Protocollo DNS

mercoledì 2 agosto 2023 16:27

Quando inoltriamo il traffico come sappiamo ci serve il numero di porta del socket e l'indirizzo IP del destinatario, fino ad ora abbiamo visto che gli host possono essere identificati con: **hostname**(usato dalle persone) e **indirizzi IP**(usato dai router), occorre conciliare questi due approcci, ovvero tradurre gli hostname nei loro corrispettivi indirizzi IP.
DNS si occupa di questa traduzione ed usa come protocollo di trasporto UDP.
Il DNS è quindi un protocollo che consente agli host di interrogare un database; quest'ultimo si dice **distribuito** (opposto di *CENTRALIZZATO*) ovvero implementato in una gerarchia di tanti DNS server (*DNS server: perché ai database di questi associato c'è il protocollo DNS)*.
DNS viene utilizzato da altri protocolli a livello applicativo (HTTP, SMTP), es: quando un server di posta elettronica fa una query al server DNS, sappiamo che il server destinatario ha hostname il dominio, dove gli chiediamo se esiste il server destinatario e il suo indirizzo IP, ecco cosa succede esattamente..
Supponiamo ora che un client HTTP (un browser) in esecuzione sull'host di un utente richiede l'URL www.someschool.edu/index.html. L'host dell'utente per essere in grado di inviare una richiesta HTTP al web server www.someschool.edu, deve come prima cosa ottenere il suo indirizzo IP per poterlo identificare:

1. Il browser estrae il nome dell'host da tradurre, nel nostro caso www.someschool.edu, dall'URL e lo passa al lato client dell'applicazione DNS;
2. Il lato client DNS invia una interrogazione (query) contenente l'hostname ad un DNS server, quando il client DNS riceve la risposta questa include l'indirizzo IP corrispondente all'hostname;
3. Quindi una volta ricevuto l'indirizzo IP dal DNS, il browser può dare inizio ad una connessione TCP verso il processo server HTTP collegato alla porta 80 di quell'indirizzo IP.

Nei web server (es. .com) con siti con molto traffico per distribuirlo, i siti vengono replicati su piu' server ognuno eseguito su un host diverso con un indirizzo IP differente. In questo caso quindi ad ogni hostname canonico si associa un insieme di indirizzi IP.
Il database DNS contiene questi indirizzi, quando i client effettuano una query DNS per un nome associato ad un insieme di indirizzi IP, il server risponde con l'intero insieme con ordine diverso ad ogni risposta; questo perchè quando un client (browser) invia il suo messaggio di richiesta HTTP, generalmente lo fa al primo indirizzo IP elencato nell'insieme, cambiando ogni volta l'ordinamento la rotazione DNS distribuisce il traffico sui server replicanti.
Tutte le query DNS e i messaggi di risposta vengono inviati all'interno di datagrammi UDP diretti alla porta 53.

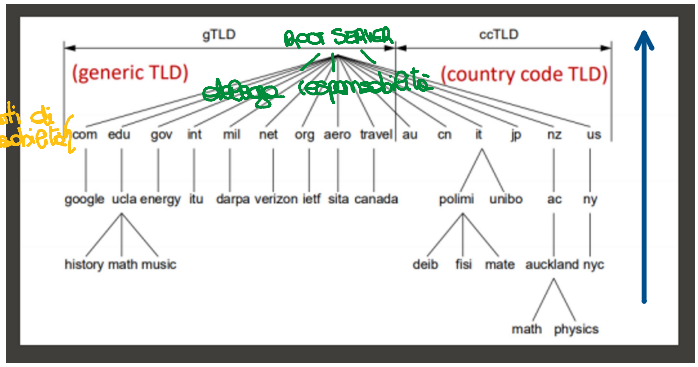
Perché non usiamo un sistema centralizzato? Uno schema centralizzato consiste nell'utilizzare un solo DNS contenente tutte le corrispondenze, tutti i client dirigerebbero tutte le richieste al singolo server e questo risponderebbe direttamente, tuttavia ha altri molteplici svantaggi:

- **Un solo point of failure**: Se il DNS server si guastasse, ne soffrirebbe l'intera Internet;
- **Volume di traffico:** Un singolo DNS server dovrebbe gestire tutte le richieste;
- **Database centralizzato distante:** Un singolo DNS server non può essere vicino a tutti i client, le query che stanno dall'altra parte del mondo devono viaggiare fino all'altro capo, causando ritardi significativi;
- **Manutenzione:** un database centralizzato deve essere aggiornato frequentemente per tenere conto di ogni nuovo host…

Database distribuito gerarchico

Per gestire il problema della ***scalabilità***, DNS utilizza un gran numero di server organizzati in maniera **gerarchica e distribuita** nel mondo. Le corrispondenze Host-Indirizzo IP sono quindi distribuite tra tutti i DNS server.
Abbiamo 3 classi di DNS server in ordine gerarchi:

1. **Root server:** ce ne sono più di 1000 sparsi nel mondo (coordinati attraverso la IANA) e forniscono gli indirizzi IP dei server autoritativi;
2. **TLD:** Questi server si occupano dei domini di primo livello (com, org, net, edu, e gov..) e compresi quelli relativi ai vari paesi (it, jp, uk, fr, ca..). I TLD server forniscono gli indirizzi IP dei server autoritativi.
3. **DNS server autoritativi:** ogni organizzazione (es. Università, società) dotata di host pubblicamente accessibili tramite Internet, ovvvero mail server e web server, deve fornire record DNS pubblicamente accessibili che associno i nomi di tali host ad Indirizzi IP. Il DNS server autoritativo dell'organizzazione ospita questi record.
   Un organizzazione può scegliere di implementare il proprio server autoritativo o di pagare un fornitore di servizi per ospitare questi record su un suo server.



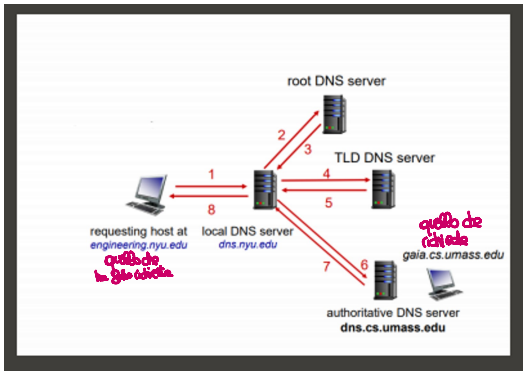
Quando un client DNS vuole determinare l'indirizzo IP relativo ad un hostname (es www.amazon.com), contatta uno dei root server, che gli restituisce uno o più indirizzi relativi al TLD server per il dominio "com".
Quindi il client DNS ora contatta uno di questi TLD server, utilizzando gli indirizzi IP ricevuti dal root, il TLD server gli restituisce uno o più indirizzi IP del server autoritativo per **amazon.com**.
Infine il client DNS contatta uno dei server autoritativi per "amazon.com" che gli restituisce l'indirizzo IP dell'hostname www.amazon.com.
Ciascun ISP ha un DNS server locale, quando un host si connette ad un ISP, quando un host si connette a un ISP, quest'ultimo gli fornisce un indirizzo IP tratto da uno o più dei suoi DNS locali, (l'IP è assegnato generalmente tramite DHCP). A questo punto una richiesta DNS viene inviata al DNS server locale, che opera da proxy e inoltra la query (della richiesta) alla gerarchia dei DNS server.

Nota, indirizzo IP: 121.7.106.83, in cui ogni punto separa uno dei byte espressi con un numero decimale tra 0 e 255.
E' gerarchico, infatti leggendolo da sinistra verso destra, otteniamo informazioni sempre più specifiche sulla collocazione dell'host in internet.

Host aliasing: host con un nome complicato, detto host canonico che ha dei soprannomi (alias).

Mail aliasing: server di posta con un nome complicato, detto mail canonico che ha dei soprannomi (alias).

Nota: abbiamo un ulteriore tipo di DNS server: DNS server LOCALE che non appartiene strettamente alla gerarchia dei server, come allora conosciamo questo local DNS?
Generalmente per funzionare in rete un computer ha bisogno di: 1.Indirizzo IP, 2.subnet mask, 3. default gateway, 4.indirizzo del local DNS (informazioni incrementali).
Con i primi 2 posso comunicare con computer della mia stessa sottorete, con in aggiunta del terzo posso comunicare con computer su altre reti, con tutto configurato ho inoltre la traduzione hostname-IP. Quest'ultimo possiamo configurarlo noi manualmente oppure usiamo un servizio DHCP.



1) engineering.nyu.edu invia un messaggio (con il nome da tradurre) di richiesta DNS al proprio server locale dns.nyu.edu;
2) Il server locale inoltra il messaggio di richiesta a un root server;
3) Il root server prende nota del suffisso **edu** e restituisce al server locale un elenco di indirizzi IP per i TLD server responsabili di "**edu**";
4) A questo punto il server locale invia il messaggio di richiesta a uno dei TLD DNS server;
5) Il TLD server prende nota del suffisso "umass.edu" e risponde con l'indirizzo IP del server autoritativo per l'università del Massachusetts, ovvero dns.umass.edu;
6) Infine, Il DNS locale rimanda il messaggio di richiesta direttamente a dns.umass.edu (server autoritativo);
7) Il server autoritativo dns.umass.edu risponde con l'indirizzo IP di gaia.cs.edu;
8) Come ultima cosa il server locale restituisce l'IP all'host che ha richiesto la query.

Nota: possiamo fare anche l'operazione inversa, tramite una pointer query, partiamo da un indirizzo IP e risaliamo all'hostname corrispondente, vengono utilizzate per aumentare l'attendibilità di un host.
Viene utilizzato un dominio "speciale" di secondo livello: in-addr.arpa

Sono necessari ben quattro messaggi DNS di richiesta e quattro di risposta!

In questo esempio abbiamo ipotizzato che il TLD server conoscesse il server autoritativo per quel dato nome, ma non è sempre così.. capita che il TLD server conosca un DNS server intermedio, il quale conosce a sua volta il server autoritativo all'hostname. In quest'ultimo caso aumentando il numero di host intermedi proporzionalmente anche il numero di messaggi DNS.

DNS caching

Il DNS server che riceve una risposta DNS può mettere in cache le informazioni contenute..

Il caching viene utilizzato per migliorare le prestazioni di ritardo e per ridurre il numero di messaggi DNS che rimbalzano in internet, concetto di **scalabilità**.
Abbiamo una coppia < *hostname, indirizzoIP* > posta nella cache di un DNS server, a questo punto un'altra richiesta con lo stesso hostname che giunge al sever, il DNS può fornire l'indirizzo IP desiderato, anche se non è autoritativo per tale indirizzo.
I DNS server invalidano le informazioni in cache dopo un periodo di tempo prefissato (in genere 2 giorni).
Un DNS server locale può memorizzare inoltre in cache gli indirizzi IP dei TLD server, consentendogli di aggirare i root server nella catena di richieste.

Nota: le voci nella cache possono essere obsolete, se l'host del nome cambia indirizzo IP, potrebbe non essere conosciuto in tutta Internet fino alla scadenza di tutti i TTL ovvero il tempo prima che le entry(i record) della cache scompaiano.

DNS record

I server che implementano il database distribuito di DNS memorizzano i cosiddetti **record di risorsa** (RR o resource record). Ogni messaggio di risposta DNS trasporta uno o più di questi.
Sono così strutturati:

(*Name*	*Value*	*Type*	*TTL)*
Dipende dal valore di Type.	*Dipende dal valore di Type.*	Type = A, allora Name è il nome dell'host e Value è il suo indirizzo IP; Type = NS, allora Name è un dominio (es: .com) e Value è l'hostname del DNS server autoritativo che sa come ottenere gli indirizzi IP degli host nel dominio; Type = CNAME, allora Value rappresenta il nome canonico dell'host per il sinonimo Name; Type = MX, allora Value è il nome canonimo di un mail server che il sinonimo Name.	Tempo residuo di vita di un record e determina quando una risorsa va rimossa dalla cache.