Introduzione alle applicazioni di rete

Definizioni base
Infrastruttura delle applicazioni di rete
Decisioni di progetto
Modello client-server
Identificazione di un servizio e concorrenza
Modello peer-to-peer

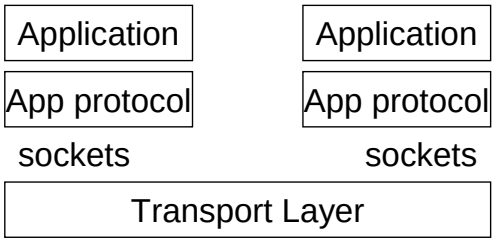
Prof. Filippo Lanubile

Definizioni base

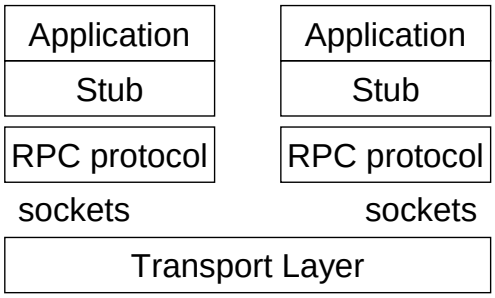
- Applicazione di rete (distribuite)
  - insieme di processi comunicanti che girano su host diversi e forniscono/realizzano un servizio ad un utente finale
- Applicazioni Internet
  - utilizzano servizi della pila di protocolli TCP/IP
  - le nuove applicazioni guidano l'evoluzione di Internet
- Applicazioni conformi a standard Internet
  i protocolli che descrivono le specifiche sono proposti come Request for Comments (RFC) e poi accettati come standard
  - Esempi: telnet, ftp, smtp, http, chat,
- Meccanismo a basso livello di comunicazione tra processi: sockets (BSD sockets, Windows sockets)
- Meccanismo ad alto livello di comunicazione tra processi: protocollo Remote Procedure Calls (RPC), RFC 1057

Infrastruttura delle applicazioni di rete

- Applicazione basata su protocolli a livello di applicazione

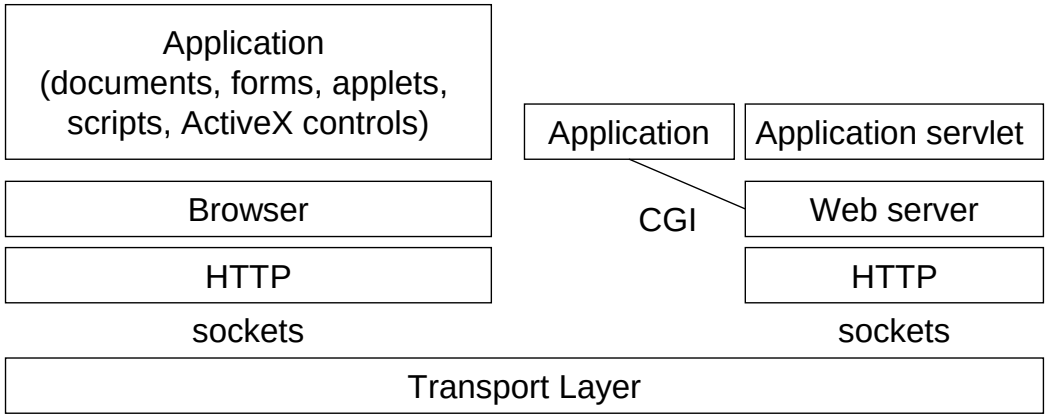


- Applicazione basata su RPC (DCOM, CORBA, Java RMI forniscono un ulteriore livello sul meccanismo base RPC)



Infrastruttura delle applicazioni di rete (cont.)

- Applicazione basata sul web

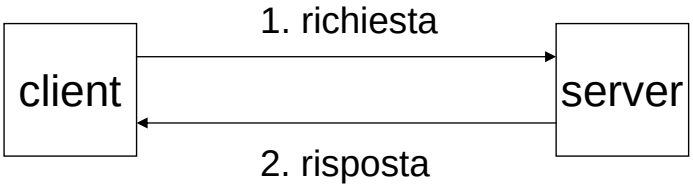


Decisioni di progetto

- Tipo di interazione (tra applicazioni o componenti)
  - interazione simmetrica o asimmetrica
  - interazione sincrona o asincrona
  - interazione 1-1, 1-M, M-M
- Servizio di trasporto richiesto
  - orientato alla connessione o privo di connessione
  - affidabile o best-effort
  - qualita' del servizio (QOS: quality of service) garantita o no
- Servizio di presentazione richiesto
  - conversione di dati richiesta oppure no

Modello client-server

- Quasi tutte le applicazioni di rete usano come modello di interazione il modello client-server
- Il server offre un servizio (accesso alle risorse condivise) su richiesta (attesa passiva)
- Il client richiede un servizio iniziando la comunicazione
- Interazione asimmetrica: produttore-consumatore
- Interazione sincrona: richiesta-risposta



Caratteristiche di un client e di un server

Client

- Esegue sul computer locale
- E' invocato dall'utente
- Inizia il contatto con il server
- Spedisce una richiesta, opzionalmente con dati
- Puo' accedere a piu' servizi (uno per volta)

Server

- Esegue su un computer remoto
- Parte all'inizializzazione del sistema
- Aspetta le richieste di servizio dai client: loop fino a che non arriva una richiesta
- Spedisce una risposta, opzionalmente con dati
- Accetta richieste da client arbitrari: un servizio per ogni client

Scelta del tipo di connessione

TCP - orientato alla connessione

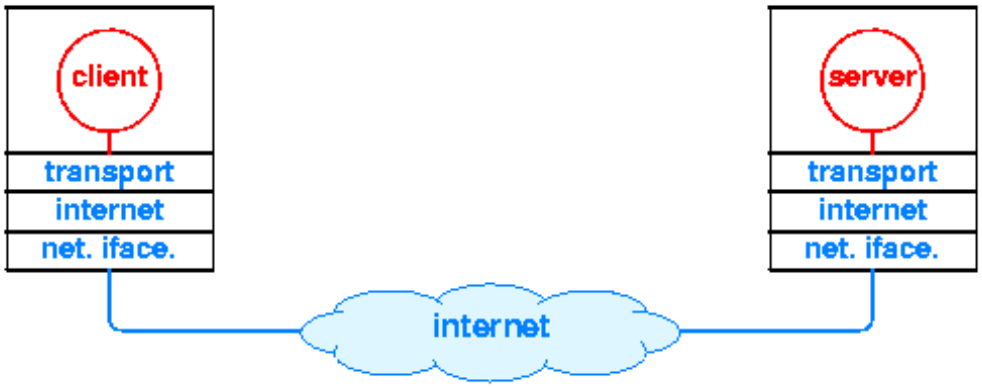
- Il client stabilisce la connessione al server
- Il client e il server si scambiano messaggi multipli di dimensione arbitraria
- Il client termina la connessione

UDP - privo di connessione

- Il client costruisce il messaggio
- Il client spedisce il messaggio al server
- Il messaggio deve entrare in un datagramma UDP
- Il server risponde

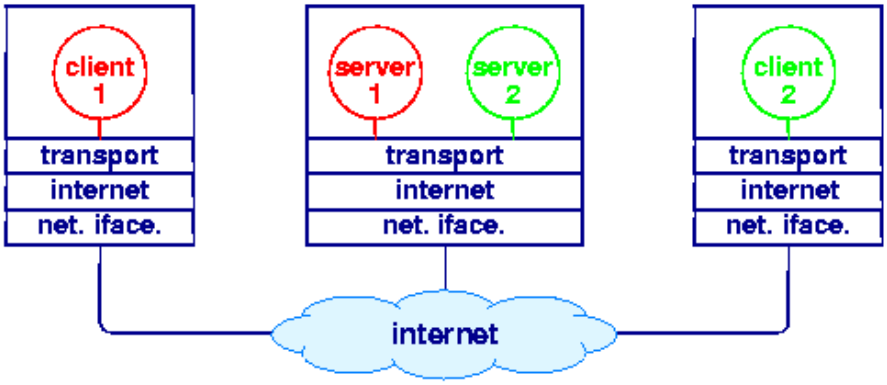
Client-server e protocolli di rete

- Sia il client che il server devono lavorare con la stessa pila di protocolli e interagire con lo stesso livello di rete



Servizi multipli su un computer

- Calcolatori potenti, con sistema operativo multi-tasking, possono ospitare piu' di un server
- Ogni server offre un servizio specifico ai client che lo richiedono
- I server eseguono come processi indipendenti e possono gestire i client simultaneamente



Concorrenza nei server

- Problema:
  - La risposta a una richiesta di un client può richiedere molto tempo
  - Altri client devono aspettare il completamento delle richieste precedenti
- Soluzione
  - Server concorrente che permette di gestire contemporaneamente le richieste di client multipli
  - Il server crea dinamicamente un nuovo thread (copia del processo) per ogni richiesta accettata
  - Il thread concorrente (con il suo flusso di controllo autonomo operante su memoria condivisa) soddisfa la richiesta del client
  - Il server si rende subito disponibile e attende le successive richieste

Identificazione di un servizio

- A ogni servizio offerto da un server è assegnato un identificatore unico per l'host (numero di porta)
- Il server si registra con il software del protocollo di trasporto usando il numero di porta
  - Registrazione da programma (bind)
  - Registrazione da file di sistema (services)
- Il client contatta il software del protocollo usando l'identificatore dell'host (indirizzo IP) + l'identificatore del servizio (numero di porta)
  - L'indirizzo IP e il numero di porta del server deve essere noto a priori
- Anche il client è identificabile tramite indirizzo IP + numero di porta
  - Il client comunica al server il proprio indirizzo completo al momento della richiesta
- Un numero di porta non può essere usato da più processi (server o client) contemporaneamente
- Un processo (server o client) può usare più numeri di porta contemporaneamente
  - Ogni thread di un server concorrente usa numeri di porta distinti

Connessioni e server concorrenti

- Problema
  - I server concorrenti creano un thread per ogni richiesta accettata
  - Richieste simultanee sono servite da thread concorrenti
  - Ogni client è identificato da: (ind. IP, n.ro di porta)
  - Ogni servizio (server) è identificato da: (ind. IP, n.ro di porta)
  - Client e server interagiscono scambiandosi più messaggi
  - Come fa un client a far arrivare i messaggi (incoming messages) al thread assegnato?
- Soluzione
  - Utilizzare l'identificazione della sessione di trasporto
    - (ind. IP server, n.ro di porta server, ind. IP client , n.ro di porta client)
  - Il software che implementa il protocollo di trasporto associa ogni thread creato a una quadrupla distinta

Modello peer-to-peer (1)

- Il modello di interazione peer-to-peer (P2P) è l'opposto del modello client-server
  - L'applicazione è suddivisa in componenti, posti su nodi distinti della rete, che agiscono in modo paritetico (interazione simmetrica)
  - Le informazioni sono suddivise tra i nodi della rete piuttosto che essere concentrate in un singolo nodo
  - Ogni pari può stabilire connessioni dirette con altri pari attivi nella rete (interazione sincrona)

Modello peer-to-peer (2)

- Il modello P2P in realtà è già utilizzato nei livelli più bassi (Ethernet, Internet routing)
- Il modello P2P è innovativo nella progettazione di applicazioni di rete
  - File sharing
    - Napster (www.napster.com)
    - Gnutella (gnutella.wego.com)
  - Elaborazione distribuita
    - SETI@home (setiathome.ssl.berkeley.edu)
    - Folding@home (www.stanford.edu/group/pandegroup/Cosm)
  - Instant Messaging
    - Jabber (www.jabber.org)
  - Collaborazione
    - Groove (www.groove.net)

Caratteristiche PIE delle applicazioni P2P

- PIE: Presence, Identity, Edge Resources
- Edge Resources
  - Le applicazioni P2P sfruttano le risorse disponibili (memoria, CPU, contenuto, presenza umana) ai confini di Internet (principalmente desktop PC)
  - Le applicazioni P2P organizzano le risorse distribuite in un pool di dimensione variabile
    - Il problema è che le risorse sono transitorie
- Identity
  - Le applicazioni P2P devono essere in grado di identificare univocamente le risorse disponibili
    - Non è possibile usare nomi DNS
- Presence
  - Capacità di riconoscere quando una risorsa è online