Programmazione in Rete a.a. 2005/2006

http://www.di.uniba.it/~lisi/courses/prog-rete/prog-rete0506.htm

dott.ssa Francesca A. Lisi lisi@di.uniba.it

Orario di ricevimento: mercoledì ore 10-12



Sommario della lezione di oggi: Lo strato di applicazione (1/5)

- Principi dei protocolli dello strato di applicazione
- Trasferimento di file & il protocollo FTP
- □ World Wide Web & HTTP
- □ Posta elettronica & SMTP
- DNS: il servizio directory di Internet
- Condivisione di file
- Programmazione dei socket



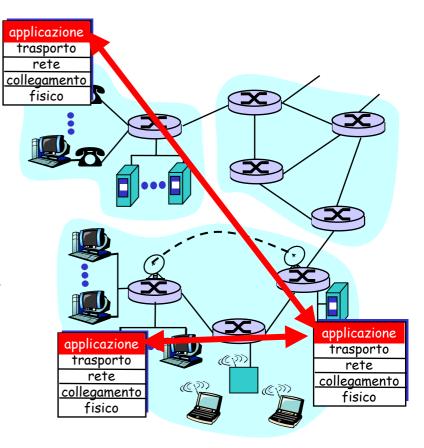
<u>Applicazioni di rete e</u> <u>protocolli dello strato di applicazione</u>

Applicazione di rete = insieme di processi distribuiti e comunicanti

- processi in esecuzione sui terminali nello "spazio utente"
- processi si scambiano messaggi per implementare l'applicazione
- o e.g., email, Web

Protocollo dello strato di applicazione

- "pezzo" di un'applicazione di rete
- definisce messaggi scambiati dalle applicazioni e azioni da intraprendere
- usa i servizi forniti da protocolli dello strato sottostante





Applicazioni di rete: un po' di gergo tecnico

- □ Un processo è un programma in esecuzione su un terminale.
- Nello stesso terminale, due processi comunicano con comunicazione interprocesso definita dal sistema operativo.
- ☐ I processi in corso su diversi terminali comunicano tramite un protocollo dello strato di applicazione

- □ Un agente utente è una interfaccia fra l'utente e l'applicazione di rete.
 - Web: browser
 - E-mail: mail reader
 - streaming audio/video: media player



I protocolli dello strato di applicazione

Application Programming Interface (API):

- □ definisce l'interfaccia fra strato di applicazione e strato di trasporto
- □ socket: Internet API
 - due processi comunicano inviando dati nel socket, leggendo dati dal socket

- D: come fa un processo P1 ad "identificare" il processo P2 con cui vuole comunicare?
 - Indirizzo IP del terminale
 T2 che esegue P2
 - numero di porta su cui P2 sta girando - consente al T2 di determinare a quale processo locale il messaggio dovrebbe essere consegnato



<u>Di quale servizi di trasporto</u> <u>ha bisogno un'applicazione di rete?</u>

Trasferimento affidabile dei dati

- alcune applicazioni, p.e. audio, possono tollerare una certa perdita di dati
- altre applicazioni, p.e. file transfer e telnet, richiedono affidabilità al 100%

Tempismo

alcune applicazioni, p.e. telefonia Internet e giochi interattivi, richiedono un basso ritardo per essere "efficace"

Larghezza di banda

- alcune applicazioni, p.e. multimedia, richiedono un minimo ammontare di larghezza di banda per essere "efficace"
- □ altre applicazioni
 ("applicazioni elastiche")
 fanno uso di qualsiasi
 larghezza di banda esse
 dispongano



Applicazioni Internet: requisiti sui servizi di trasporto

Applicazione	Tolleranza alla perdita di dati	Ampiezza di banda	Sensibilità al tempo
Trasferimento file	No	Variabile	No
Posta elettronica	No	Variabile	No
Documenti Web	No	Variabile	No
Audio/video in tempo reale	Sì	Audio: da 5 Kbps a 1 Mbps Video: da 10 Kbps a 5 Mbps	Sì, centinaia di ms
Audio/video memorizzati	Sì	Come sopra	Sì, pochi secondi
Giochi interattivi	Sì	Fino a pochi Kbps	Sì, centinaia di ms
Messaggistica istantanea	No	Variabile	Sì e no



<u>Servizi forniti dai</u> <u>protocolli di trasporto di Internet</u>

Servizi TCP:

- orientamento alla connessione: setup di richiesta fra client e server
- □ trasferimento affidabile dei dati
- □ controllo di flusso: il mittente non sommergerà il ricevente
- controllo della congestione:
 rallenta il mittente se la rete è sovraccarica
- non fornisce: tempismo, garanzie di ampiezza minima di banda

Servizi UDP:

- 🗖 trasferimento dei dati
- non fornisce: orientamento alla connessione, trasferimento affidabile dei dati, controllo di flusso, controllo di congestione, tempismo, o garanzia di ampiezza di banda

D: Perché esiste un modello di servizi UDP?



Applicazioni Internet: protocolli di applicazione e di trasporto

Applicazione	Protocollo a livello applicazione	Protocollo di trasporto sottostante
Posta elettronica	SMTP [RFC 2821]	TCP
Accesso a terminali remoti	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
Trasferimento file	FTP [RFC 959]	TCP
Multimedia in streaming	Proprietario (ad esempio, RealNetworks)	TCP o UDP
Telefonia Internet	Proprietario (ad esempio, Vonage, Dialpad)	Tipicamente UDP



Architettura delle applicazioni di rete

- □ Client-server
- □ Peer-to-peer (P2P)
- □ Architetture ibride (client-server e P2P)



Il paradigma client-server

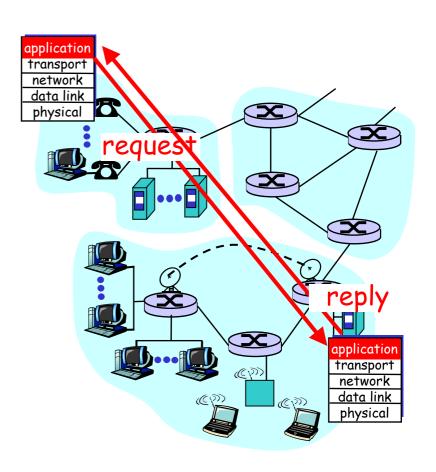
Una tipica appl. di rete ha due pezzi: un *client* ed un *server*

Client:

- si mette in contatto con il server ("parla per primo")
- tipicamente richiede un servizio al server,
- per il Web, il *client* è implementato nel browser; per la e-mail, nel mail reader

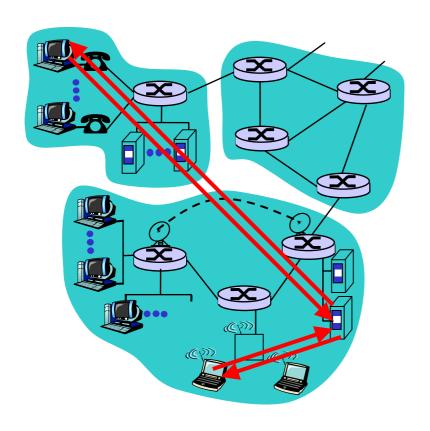
Server:

- 🗖 fornisce il servizio richiesto
- p.e., il server Web invia la pagina Web richiesta, il mail server consegna la e-mail





Architettura delle applicazioni di rete: modello client-server



server:

- host sempre attivo
- o indirizzo IP fisso
- server farm per creare un potente server virtuale

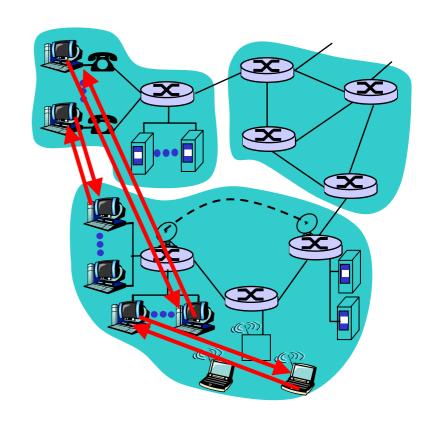
client:

- o comunica con il server
- può contattare il server in qualunque momento
- può avere indirizzi IP dinamici
- non comunica direttamente con gli altri client



Architettura delle applicazioni di rete: modello P2P

- non c'è un server sempre attivo
- coppie arbitrarie di host (peer) comunicano direttamente tra loro
- □ i peer non devono necessariamente essere sempre attivi, e cambiano indirizzo IP
- Un esempio: Gnutella
 Facilmente scalabile
 Difficile da gestire





Architettura delle applicazioni di rete: modello ibrido

Napster

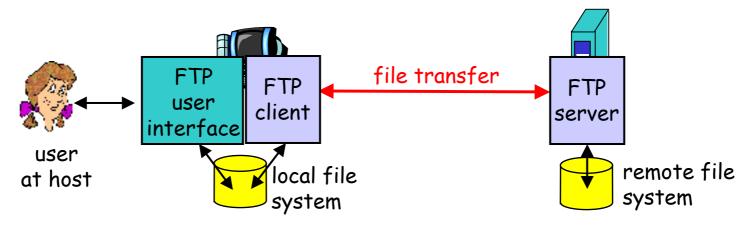
- Scambio di file secondo la logica P2P
- Ricerca di file centralizzata:
 - i peer registrano il loro contenuto presso un server centrale
 - i peer chiedono allo stesso server centrale di localizzare il contenuto

Messaggistica istantanea

- La chat tra due utenti è del tipo P2P
- Individuazione della presenza/location centralizzata;
 - l'utente registra il suo indirizzo IP sul server centrale quando è disponibile online
 - l'utente contatta il server centrale per conoscere gli indirizzi IP dei suoi amici



Trasferimento di file: il protocollo FTP



- Trasferimento di file da/a un terminale remoto
- modello client/server
 - o client: lato che inizia il trasferimento (da/a remoto)
 - o server: terminale remoto
- □ FTP: RFC 959
- FTP server: porta 21

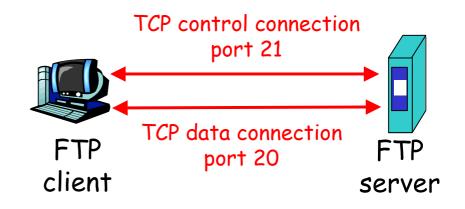


Il protocollo FTP: controllo separato, connessioni di dati

- □ Il client FTP contatta il server FTP alla porta 21, specificando il TCP come protocollo di trasporto
- due connessioni TCP parallele aperte:
 - controllo: scambiano comandi, risposte fra client e server.

"out of band control"

- o dati: da/a server
- il server FTP mantiene lo "stato": directory corrente, previa autenticazione





Il protocollo FTP: comandi e risposte

Esempi di comando:

- inviati come testo ASCII sulla connessione di controllo
- □ USER username
- □ PASS password
- Ls restituisce l'elenco dei file nella directory corrente
- ☐ GET filename ritrova il file nel terminale remoto
- ☐ PUT filename memorizza il file sul terminale remoto

Esempi di codici di ritorno:

- codice e frase di status
- □ 331 Username OK, password required
- 125 data connection already open; transfer starting
- □ 425 Can't open data connection
- ☐ 452 Error writing file

Il protocollo FTP: una sessione di esempio

Connessione con un terminale remoto

Directory locale ora C:\WINDOWS\Desktop.

```
ftp> open <hostname>
Connesso a <hostname>.

220 <hostname> FTP server (UNIX(r) System V Release 4.0) ready.
Utente (<hostname> :(none)): lisi
331 Password required for lisi.
Password:
230 User lisi logged in.
ftp> lcd
```

• • •



Il protocollo FTP: una sessione di esempio (cont.)

Richiesta di elenco file e cambio directory in remoto

```
ftp> Is -I
200 PORT command successful.
150 ASCII data connection for /bin/ls (193.204.187.247,1375)
  (0 bytes).
total 2
drwxr-xr-x 4 lisi nobody 512 Oct 24 10:59 public_html
226 ASCII Transfer complete.
76 bytes received in 0.00secondi 76000.00 Kbytes/sec)
ftp > cd public_html
250 CWD command successful.
```



Il protocollo FTP: una sessione di esempio (cont.)

Un'altra richiesta di elenco file in remoto

```
ftp> Is -I
200 PORT command successful.
150 ASCII data connection for /bin/ls (193.204.187.247,1378)
  (0 bytes).
total 24
drwxr-xr-x 5 lisi nobody 512 Feb 25 16:34 courses
-rw-rw-rw- 1 lisi nobody 10185 Oct 10 14:14 lisi.jpg
drwxr-xr-x 2 lisi nobody 512 Oct 10 14:13 publications
226 ASCII Transfer complete.
205 bytes received in 0.22secondi 0.93 Kbytes/sec)
```



Il protocollo FTP: una sessione di esempio (cont.)

Richiesta di ritrovamento file sul terminale remoto

```
ftp> binary
200 Type set to I.
ftp> get lisi.jpg
200 PORT command successful.
150 ASCII data connection for lisi.jpg (193.204.187.247,1379)
  (10185 bytes).
226 ASCII Transfer complete.
10220 bytes received in 0.00secondi 10220000.00 Kbytes/sec)
ftp> close
221 Goodbye.
```



Sommario della prossima lezione: Lo strato di applicazione (2/5)

- Principi dei protocolli dello strato di applicazione
- Trasferimento di file & il protocollo FTP
- World Wide Web & HTTP
- □ Posta elettronica & SMTP
- DNS: il servizio directory di Internet
- Condivisione di file
- Programmazione dei socket

