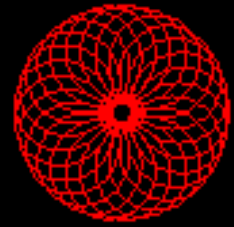


Lo Spirografo

www.spirografo.net



RETI DISTRIBUITE

Argomenti

Il tema dell'incontro sono le strutture funzionali delle reti distribuite.

Si osserveranno i meccanismi che le governano e si discuteranno le modalità di implementazione in un protocollo.

Premesse

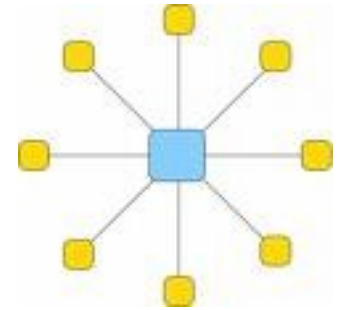
- In questo incontro si tratteranno i protocolli a livello di struttura funzionale e di modellistica. Le implementazioni a livello software saranno volutamente lasciate in secondo piano.
- In questa sede si assume di poter usare le linee di collegamento tradizionali (Internet, Lan, ecc...) per il trasporto fisico delle informazioni.

Introduzione

A livello topologico, una rete è sempre composta da un insieme di punti, detti nodi, collegati fra loro da una serie di connessioni chiamate archi.

In questa trattazione osserveremo come particolari disposizioni degli archi possano assolvere a vari compiti in modo più efficiente e sicuro di quelle usate normalmente.

Il modello client-server



- L'erogazione dei servizi è svolta da nodi specializzati (server) su richiesta di un altro nodo (client)
- Generalmente un solo server evade le richieste di molti client (da qualche unità a svariati milioni)

Il modello client-server

VANTAGGI

- Facile realizzazione
- Facile manutenzione
- Facile gestione e controllo (non sempre questo è un vantaggio)

SVANTAGGI

- **Robustezza**
 - In caso di malfunzionamento del server il servizio non è più disponibile
- **Traffico**
 - Un singolo nodo deve rispondere alle richieste di tutti i client (effetto “collo di bottiglia”)
- **Riservatezza**
 - È possibile cifrare solo i contenuti delle comunicazioni, mentre mittente e destinatario sono noti

Il modello distribuito

Perché un protocollo di rete sia distribuito deve avere queste caratteristiche:

- I nodi che erogano uno stesso servizio devono essere fisicamente dislocati in luoghi diversi
- I nodi di distribuzione devono essere molti ed indipendenti (sia fisicamente che “politicamente”)
- La struttura funzionale deve essere ridondante e non deve prevedere punti nevralgici vitali

Il modello distribuito

Esempi:

- Tor
- Kademlia
- Netsukuku
- Rete DNS
- Gnutella

Come si evince dagli esempi, non è sempre necessario rinunciare alla logica client-server.

In questi caso però non si ha più un solo server, ma un insieme di tanti server che formano a loro volta una “rete nella rete” atta a fornire specifici servizi.

Questa logica può essere molto utile qualora alcuni servizi siano fuori dalla portata di alcuni utenti (vedi file-sharing e NAT)

Il coefficiente di distribuzione

A questo punto ci occorre quantificare quanto una rete sia distribuita.

Per questo definiamo un apposito indice, che chiameremo coefficiente di distribuzione, per esprimere in prima approssimazione quanto sia effettivamente decentrata una specifica configurazione.

Per calcolarne il valore basta applicare questa semplice formula:

$$K = \frac{S}{N}$$

Dove K è il nostro coefficiente, S è il numero complessivo dei nodi specializzati e N è il numero totale dei nodi collegati alla rete. Moltiplicando per 100 il valore di K si ottiene l'espressione in forma percentuale.

Reti P2P

Quando il coefficiente di distribuzione raggiunge valori prossimi a 1 (distribuzione del 100%), diremo che la rete è omogenea.

Questo significa che è composta da nodi dello stesso tipo.



Reti P2P

Vantaggi:

- **Robustezza:**

- Grazie alla ridondanza, la compromissione di un numero limitato di nodi non altera significativamente l'assetto della rete

- **Traffico:**

- Le richieste possono essere smistate fra vari nodi, bilanciando dinamicamente il carico ed evitando ingorghi. Inoltre sono assai più rari i “colli di bottiglia”.

- **Riservatezza:**

- È possibile occultare sia i contenuti della trasmissione che l'identità degli interlocutori

Reti P2P

Svantaggi:

- Difficile realizzazione
- Ogni nodo ha difficoltà a mappare la rete
- Difficile ricerca di una risorsa
- Permeabilità a nodi-civetta e a soggetti indesiderabili
- Vulnerabilità alle reazioni a catena

Casi pratici

Nella pratica spesso si adottano soluzioni ibride, oppure si mettono a punto algoritmi specifici per ovviare ai vari inconvenienti.

Per esempio:

- Tor
- Kademlia
- Netsukuku