

Sviluppo di software di rete con l'utilizzo di sistemi UNIX

TCP/IP

Claudio Vicari

Protocolli TCP/IP

- ➔ la suite di protocolli TCP/IP è strutturata a livelli (si parla di stack di protocolli)
- ➔ nel TCP/IP vi sono cinque livelli:
 - 1 livello fisico: non specificato dall'architettura
 - 2 livello datalink: anche questo non è specificato
 - 3 livello network (IP)
 - 4 livello transport (TCP)
 - 5 livello application

Livello 3 - network

- ➔ il livello 3 assume l'esistenza di un canale non affidabile sottostante
- ➔ si occupa di inoltrare i pacchetti, senza controlli sulle perdite e sulle congestioni della rete
- ➔ nel livello 3 si situano diversi protocolli:
 - ICMP (messaggi di controllo)
 - IGMP
 - ARP (address resolution protocol)
 - RARP (reverse address resolution)
 - IP (Internet Protocol)

Livello 4 - transport

- ➔ serve a fornire un canale affidabile ai protocolli di livello superiore (permette di isolarli da questi dettagli)
- ➔ i pacchetti vengono ricevuti al livello superiore nello stesso ordine in cui vengono inviati
- ➔ gestisce la comunicazione fra due host nella rete, indipendentemente dalla loro locazione fisica
- ➔ fornisce anche controlli sulla congestione
- ➔ protocolli: TCP, UDP

Livello 5 - application

- ➔ nel livello application vivono tutti i protocolli di uso comune per l'utenza finale:
 - DNS
 - POP, IMAP
 - SMTP
 - FTP
 - HTTP, HTTPS
 - telnet, ssh
 - smb, nfs
 - NIS
 - ...

Identificazione delle risorse in internet

- ⇒ una interfaccia di rete, nella rete, è identificata mediante il suo indirizzo IP
 - questo viene utilizzato dai router, per inoltrare i pacchetti
 - un host può avere più interfacce di rete
- ⇒ nei protocolli TCP e UDP, si fa uso di numeri di porta per distinguere fra differenti servizi, all'interno dello stesso host

Identificazione delle risorse in internet

- ⇒ il protocollo DNS permette di effettuare una conversione fra nomi di host e indirizzi IP
 - es: www.google.com, google.com
- ⇒ nei protocolli di livello application, si fa uso delle URL per identificare una risorsa
 - le URL sono composte di: protocollo, host, numero di porta, sottorisorsa
 - es: <http://www.google.com>

TCP: modalità client-server

- ➔ normalmente un singolo host svolge il ruolo di server
- ➔ altri host svolgono il ruolo di client
- ➔ il server attende che arrivino richieste da parte dei client
- ➔ gestisce ogni richiesta
- ➔ è in grado di gestire un numero arbitrario di richieste

TCP: stabilimento di una connessione

- 1 il server deve essere pronto ad accettare l'arrivo di una nuova connessione. In questa fase bisogna specificare la porta
- 2 il client richiede l'apertura di una connessione
- 3 il server conferma la connessione
- 4 i due host possono iniziare a colloquiare... da questo momento, la connessione fra i due non distingue nettamente un client da un server
-
- 6 uno dei due host richiede l'abbattimento della connessione
- 7 l'altro host richiede la chiusura, e la connessione a questo punto è definitivamente chiusa

Socket: definizione

- ⇒ socket in TCP:
 - è la coppia formata da (indirizzo IP, numero di porta)
 - dunque, un socket è l'indirizzo cui inviare dati
- ⇒ connessione in TCP:
 - è costituita da una coppia di socket
- ⇒ socket in Unix:
 - è un oggetto, che descrive un endpoint per la comunicazione
 - può indicare un server che attende una connessione
 - si usa anche per una connessione già stabilita
 - il terminale indica anche connessioni non di rete

Visualizzare le connessioni nel sistema

- ➔ in Unix possiamo visualizzare lo stato delle connessioni
- ➔ `netstat -a` mostra tutte i socket, compresi quelli puramente in ascolto
- ➔ con `netstat -lt`, visualizziamo tutti i server tcp in attesa di accettare connessioni
- ➔ con `netstat -t`, visualizziamo tutte le connessioni tcp
- ➔ `netstat` ci fa visualizzare lo stato di tutti i socket nel sistema!