

Lezione 6

Domain Name System (DNS)

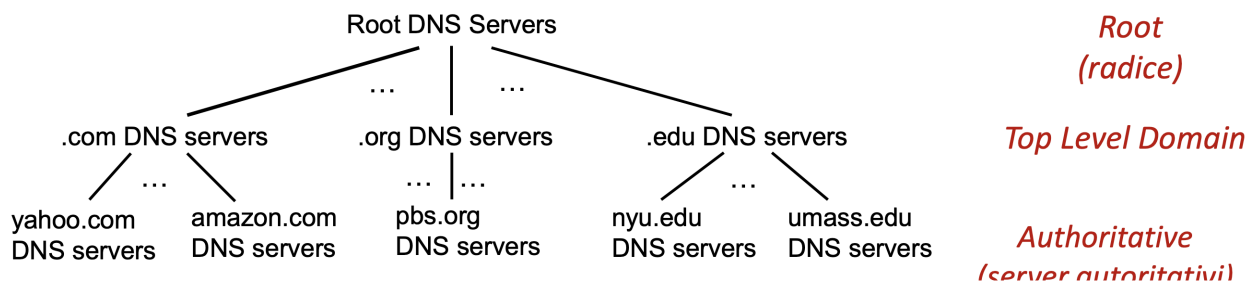
- Database distribuito implementato in una gerarchia di name server
- Protocollo a livello di applicazione che consente agli host, ai router e ai server DNS di comunicare per risolvere i nomi (traduzione nome/indirizzo)
- si noti: funzioni critiche di Internet, implementate come protocollo a livello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

Servizi DNS

- Traduzione degli hostname in indirizzi IP
 - Ecco cosa succede quando un browser in esecuzione sull'host di un utente richiede l'URL
 - L'host dell'utente , per essere in grado di inviare un messaggio di richiesta HTTP al web server deve come prima cosa ottenere il suo indirizzo IP , e avviene così :
 1. La stessa macchina utente esegue il lato client dell'applicazione DNS.
 2. Il browser estrae il nome dell'host , dall'URL e lo passa al lato client dell'applicazione DNS.
 3. Il client DNS invia una interrogazione contenente l'hostname a un DNS server.
 4. Il client DNS prima o poi riceve una risposta , che include l'indirizzo IP corrispondente all'hostname.

5. Una volta ricevuto l'indirizzo IP dal DNS , il browser può dare inizio a una connessione TCP verso il processo server HTTP collegato alla porta 80 di quell'indirizzo 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



Il client vuole l'indirizzo IP di *www.amazon.com* ; 1^a 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. 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.

Top-Level Domain e server autoritativi

Top-Level Domain (TLD) DNS server :

- si occupano dei domini .com, .org, .net, .edu, .aero, .jobs, .museums, e di tutti i domini 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: 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!

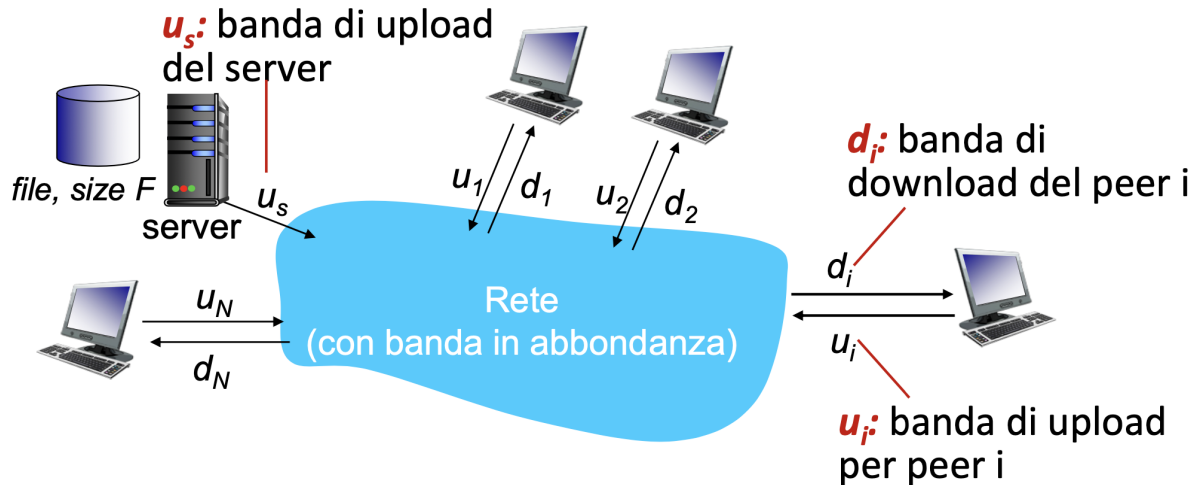
Architettura Peer-to-Peer

- nessun server sempre attivo
- sistemi periferici arbitrari comunicano direttamente
- i peer richiedono un servizio ad altri peer e forniscono un servizio in cambio ad altri peer
 - scalabilità intrinseca - nuovi peer portano nuova capacità di servizio e nuove richieste di servizio
- I peer sono connessi a intermittenza e cambiano indirizzo IP
 - gestione complessa
- Esempi: P2P file sharing (BitTorrent), streaming (KanKan), VoIP (Skype)

Distribuzione di file : client - server vs P2P

D: quanto tempo per distribuire un file (di dimensione F) da un server a N peer?

- la capacità di upload/download dei peer è una risorsa limitata



File distribution time: client-server

- trasmissione via server: deve inviare (caricare) in sequenza N copie di file:
 - tempo per inviare una copia: F/u_s
 - tempo per inviare N copie: NF/u_s
- client: ogni cliente deve scaricare una copia del file
 - d_{min} = banda di download più bassa
 - tempo di download per il client con banda minima è almeno: F/d_{min}

Tempo per distribuire F a N client usando l'approccio client-server:

$$D_{C-S} \geq \max(NF/u_s, F/d_{min})$$

Distribuzione di file : P2P

- trasmissione via server: deve trasmettere almeno una copia del file:

- tempo per inviare una copia: F/u_s
- client: ogni cliente deve scaricare una copia del file
 - Tempo per il client più lento, almeno F/d_{min}
- I client: come aggregato devono scaricare NF bit
 - capacità totale di upload (che limita la massima velocità di download) è $u_s + \sum u_i$

Tempo per distribuire F a N client usando l'approccio P2P :

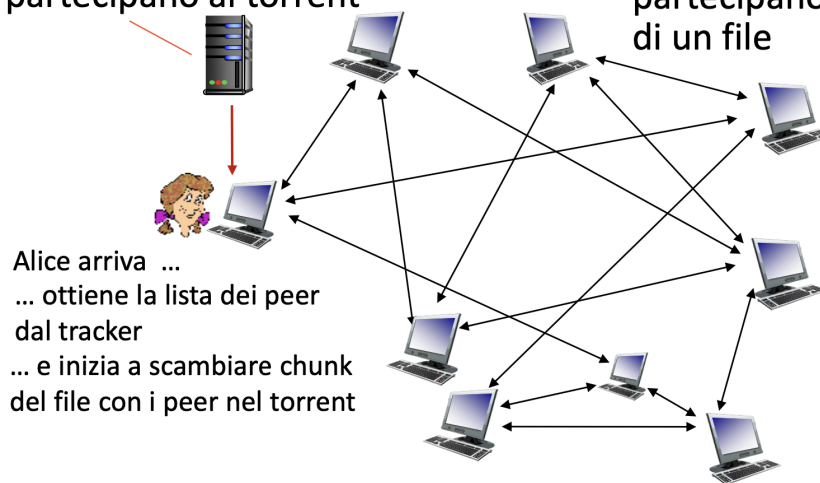
$$D_{P2P} \geq \max F/u_s, F/d_{min}, NF/(u_s + \sum u_i)$$

Distribuzione di file P2P : BitTorrent

- torrent : insieme di tutti i peer che partecipano alla distribuzione di un file
- file diviso in chunk (parti), in genere di 256 kB
- i peer nel torrent inviano/ricevono chunk del file

tracker: tiene traccia dei peer che partecipano al torrent

torrent: gruppo di peer che partecipano alla distribuzione di un file



Applicati

- Un peer che entra a far parte del torrent:
 - non ha chunk del file, ma li accumulerà nel tempo da altri peer
 - si registra con un tracker, ottenendo la lista di un sottoinsieme dei peer nel torrent (es. 50), stabilisce una connessione con un sottoinsieme di questi, che sono detti peer "vicini" ("neighbors")
 - informa periodicamente il tracker che è ancora nel torrent

- mentre scarica chunk, un peer invia i chunk già in suo possesso agli altri peer
- un peer può cambiare i peer con cui scambia i chunk
- i peer possono andare e venire
- una volta che un peer ha acquisito l'intero file, può (egoisticamente) lasciare il torrent oppure può (altruisticamente) rimanere nel torrent (come *seeder*)

Richiesta di chunk:

- In ogni momento , peer diversi hanno sottoinsiemi diversi di chunk
- periodicamente , Alice chiede ai peer vicini l'elenco dei chunk in loro possesso
- Alice richiede ai peer i chunk mancanti , adottando la strategia del **rarest first** : uniformando la distribuzione dei chunk , migliora la disponibilità globale e aumenta le possibilità di scambio
- Un peer appena entrato può chiedere un blocco in modo casuale (perché vuole avere il prima possibile un blocco da condividere) ; mentre , quando sta per completare il file , può adottare la strategia end game e richiedere lo stesso blocco a più peer simultaneamente (cancellando le richieste pendenti appena riceve un blocco)

Invio di chunk : tit-for-tat:

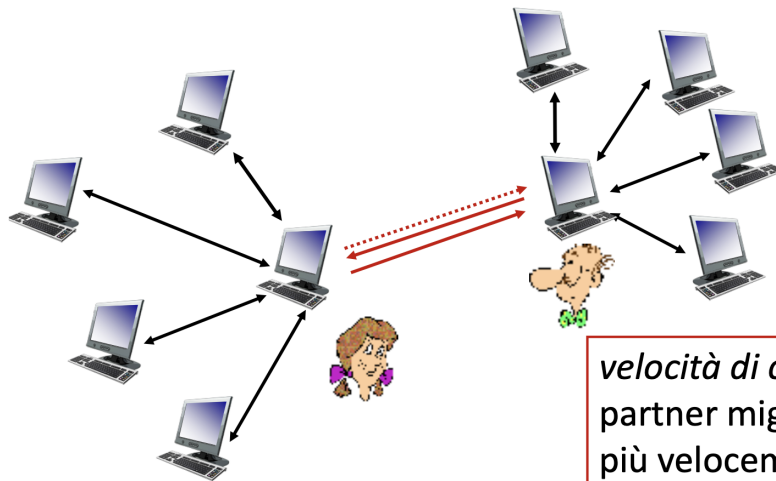
- Alice invia i chunk ai quattro peer vicini che attualmente le inviano i chunk *alla velocità più alta*
 - altri peer sono detti choked ("soffocati") non ricevono chunk da Alice
 - rivaluta i primi 4 posti ogni 10 secondi
- ogni 30 secondi : seleziona in modo casuale un vicino , inizia a inviare chunk

- questo peer è detto "optimistically unchoked"
- il nuovo peer scelto può entrare nella top 4

(1) Alice sceglie Bob come "optimistically unchoked"

(2) Alice diventa uno dei primi quattro fornitori di Bob; Bob ricambia

(3) Bob diventa uno dei primi quattro fornitori di Alice



velocità di caricamento più elevata: trovare partner migliori per gli scambi, ottenere file più velocemente!