Chapter 28
Client/Server Computing

Network Services

- Protocols provide communication services.
- (User-level) processes are the entities that invoke the services.
  - Servers are user-level processes that offer network services
  - Clients are user-level processes that use server's services
- Clients and Servers communicate via application-level protocols (i.e., the protocol depends on the application/service)
- Example services include things like:

Client-Server Paradigm

- Server application is the “listener”. Repeatedly

- Client application contacts the server to request a service

Transport protocols and client-server paradigm

- Clients and servers exchange messages using the network software on their machine (typically in the Operating System) call the "protocol stack") e.g., TCP or UDP
- Both client and server must have same protocol stack.

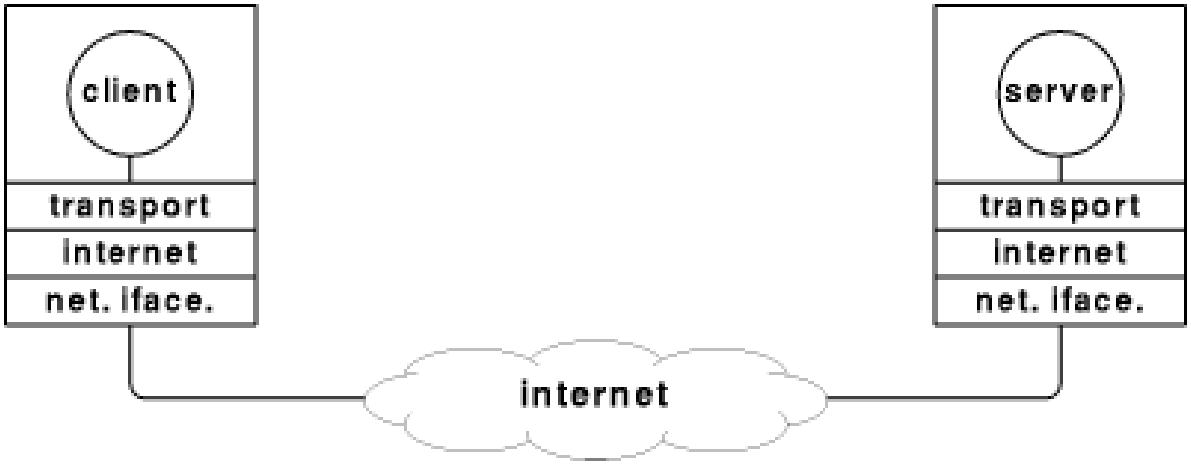

Figure 1: Communication via Protocols Stacks

Message Exchange

- Typically, client and server exchange 2 messages (i.e., in pairs):
  - Client sends request message.
  - Server sends response message.
- Other variations are possible:
  - Client send 1 request msg, gets lots of responses (e.g., database query).
  - Client sends 1 request msg, gets no response
  - Client sends 1 request msg, gets continuous stream of response messages (e.g., registering or weather report updates)

Characteristics of a Client

- Arbitrary application program
  - Also performs other computations
  - Becomes client when network service is needed
- Invoked directly by user
- Runs locally on user's computer
- Initiates contact with server
- Can access multiple services (one at a time)
- Does not require special hardware or sophisticated operating system

Characteristics of Server

- Special purpose application dedicated to providing network service
- Typically starts at system initialization time
- Runs on a server class machine (often a centralized, shared computer)
- Waits for service requests from clients; loops to wait for next request
- Will accept requests from arbitrary clients; provides one service to each client
- Requires powerful hardware and sophisticated operating system

"Server-class" Computers

- Shared, centralized computers that run many server applications are sometimes called "servers"
- More precisely, the applications are the "servers" and the computer is a "server-class computer"
- Note that servers do not have to be run on server-class computers. Servers can run on very simple computers (e.g., World's Smallest Web Server as of Aug 1999 - see http://www-ccs.cs.umass.edu/%7Eshri/iPic.html)

Identifying/Locating a Service

- Problem : How does a client connect to a service? There are two subproblems:
  - Identification
  - Location
- In the Internet, each service is given a unique identifier; namely, a protocol port number. Port numbers are
  - uniform throughout the Internet
  - set by standards bodies
- Server
  - Informs Operating System it is using port P
  - Waits for request to arrive on port P
- Client
  - Forms request message
  - Sends request to port P on server computer

Selecting from Multiple Servers

- What if two (or more) instantiations of the same service are running on the same computer? How do incoming messages get delivered to the correct server?
- In TCP, each transport session has two unique identifiers
  - (IP address, port number) on server
  - (IP address, port number) on client
- No two clients on one computer can use same source port
- Thus, client endpoints are unique, and server computer protocol software can deliver messages to correct server process

Connection-oriented and Connectionless Transport

- Which to choose?
- TCP - connection-oriented
  - Client establishes connection to server
  - Client and server exchange multiple messages of arbitrary size
  - Client terminates connection
- UDP - connectionless
  - Client constructs message
  - Client sends message to server
  - Server responds
  - Message must fit in one UDP datagram
- Some services use both
  - e.g., DNS
  - Can be provided by single server
- common services found in /etc/services on UNIX systems