COMP476
Networked Computer Systems

Client/Server Programming

Establishing Contact through IP

- *Listening* application informs local protocol software that it is ready to accept incoming messages
- *Connecting* application uses internet protocol to contact listener
- Applications exchange messages through resulting connection

Client-Server Paradigm

- *Server* application is “listener”
  - Waits for incoming message
  - Performs service
  - Returns results
- *Client* application establishes connection
  - Sends message to server
  - Waits for return message

Characteristics of Client

- Arbitrary application program
  - Becomes client when network service is needed
  - Also performs other computations
- Invoked directly by user
- Runs locally on user's computer
- Initiates contact with server
- **Must know (or find) the address of server**
- Can access multiple services

Characteristics of Server

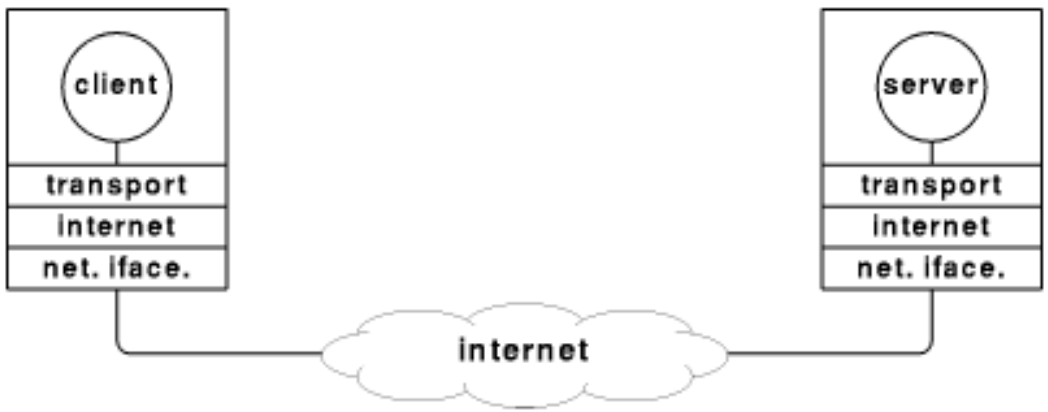
- Special purpose application dedicated to providing network service
- Starts at system initialization time
- Runs on a remote computer (usually centralized, shared computer)
- Waits for service requests from clients; loops to wait for next request
- Will accept requests from arbitrary clients; provides one service to each client

Message Exchanges

- Typically, client and server exchange messages:
  - Client sends request, perhaps with data
  - Server send response, perhaps with data
- Client may send multiple requests; server sends multiple responses
- Server may send multiple response - consider streaming audio

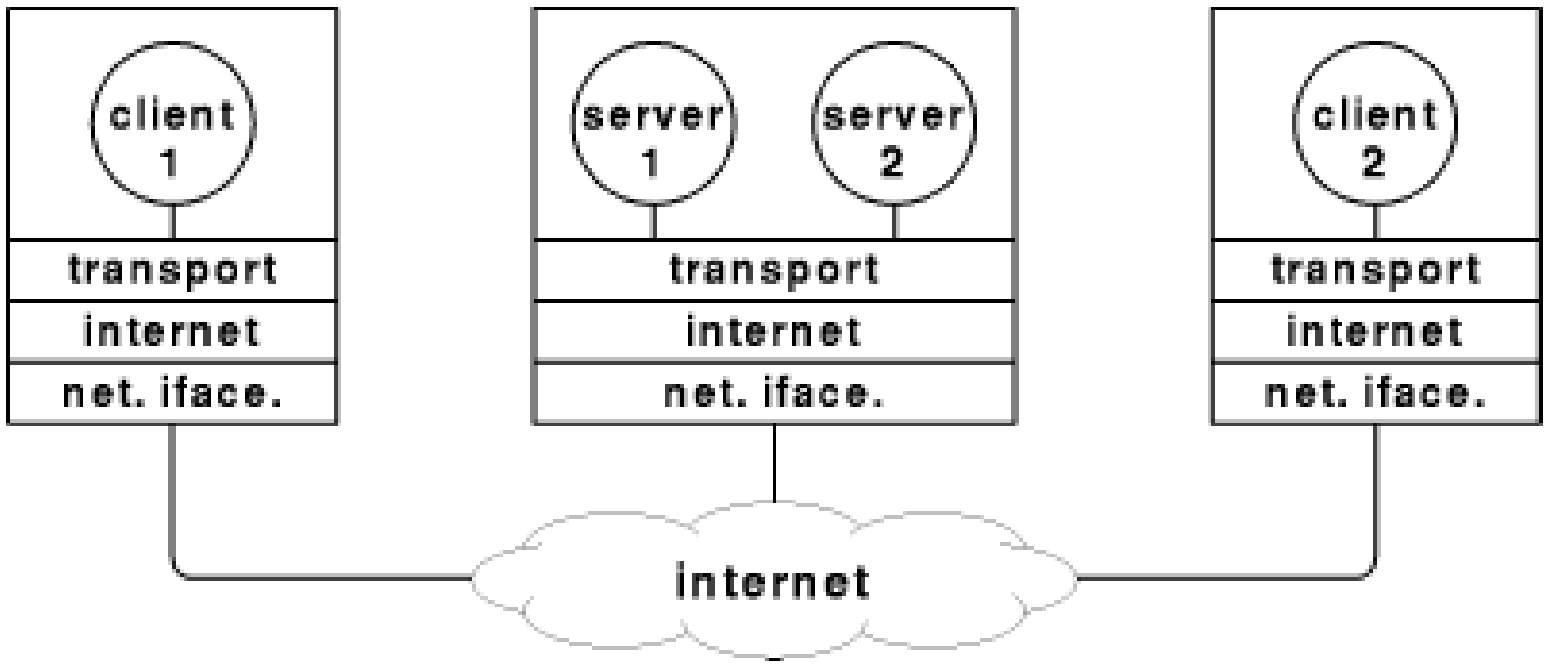
Transport Protocols and Client-Server Paradigm

- Clients and servers exchange messages through transport protocols; e.g., TCP or UDP
- Both client and server must have same protocol stack and both interact with transport



Multiple Services on One Computer

- Servers run as independent processes and can manage clients simultaneously



Multiple Services on One Computer

- Can reduce costs by sharing resources among multiple services
- Reduces management overhead - only one computer to maintain
- One server can affect others by exhausting computer resources
- Failure of single computer can bring down multiple servers

Selecting from Multiple Servers

- How do incoming messages get delivered to the correct server?
- 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

Binding of Server Location

- How and when does a client application learn the location of a service?
- Goals:
  - portability - allow the application to be used on different systems
  - load balancing - select server with lowest utilization
  - failure recovery - select a different server if original fails
  - efficiency – avoid many messages or broadcasts

Finding a Service

- Write server name in code
- Read a file of server addresses
- Broadcast request for a server
- Ask a human
- Name server
- UDDI

- **Much more on this topic later on**

Client-Server Interactions

- Clients can access multiple services
- Clients may access different servers for one service
- Servers may become clients of other servers

Summary

- *Client-server* paradigm used in almost every distributed computation
  - *Client* requests service when needed
  - *Server* waits for client requests
- Clients and servers use *transport protocols* to communicate
- Often, but not always, there is an *application protocol*