Client-Server Paradigm

- Conceptual basis for many distributed applications
- Based on premise that a program makes a request to which another responds
- Related to remote procedure call mechanism
- Client
  - Any application program
  - Makes a request
  - Awaits a response
- Server
  - Specialized program (process)
  - Awaits a request
  - Computes an answer
  - Issues a response

Examples Of Standard Servers

- Date and time of day server
- Domain name server
- Electronic mail server
- Remote login server
- File server
- Finger server
- ICMP echo
- UDP echo
- Information server

Widely Used Application Level Internet Services

- Remote login
- Electronic mail
- File transfer
- World Wide Web

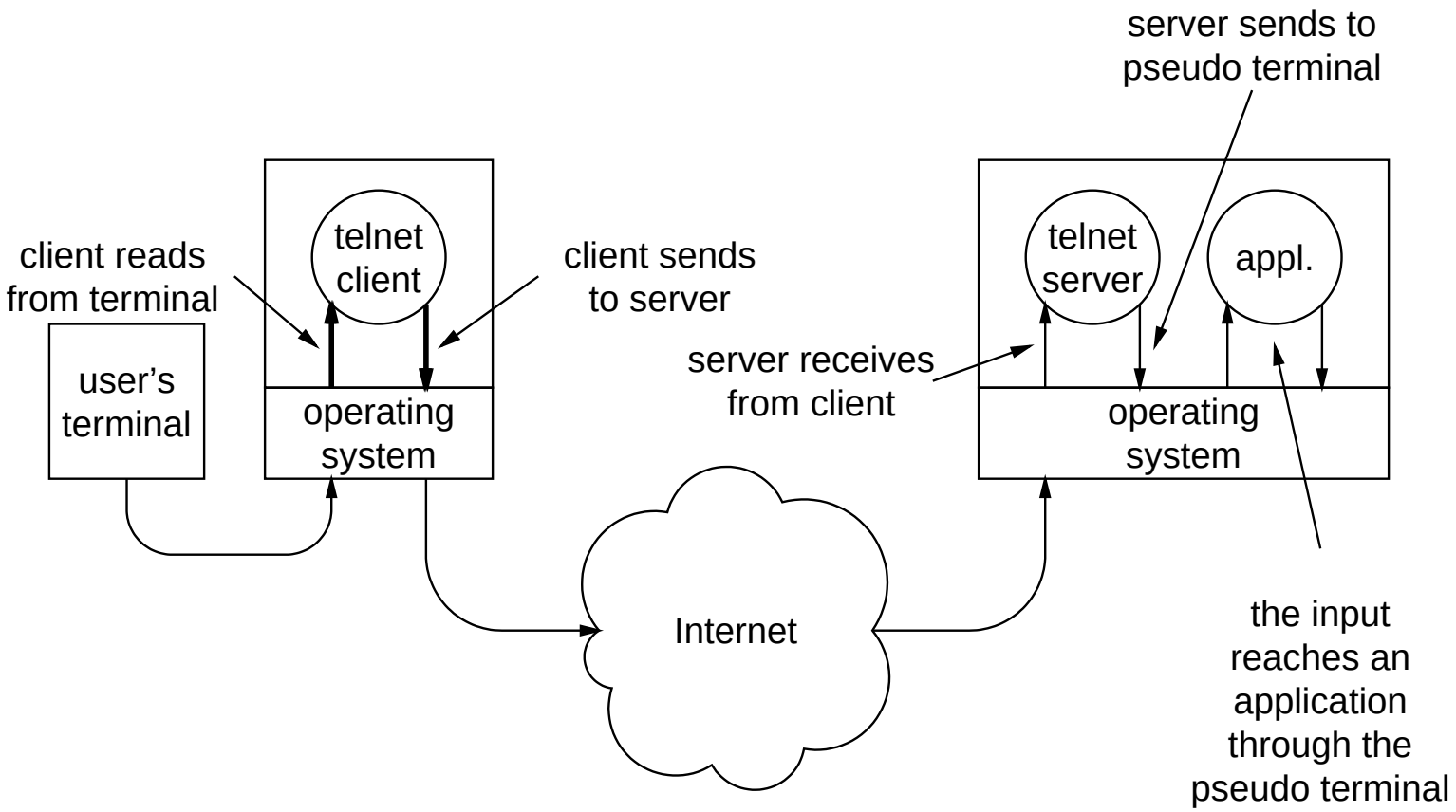
Remote Login

- Client
  - User invokes client program on local machine
  - Client program
    - Forms connection to server
    - Passes keystrokes to server
    - Displays output from server on terminal
- Server
  - Requires operating system primitive that supplies "pseudo terminal" to running program
  - Master server invoked at system startup
  - Master creates slave to handle each new connection
  - Slave relays characters between TCP connection and pseudo terminal

TELNET: Internet Standard Remote Terminal Protocol

- Basic facility
- Uses TCP connection to pass data
- Supports options:
  - Character set translation
  - Line terminator
  - Many others

Illustration of TELNET



- Server process shown is the slave for this connection
- The application is usually a command interpreter

Rlogin (UNIX Remote Login)

- From 4BSD UNIX
- Passes environment information (e.g., terminal type) to remote machine
- Understands equivalence of login ids
- Preserves UNIX standard input, standard output, standard error
- Nonstandard (limited to UNIX)

Rsh (UNIX Remote Shell)

- From 4BSD UNIX
- Allows remote command execution
- Honors equivalence of login ids
- Preserves UNIX standard input, standard output, standard error
- Nonstandard (limited to UNIX)

Example Of Rsh

- Simple Example `rsh ace.cs.ohiou.edu ps`
  - Specifies machine *ace* in domain *cs.ohiou.edu*
  - Executes command *ps*
  - Displays output on user's terminal
- Local redirection
  - Can easily specify remote execution with output to local file

    `rsh ace.cs.ohiou.edu ps > x`
- Remote redirection
  - Can just as easily specify remote execution with output to remote file

    `rsh ace.cs.ohiou.edu 'ps > x'`

SSH - Better Rsh

- Of course, ssh is a much better alternative to rsh
  - Encrypted streams
  - Improved authentication
- Has all the same functionality of rsh

File Transfer

- Bulk data transfer
- Two standard protocols exist
  - FTP for large transfer
  - TFTP for simpler tasks
- Follows client-server model

File Transfer Protocol (FTP)

- Defined early in history of Internet
- Heavily used
- Uses TCP for transport
- Provides ability to list directories as well as transfer files
- Client can send or receive files
- Third party transfer supported
- Character translation supported
- Basic file formats defined

FTP Details

- Client Side
  - Makes a TCP connection to well-known port for control
  - Obtains authorization by sending a login id and password
  - Sends requests to server over control connection
- Server Side
  - Waits at well-known port
  - Uses TELNET for control connection
  - Creates slave to handle each request
  - Makes new TCP connection for each transfer
  - Enforces file access protections

FTP Example

```
% ftp
ftp> open bigbird.cs.ohiou.edu
Connected to bigbird.cs.ohiou.edu
220 bigbird.cs.ohiou.edu FTP server ready.

Name (bigbird:sdo): anonymous
331 Guest login ok, send ident as password.
Password: sdo@ostermann.cs.ohiou.edu
230 Guest login ok, access restrictions apply.

ftp>  ls
200 PORT command successful.
150 Opening data connection for /bin/ls
    (192.5.48.3,1058) (0 bytes).
file1
file2
bigfile
226 Transfer complete. 20 bytes received
    in 0.2 seconds (0.072 Kbytes/s)
```

FTP Example (continued)

```
ftp> get
(remote-file) file2
(local-file) junk
200 PORT command successful.
150 Opening data connection for file2
   (192.5.48.3,1061) (133120 bytes).
226 Transfer complete. 137888 bytes
   received in 3.7 seconds (36 Kbytes/s)

ftp>  close
221 Goodbye.

ftp>  quit
```

FTP Communication Details
Active Connection

user types	localhost sends	ace sends
ftp ace		
	3-way handshake	
		220 *welcome message*
username		
	USER anonymous	
		230 Guest login ok, send your complete e-mail address as password.
password		
	PASS sdo@ohiou.edu	
		230 Guest login ok, access restrictions apply.
ls		
	PORT 132,235,3,128,179,186	
		server connects to port 179,186 from port ftp-data
		200 PORT command successful.
	NLST	
		150 Opening ASCII mode data connection for file list.
		file transferred, connection closed
		226 Transfer complete.
quit		
	QUIT	
		221 Goodbye.
	close connection	

FTP Communication Details
Passive Connection

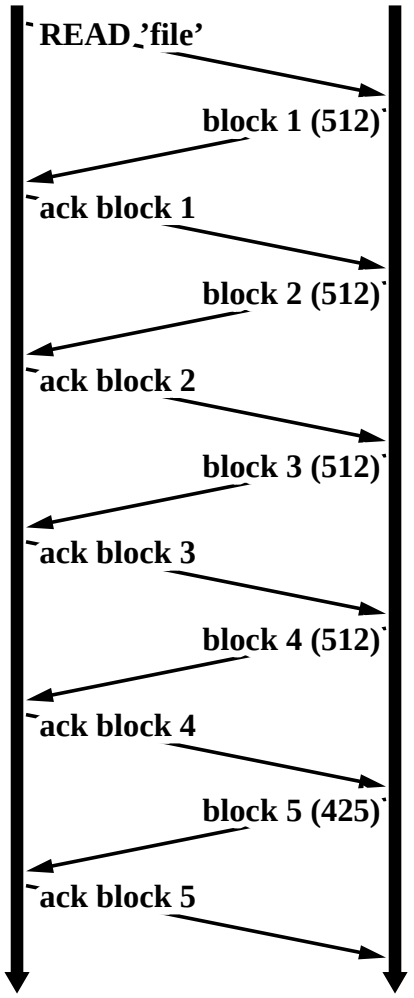
user types	localhost sends	ace sends
		... same as active case
ls		
	PASV	
		227 Entering Passive Mode (132,235,1,2,13,96)
	NLST	
	client connects to port 13.96 on server	
		file transferred, connection closed
		226 Transfer complete.
		... same as active case

Trivial File Transfer Protocol (TFTP)

- Internet standard
- Alternative to FTP for file transfer only
- Built on UDP
- Uses global file access rights
- Can be implemented in ROM
- Useful for bootstrap
- Useful on small machines

TFTP Example

- Retrieval of 2454-byte file



Network File System (NFS)

- Internet file access standard
- Defined by SUN Microsystems, Inc.
- Allows one computer to mount files from another in its directory system
- Application program cannot distinguish local and remote files
- Built on top of SUN's RPC system
- Uses UDP to transfer messages between NFS clients and servers

Electronic Mail

- Memos delivered from one user to another
- Spooling system accepts outgoing mail and keeps a copy until delivery possible
- Periodic retry (automated)
- Separate standards for mail message format and mail transfer
- High degree of interoperability

Mail Format (RFC 822)

- Internet standard
- Used by other groups
- Blank line separates header and body
- Specifies headers of the form

  `KEYWORD: information`

Mail Format (continued)

- Only a few keywords are important
  - To:
  - From:
- Other keywords are optional
  - Received-by:
  - Reply-to:
- Others are allowed but not interpreted
  - Environmental-Impact: ...

Mail Transfer (SMTP)

- Internet standard
- Specifies transfer from machine to machine
- Uses TCP for stream connection

Electronic Mail Example

- User Smith at machine ace.cs.ohiou.edu sends a single mail message to three recipients:
  - Jones@ohio.edu
  - Green@ohio.edu
  - Brown@ohio.edu
- Smith's mail system stores message in spool area and initiates program to transfer message in background
- Background program uses domain name system to map the name *ohio.edu* into an Internet address
- Background program (client) contacts SMTP server at well-known port on machine ohio.edu using TCP

Electronic Mail Example (continued)

```
S: 220 ohio.edu Simple Mail Transfer Service Ready
C: HELO ace.cs.ohiou.edu
S: 250 ohio.edu
C: MAIL FROM:<Smith@ace.cs.ohiou.edu>
S: 250 OK

C: RCPT TO:<Jones@ohio.edu>
S: 250 OK

C: RCPT TO:<Green@ohio.edu>
S: 550 No such user here

C: RCPT TO:<Brown@ohio.edu>
S: 250 OK

C: DATA
S: 354 Start mail input; end with <CR><LF>.<CR><LF>
C: ...sends body of mail message...
C: ...continues for as many lines as message contains
C: <CR><LF>.<CR><LF>
S: 250 OK

C: QUIT
S: 221 ohio.edu Service closing transmission channel
```

Electronic Mail Example (continued)

- Client and server both close TCP connection
- Server delivers message to Jones' and Brown's mailboxes
- Client (background program) removes the message from the spooling area on ace.cs.ohiou.edu

X Window System

- Manages multiple, independent windows on a bit-mapped display in a networked environment
- Supports both text and graphics
- Unusual use of client-server paradigm

X Window System
(client-server details)

- Server executes on machine with bit-mapped display
  - On workstation under conventional operating system (e.g., UNIX)
  - On special-purpose X terminal with embedded system
- Clients operate on arbitrary hosts
- Special client called *window manager* handles display details, including borders and pop-up menus

X Window System
(interesting client-server twist)

- Server executes locally (on user's workstation)
- Clients execute remotely
- To create a window containing a login shell for a remote machine, the user must initiate a connection from a client on the remote machine to the server
- User can use *rsh* or *TELNET* to reach remote machine and start a client
- Default remote login client is called *xterm*