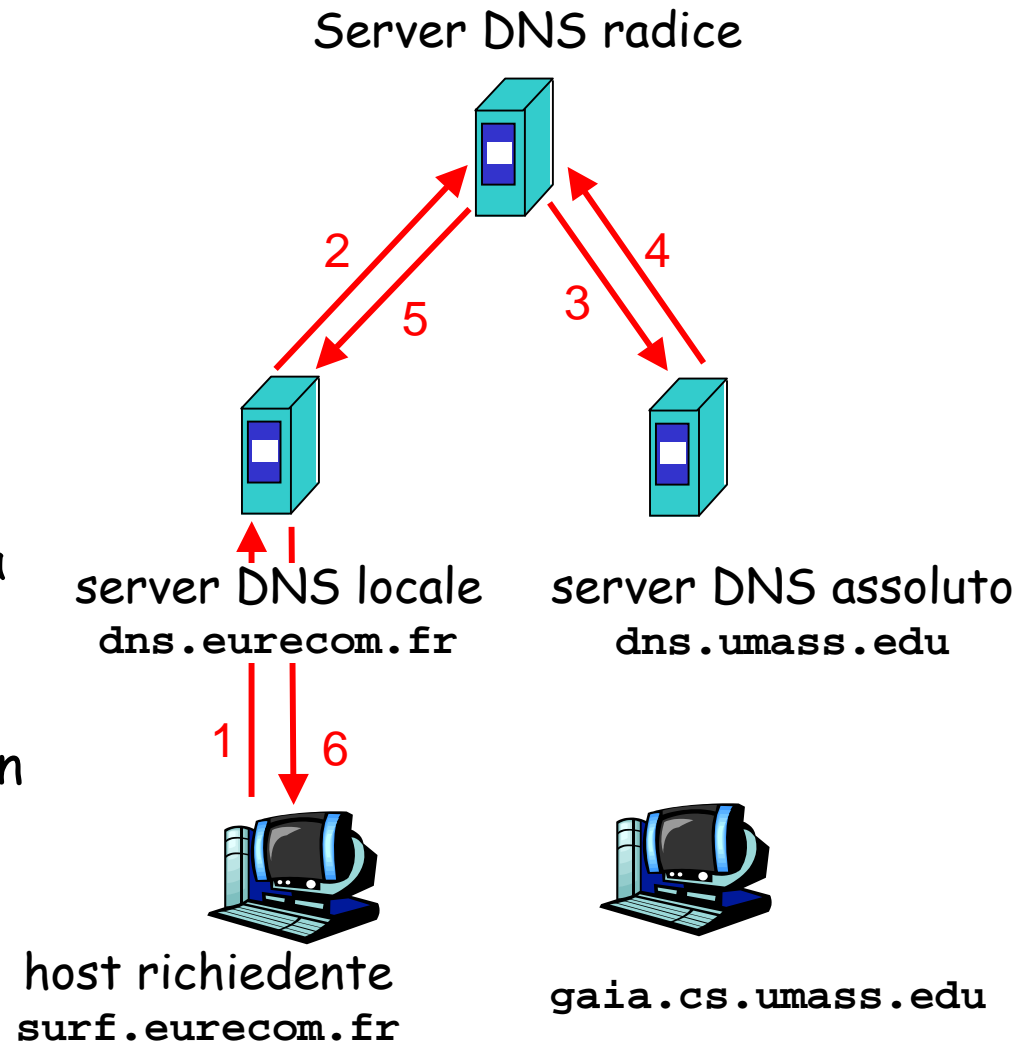
DNS: server locale

- ❑ Non appartiene strettamente alla gerarchia dei server
- ❑ Ciascun ISP (università, società, ISP residenziale) ha un server DNS locale.
 - detto anche "default name server"
- ❑ Quando un host effettua una richiesta DNS, la query viene inviata al suo server DNS locale
 - il server DNS locale opera da proxy e inoltra la query in una gerarchia di server DNS

DNS: un esempio

L'host `surf.eurecom.fr`
richiede indirizzo IP di
`gaia.cs.umass.edu`

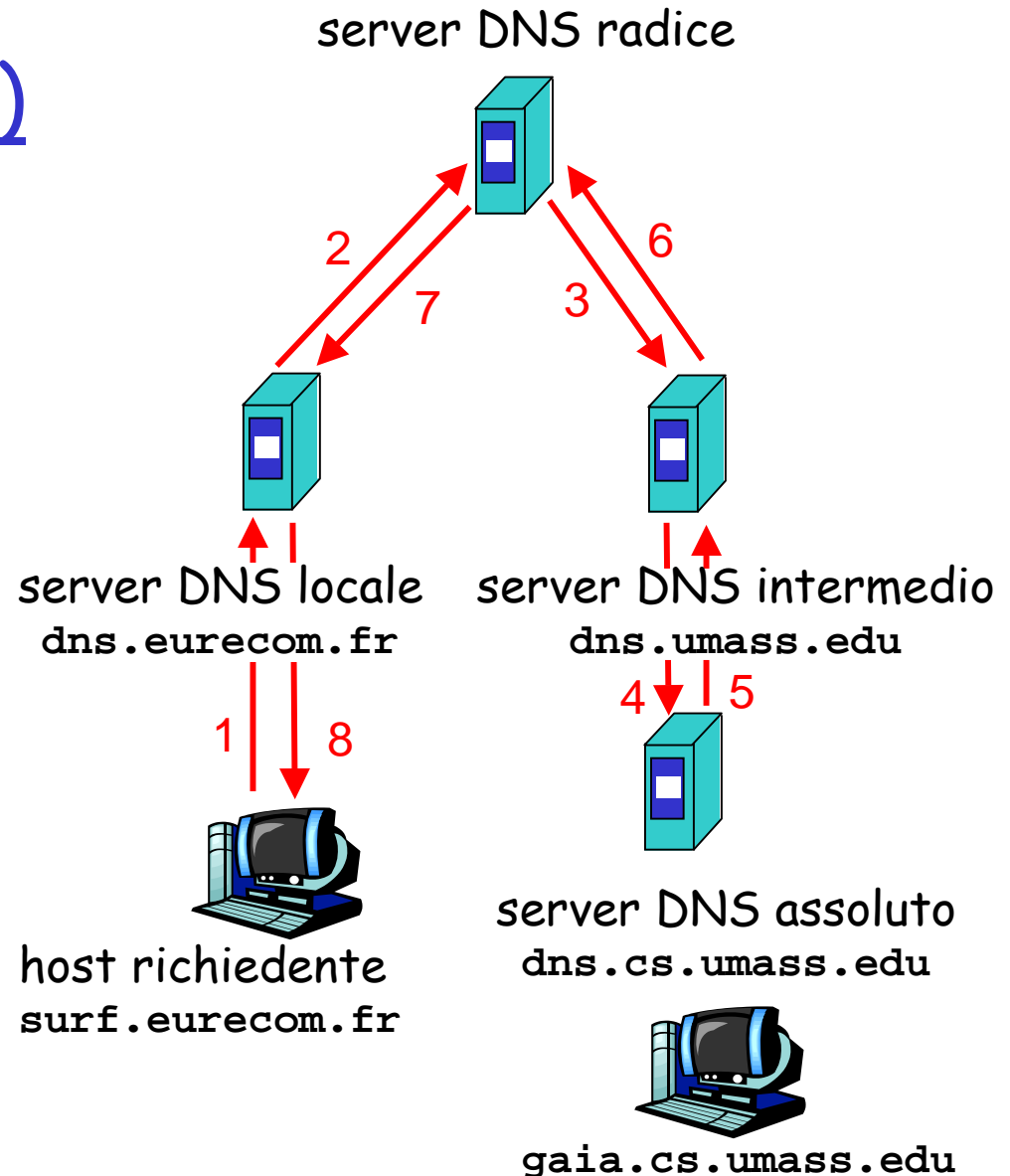
1. Contatta il suo server DNS locale, `dns.eurecom.fr`
2. `dns.eurecom.fr` contatta un server DNS radice, se necessario
3. Il server radice contatta un server DNS assoluto, `dns.umass.edu`, se necessario



DNS: un esempio (cont.)

Un server radice:

- potrebbe non conoscere un server DNS assoluto
- potrebbe invece conoscere un *server DNS intermedio o top-level domain (TLD)*, ovvero chi contattare per trovare un server DNS assoluto



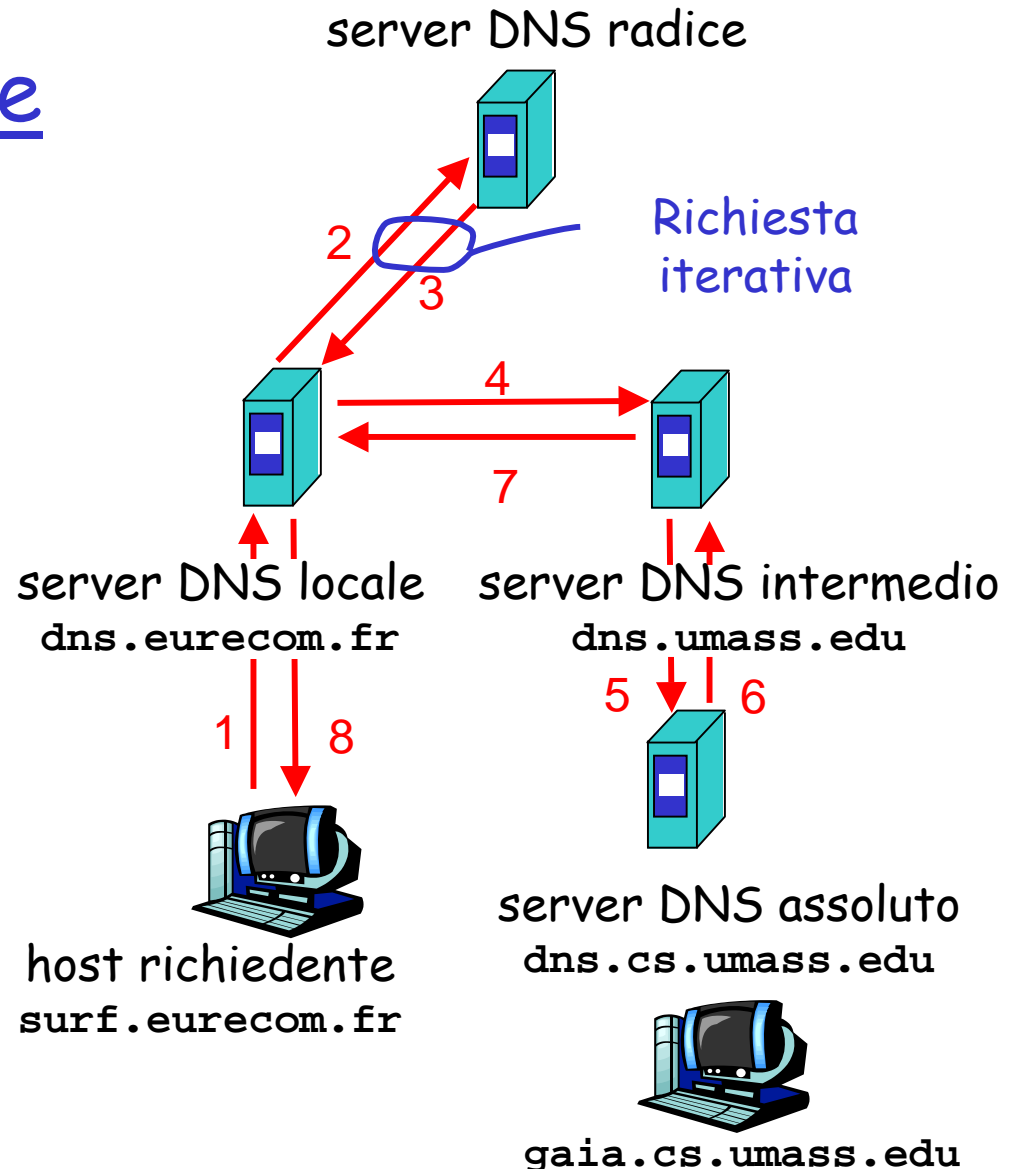
DNS: richieste iterative

richiesta ricorsiva:

- pone il carico della risoluzione dei nomi sul server DNS contattato
- carico pesante?

richiesta iterativa:

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



DNS:

caching e record di aggiornamento

- Una volta che un (qualsiasi) server dei nomi apprende un mapping, lo pone in *cache*
 - le entry della cache scadono (scompaiono) dopo un certo lasso di tempo
- meccanismi di aggiornamento/notifica in fase di progettazione presso IETF
 - RFC 2136
 - <http://www.ietf.org/html.charters/dnsind-charter.html>

DNS: record di risorsa

DNS: db distribuito contenente i **record di risorsa (RR)**

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

□ Type=A

- name host
- value indirizzo IP

□ Type=NS

- name dominio (p.es. foo.com)
- value indirizzo IP di un server assoluto dei nomi per questo dominio

□ Type=CNAME

- name alias per qualche nome "canonico" (quello vero)
- value nome canonico

□ Type=MX

- value nome di host ospitante un mailserver associato a name

DNS: inserimento dei record di risorsa

- ❑ Esempio: abbiamo appena avviato la nuova società "Network Utopia"
- ❑ Registriamo il nome `networkutopia.com` presso **registrar** (ad esempio, Network Solutions)
 - Forniamo a registrar i nomi e gli indirizzi IP dei server DNS di competenza (primario e secondario)
 - Registrar inserisce due RR nel server TLD com:

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

- ❑ Inseriamo nel server di competenza un record tipo A per `www.networkutopia.com` e un record tipo MX per `networkutopia.com`
- ❑ **In che modo gli utenti otterranno l'indirizzo IP del nostro sito web?**

DNS: messaggi del protocollo

Protocollo DNS: messaggi di *richiesta e risposta*, entrambi con lo stesso *formato di messaggio*

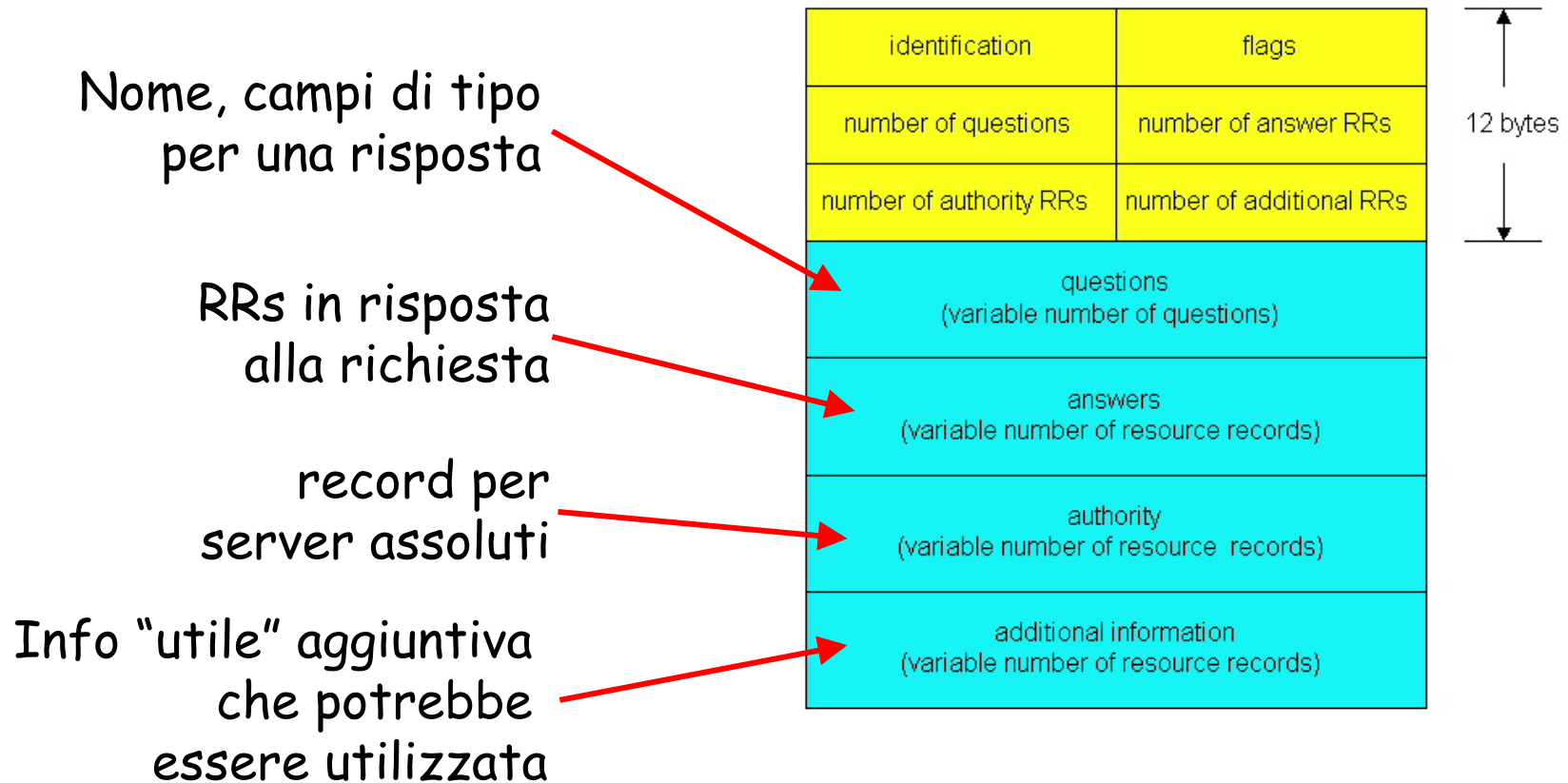
Intestazione msg

- **identificazione:** codice a 16 bit per richiesta, utilizzato anche dalla risposta
- **flags:**
 - richiesta o risposta
 - ricorsione desiderata
 - ricorsione disponibile
 - risposta assoluta

| | |
|---|--------------------------|
| identification | flags |
| number of questions | number of answer RRs |
| number of authority RRs | number of additional RRs |
| questions (variable number of questions) | |
| answers (variable number of resource records) | |
| authority (variable number of resource records) | |
| additional information (variable number of resource records) | |



DNS: messaggi del protocollo (cont.)



Condivisione di file P2P

Esempio

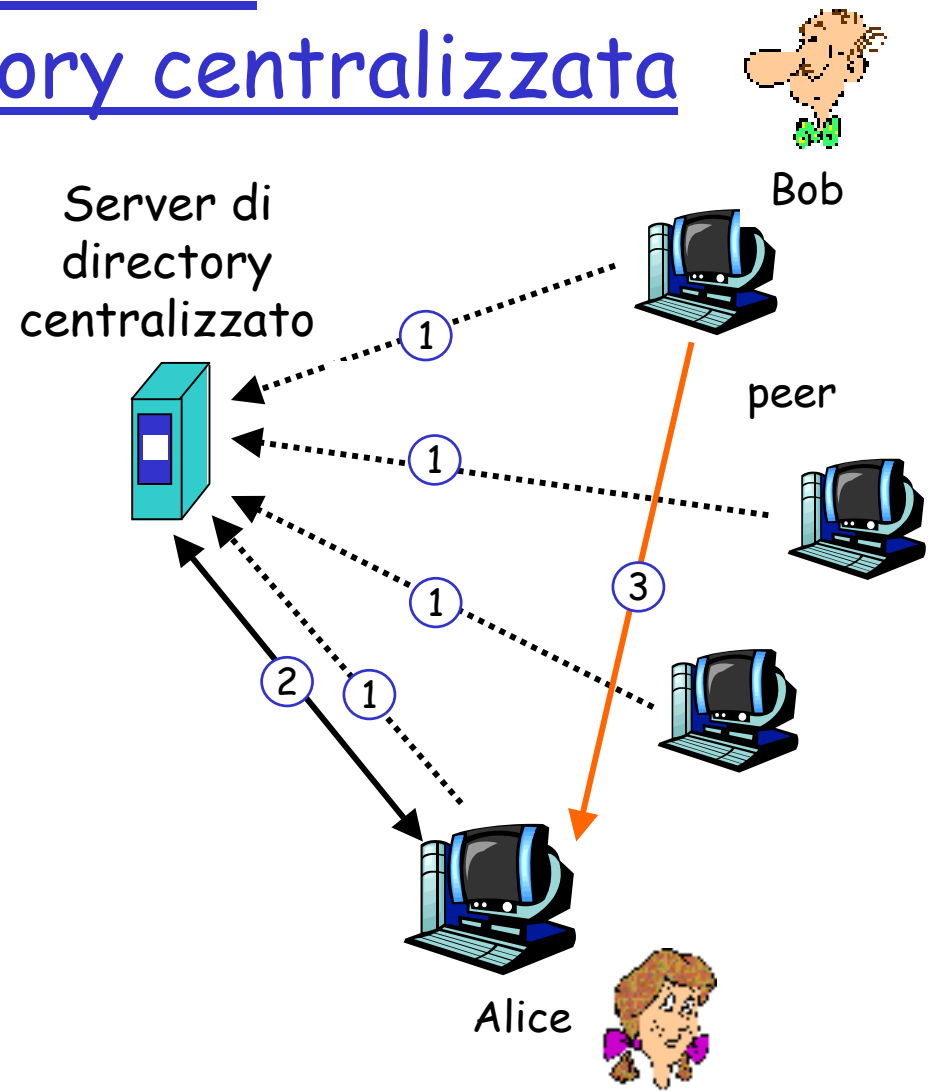
- Alice esegue un'applicazione di condivisione file P2P sul suo notebook
- Si collega in modo intermittente a Internet; ottiene un nuovo indirizzo IP ogni volta che si collega
- Cerca la canzone intitolata "Hey Jude"
- L'applicazione visualizza altri peer che hanno una copia di "Hey Jude"
- Alice sceglie uno dei peer, Bob
- Il file viene inviato dal PC di Bob al notebook di Alice: HTTP
- Mentre Alice scarica il file, altri utenti potrebbero scaricare dei file da Alice
- Il peer di Alice è sia client web sia server web transitorio

Tutti i peer sono server = grande scalabilità!

Condivisione di file P2P: soluzione con directory centralizzata

Progetto originale di
"Napster"

- 1) quando il peer si collega, informa il server centrale:
 - indirizzo IP
 - contenuto
- 2) Alice cerca la canzone "Hey Jude"
- 3) Alice richiede il file a Bob



Condivisione di file P2P: problemi con la directory centralizzata

- ❑ Unico punto di guasto
- ❑ Collo di bottiglia per le prestazioni
- ❑ Violazione del diritto d'autore

Il trasferimento dei file è distribuito, ma il processo di localizzazione è fortemente centralizzato

Condivisione di file P2P: soluzione con query flooding

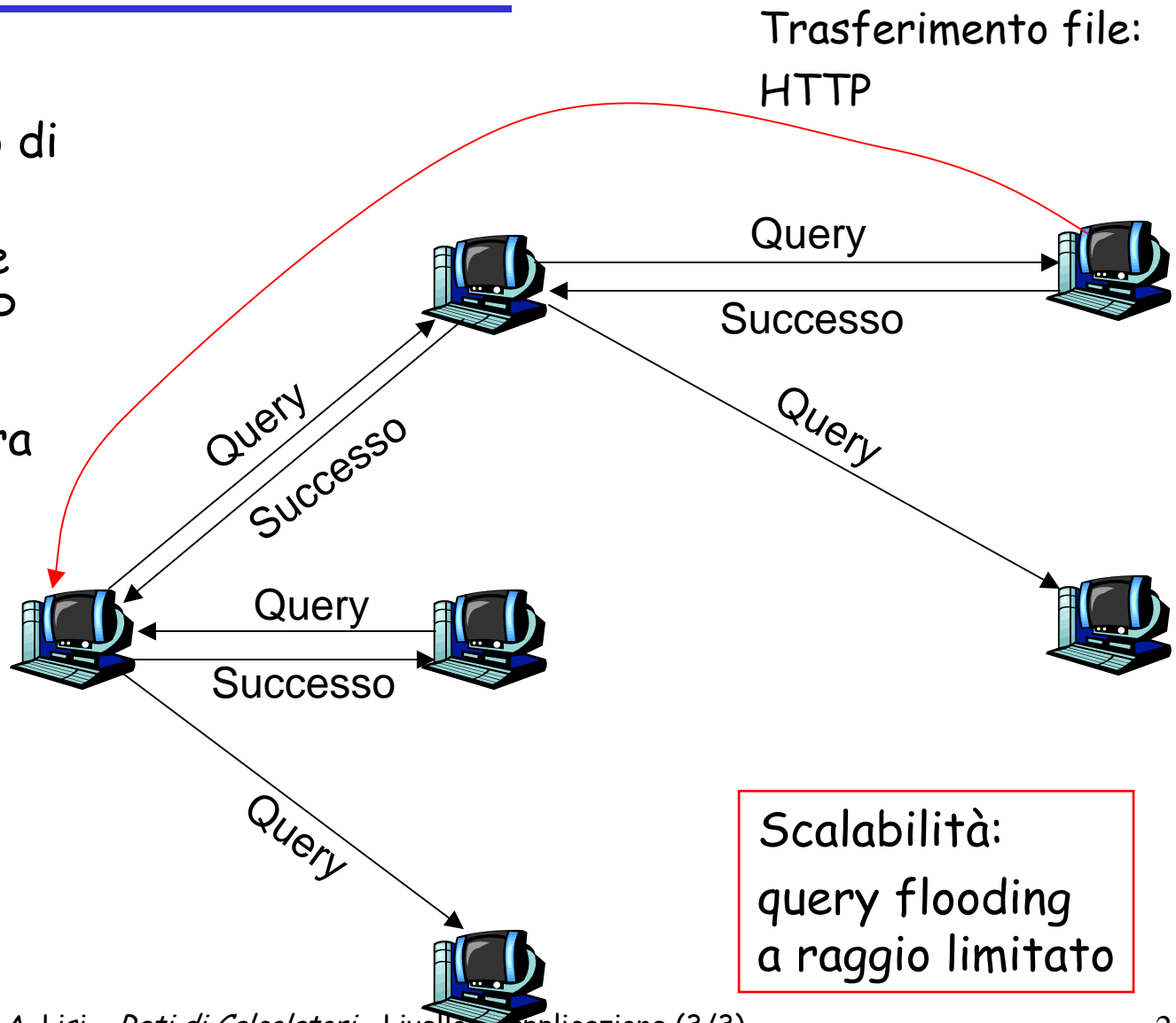
- ❑ **Es. Gnutella**
- ❑ Completamente distribuito
 - nessun server centrale
- ❑ Protocollo di pubblico dominio
- ❑ Molti client Gnutella implementano il protocollo

Rete di copertura

- ❑ Rete logica ed astratta basata sulla teoria dei grafi
- ❑ Arco tra i peer X e Y se c'è una connessione TCP
- ❑ Tutti i peer attivi e gli archi formano la rete di copertura
- ❑ Un arco non è un collegamento fisico
- ❑ Un dato peer sarà solitamente connesso con meno di 10 peer vicini nella rete di copertura

Condivisione di file P2P: il protocollo di Gnutella

- ❑ Il messaggio di richiesta è trasmesso sulle connessioni TCP esistenti
- ❑ Il peer inoltra il messaggio di richiesta
- ❑ Il messaggio di successo è trasmesso sul percorso inverso



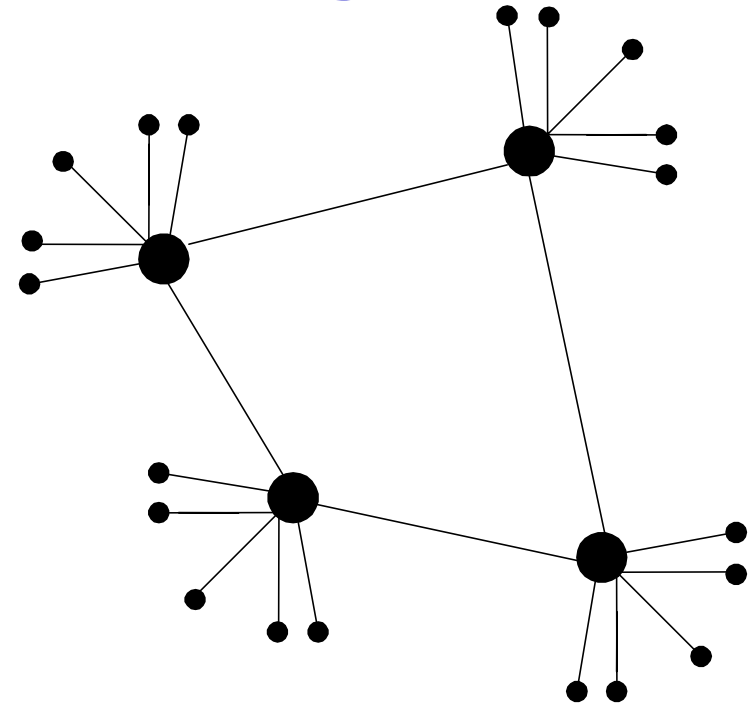
Condivisione di file P2P: inserimento dei peer in Gnutella

1. Il peer X deve innanzitutto trovare qualche altro peer che faccia parte della rete di copertura (problema del bootstrap -> soluzione con lista dei peer candidati)
2. X tenta in sequenza di impostare una connessione TCP con i peer della lista finché non stabilisce una connessione con un certo Y
3. X invia un messaggio Ping a Y; Y inoltra il messaggio Ping
4. Tutti i peer che ricevono il messaggio Ping rispondono con un messaggio Pong
5. X riceve molti messaggi Pong. Quindi può impostare delle connessioni TCP aggiuntive

Per il distacco dei peer vedi problema 16 a pag. 152!

Condivisione di file P2P: soluzione basata su peer eterogenei

- Ogni peer è un leader di gruppo o è assegnato a un leader di gruppo
 - Connessione TCP tra peer e il suo leader di gruppo
 - Connessioni TCP tra qualche coppia di leader di gruppo
- Il leader di gruppo tiene traccia del contenuto di tutti i suoi figli.



● Peer ordinario

● Peer leader di gruppo

— Relazioni di adiacenza
nella rete di copertura

Condivisione di file P2P: query in KaZaA

- ❑ Ogni file ha un identificatore hash e un descrittore
- ❑ Il client invia al suo leader di gruppo una query con una parola chiave
- ❑ Il leader di gruppo risponde con un elenco di peer che condividono i file i cui descrittori corrispondono alle parole chiave:
 - Per ogni corrispondenza: metadata, hash, indirizzo IP
- ❑ Se il leader di gruppo inoltra la query ad altri leader di gruppo, questi rispondono con le corrispondenze
- ❑ Il client quindi seleziona i file per il downloading
 - Le richieste HTTP che usano un identificatore hash sono trasmesse ai peer che hanno il file desiderato

Condivisione di file P2P: tecniche in KaZaA

- ❑ Limitare il numero di upload simultanei
- ❑ Accodamento delle richieste
- ❑ Priorità di incentivo
- ❑ Downloading parallelo

Lo strato di applicazione: Conclusioni

Abbiamo saputo di più su:

- gli aspetti concettuali ed implementativi delle applicazioni di rete
 - paradigma client-server
 - paradigma peer-to-peer
 - programmazione delle interfacce socket
- le principali applicazioni per Internet
 - file transfer, WWW, posta elettronica, servizio directory
- i protocolli sottostanti tali applicazioni
 - FTP, HTTP, SMTP, POP3, DNS
- i modelli di servizio (il COSA) offerti da TCP e UDP
- **Nelle prossime lezioni sullo strato di trasporto sapremo il COME ed il PERCHE' di TCP ed UDP**