

# Introduzione alle applicazioni di rete

Definizioni base

Infrastruttura delle applicazioni di rete

Decisioni di progetto

Modello client-server

Modello peer-to-peer

Topologie di sistemi distribuiti

Prof. Filippo Lanubile

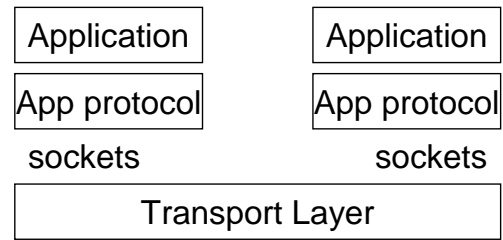
## Definizioni base

- 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

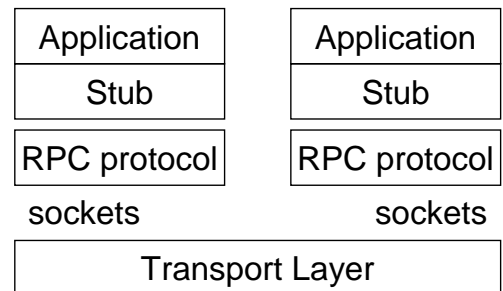
Prof. Filippo Lanubile

# Infrastruttura delle applicazioni di rete

- Applicazione basata su protocolli a livello di applicazione



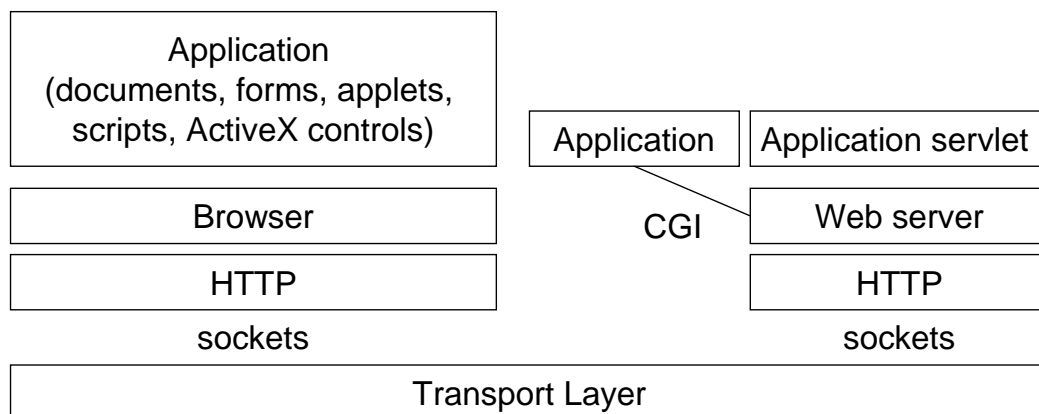
- Applicazione basata su RPC (DCOM, CORBA, Java RMI forniscono un ulteriore livello sul meccanismo base RPC)



Prof. Filippo Lanubile

## Infrastruttura delle applicazioni di rete (cont.)

- Applicazione basata sul web



Prof. Filippo Lanubile

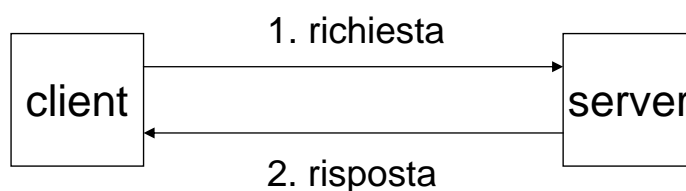
# Decisioni di progetto

- 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
  - qualità del servizio (QOS: quality of service) garantita o no
- Servizio di presentazione richiesto
  - conversione di dati richiesta oppure no

Prof. Filippo Lanubile

## 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



Prof. Filippo Lanubile

# 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

Prof. Filippo Lanubile

# 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

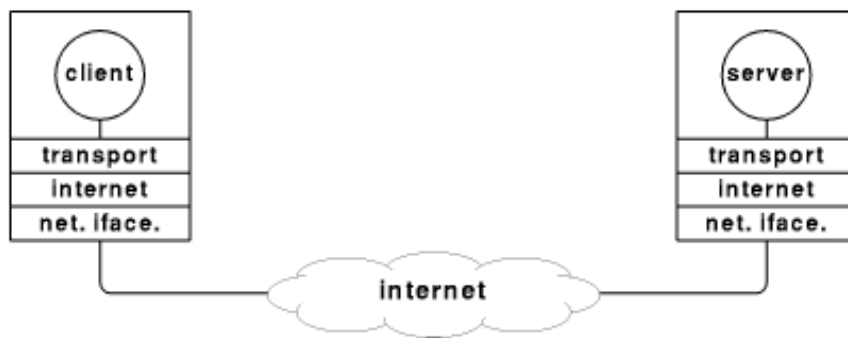
## UDP - privo di connessione

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

Prof. Filippo Lanubile

# Client-server e protocolli di rete

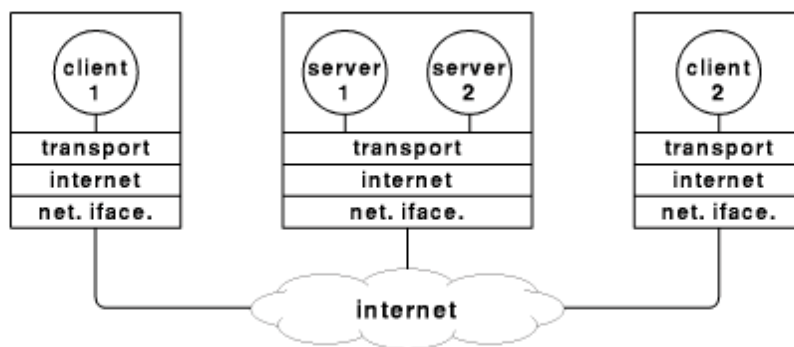
- Sia il client che il server devono lavorare con la stessa pila di protocolli e interagire con lo stesso livello di rete



Prof. Filippo Lanubile

## Servizi multipli su un computer

- Calcolatori potenti, con sistema operativo multi-tasking, possono ospitare più di un server
- I server eseguono come processi indipendenti e possono gestire i client simultaneamente



Prof. Filippo Lanubile

# Server multipli per un servizio

- Problema:
  - La risposta a una richiesta di un client puo' prendere molto tempo
  - Altri client devono aspettare
- Soluzione
  - Server multipli che gestiscono le richieste concorrentemente permettendo alle richieste piu' brevi di non aspettare il completamento di quelle piu' lunghe
  - Creazione dinamica di un processo server per ogni richiesta
  - Master server accetta le richieste in arrivo e lancia uno slave server per ogni client
  - Slave server gestisce la richiesta dal suo client
  - Master server attende le successive richieste

Prof. Filippo Lanubile

## Identificazione di un servizio

- A ogni servizio e' assegnato un identificatore unico che e' usato sia dal client che dal server
- Il server si registra con il software del protocollo locale usando l'identificatore
- Il client contatta il software del protocollo usando l'identificatore
- I protocolli di trasporto usano i numeri di porta come identificatori
- Sessione di trasporto identificata da due coppie:
  - (indirizzo IP, numero di porta) sul server
  - (indirizzo IP, numero di porta) sul client
- Non ci possono essere sullo stesso computer due server o due client con lo stesso numero di porta

Prof. Filippo Lanubile

# 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)

Prof. Filippo Lanubile

# 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](http://www.napster.com))
    - Gnutella ([gnutella.wego.com](http://gnutella.wego.com))
  - Elaborazione distribuita
    - SETI@home ([setiathome.ssl.berkeley.edu](http://setiathome.ssl.berkeley.edu))
    - Folding@home ([www.stanford.edu/group/pandegroup/Cosm](http://www.stanford.edu/group/pandegroup/Cosm))
  - Instant Messaging
    - Jabber ( [www.jabber.org](http://www.jabber.org))
  - Collaborazione
    - Groove ([www.groove.net](http://www.groove.net))

Prof. Filippo Lanubile

# 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

Prof. Filippo Lanubile

## Topologie di sistemi distribuiti

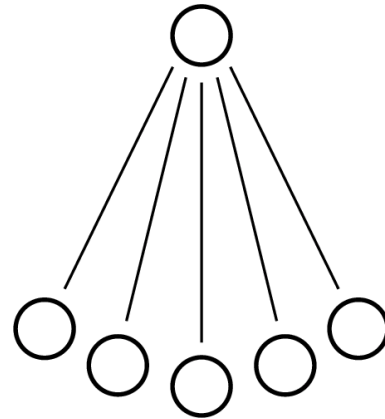
- Topologia di un sistema distribuito
  - come i nodi di un sistema distribuito sono connessi in termini di flusso delle informazioni
- Rappresentazione tramite grafi
  - I nodi rappresentano computer su cui sono eseguiti moduli dell'applicazione distribuita
  - Gli archi rappresentano condivisione delle informazioni tra i nodi
- Topologie base
  - centralizzata, ad anello, gerarchica, decentralizzata
- Topologie ibride
  - Combinazioni di topologie base

Prof. Filippo Lanubile



# Topologia centralizzata

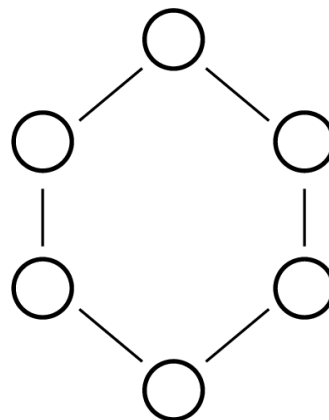
- Tipica del modello di interazione client/server
  - App 2-tier basate su database
  - Siti web semplici
  - App web 2-tier
- Utilizzata anche da applicazioni P2P
  - Napster (funzione di ricerca)
  - SETI@Home (job dispatcher)



Prof. Filippo Lanubile

# Topologia ad anello

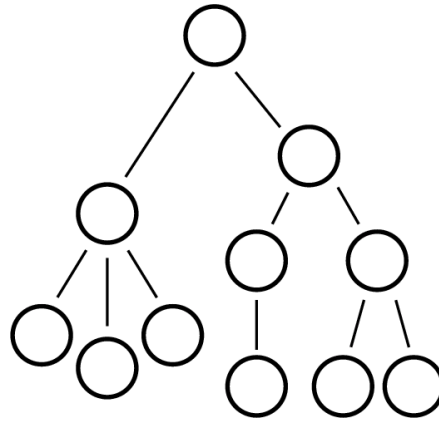
- Cluster di host che
  - condividono un unico stato
  - forniscono funzioni identiche
- Caratteristiche di
  - Fault tolerance
  - Bilanciamento del carico
- Assunzione
  - stessa rete (e stesso proprietario)



Prof. Filippo Lanubile

# Topologia gerarchica

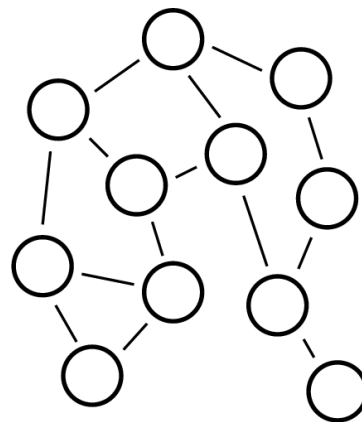
- DNS: Domain Name System ([www.dns.net/dnsrd](http://www.dns.net/dnsrd))
- NTP: Network Time Protocol ([www.ntp.org](http://www.ntp.org))
- Usenet ([www.usenet.org](http://www.usenet.org))



Prof. Filippo Lanubile

# Topologia decentralizzata

- Realizzazione “pura” del P2P
  - interazione simmetrica
  - ruoli identici
- Gnutella ([gnutella.wego.com](http://gnutella.wego.com))
- Freenet ([freenet.sourceforge.net](http://freenet.sourceforge.net))



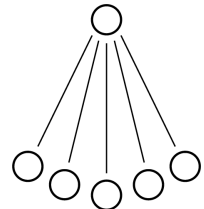
Prof. Filippo Lanubile

# Fattori di qualità

- Gestibilità
  - Quanto è facile far funzionare il sistema nel tempo?
- Coerenza dell'informazione
  - Quanto sono affidabili le informazioni condivise?
- Estensibilità
  - Quanto è facile aggiungere nuovi nodi?
- Tolleranza agli errori
  - Quanto sono sopportati i malfunzionamenti?
- Sicurezza
  - Quanto è difficile sovvertire il sistema?
- Resistenza a interventi legali
  - Quanto è difficile chiudere il sistema?

Prof. Filippo Lanubile

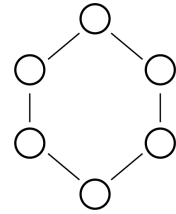
## Valutazione qualitativa della topologia centralizzata



Gestibilità	✓	Il sistema è in pratica tutto in un unico nodo
Coerenza	✓	Tutte le informazioni sono su un unico nodo
Estensibilità	X	Non è possibile aggiungere nodi centrali
Tolleranza	X	Punto singolo di malfunzionamento
Sicurezza	✓	Un solo nodo da rendere sicuro
Resistenza	X	Facile da chiudere

Prof. Filippo Lanubile

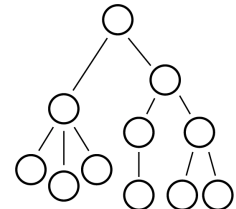
# Valutazione qualitativa della topologia ad anello



Gestibilità	✓	Regole semplici di relazionamento
Coerenza	✓	Logica semplice di aggiornamento dello stato
Estensibilità	X	Solo il proprietario può aggiungere
Tolleranza	✓	Il fallimento di un nodo lo esclude
Sicurezza	✓	L'anello ha un solo proprietario
Resistenza	X	Interdizione del proprietario

Prof. Filippo Lanubile

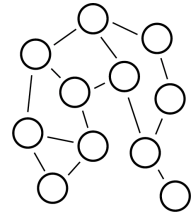
# Valutazione qualitativa della topologia gerarchica



Gestibilità	½	Catena di autorità
Coerenza	½	Consistenza della cache
Estensibilità	½	Aggiungere altri nodi foglia, ribilanciare
Tolleranza	½	Vulnerabilità del nodo radice
Sicurezza	X	Facilità di spoofing
Resistenza	X	Basta chiudere il nodo radice

Prof. Filippo Lanubile

# Valutazione qualitativa della topologia decentralizzata

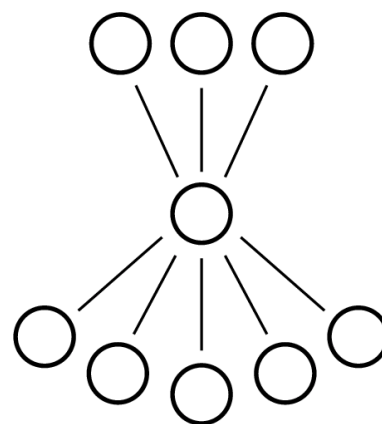


Gestibilità	X	Molto difficile, troppi proprietari
Coerenza	X	Difficile, informazioni inaffidabili
Estensibilità	✓	Chiunque può unirsi
Tolleranza	✓	Rindondanza
Sicurezza	X	Difficile, campo di ricerca aperto
Resistenza	✓	Nessuno (o troppi) da perseguire

Prof. Filippo Lanubile

## Topologia centralizzata + centralizzata

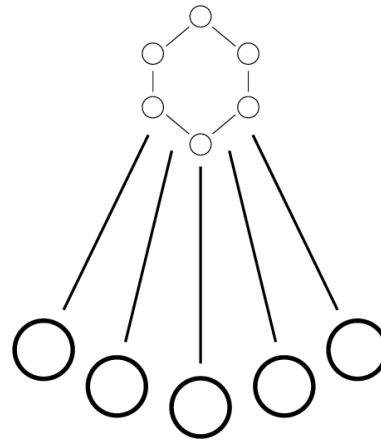
- Un modulo su un nodo intermedio è server per alcuni moduli e client per altri moduli
  - App n-tier basate su database
  - App web n-tier
  - Intermediari di web services
    - Grand Central ([www.grandcentral.com](http://www.grandcentral.com))
    - .NET My Services ([www.microsoft.com](http://www.microsoft.com))



Prof. Filippo Lanubile

# Topologia centralizzata + ad anello

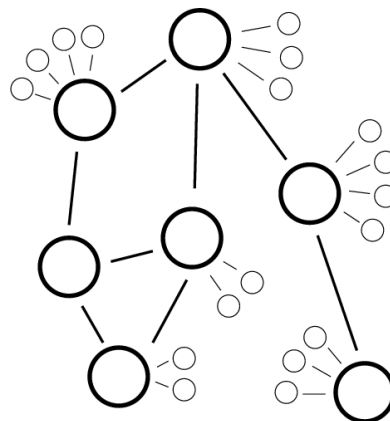
- Siti web ad alto traffico
- Applicazioni web critiche



Prof. Filippo Lanubile

# Topologia centralizzata + decentralizzata

- Email
- FastTrack file-sharing system  
([www.fasttrack.nu/](http://www.fasttrack.nu/))  
utilizzato come infrastruttura da
  - KaZaA ([www.kazaa.com](http://www.kazaa.com))
  - Grokster ([www.grokster.com](http://www.grokster.com))
  - Morpheus ([www.musiccity.com](http://www.musiccity.com))



Prof. Filippo Lanubile

# Conclusioni

- La topologia centralizzata è più semplice
  - La più diffusa per sistemi distribuiti
  - Si combina bene con topologie ad anello
- La topologia decentralizzata offre altri vantaggi ma necessita di ulteriore ricerca per migliorare
  - Gestibilità, coerenza, sicurezza
- La topologia gerarchica è trascurata
- Le topologie ibride combinano i vantaggi piuttosto che gli svantaggi

Prof. Filippo Lanubile