Operating System Design

Dr. Jerry Shiao, Silicon Valley University

Distributed System Structures v/s Centralized System

- Overview
  - □ Traditional Data Processing Function organized in a centralized data processing facility (Computers, Processing, Data).
- What is a Distributed System?
  - □ Collection of Processors that do not share memory or a clock.
  - □ Processor has its own local Memory.
  - □ Processors communicate through communication networks.
  - □ Advantages: Responsiveness from local computing facilities, Availability with multiple interconnected systems, Resource sharing, incremental growth, and increased end-user productivity.
- Types of Network-Based Operating Systems

  ↑ Network OS & Distributed OS
- Network Structure
- Network Topology
- Communication Structure
- Communication Protocols
- Robustness
- Design Issues

Distributed System Structures

- Distributed system is collection of loosely coupled processors interconnected by a communication network.
  - □ Processors varies in size and function.
  - □ Processors include workstations, minicomputers, and large General-Purpose Computer Systems.
  - □ Site: Location of a Computer System. ← building, location.
  - □ Host: Referring to specific system at a site.
  - □ Host on one site, Server, has a resource another host at another site, the client (or user), would like to use it.

Distributed System Structures

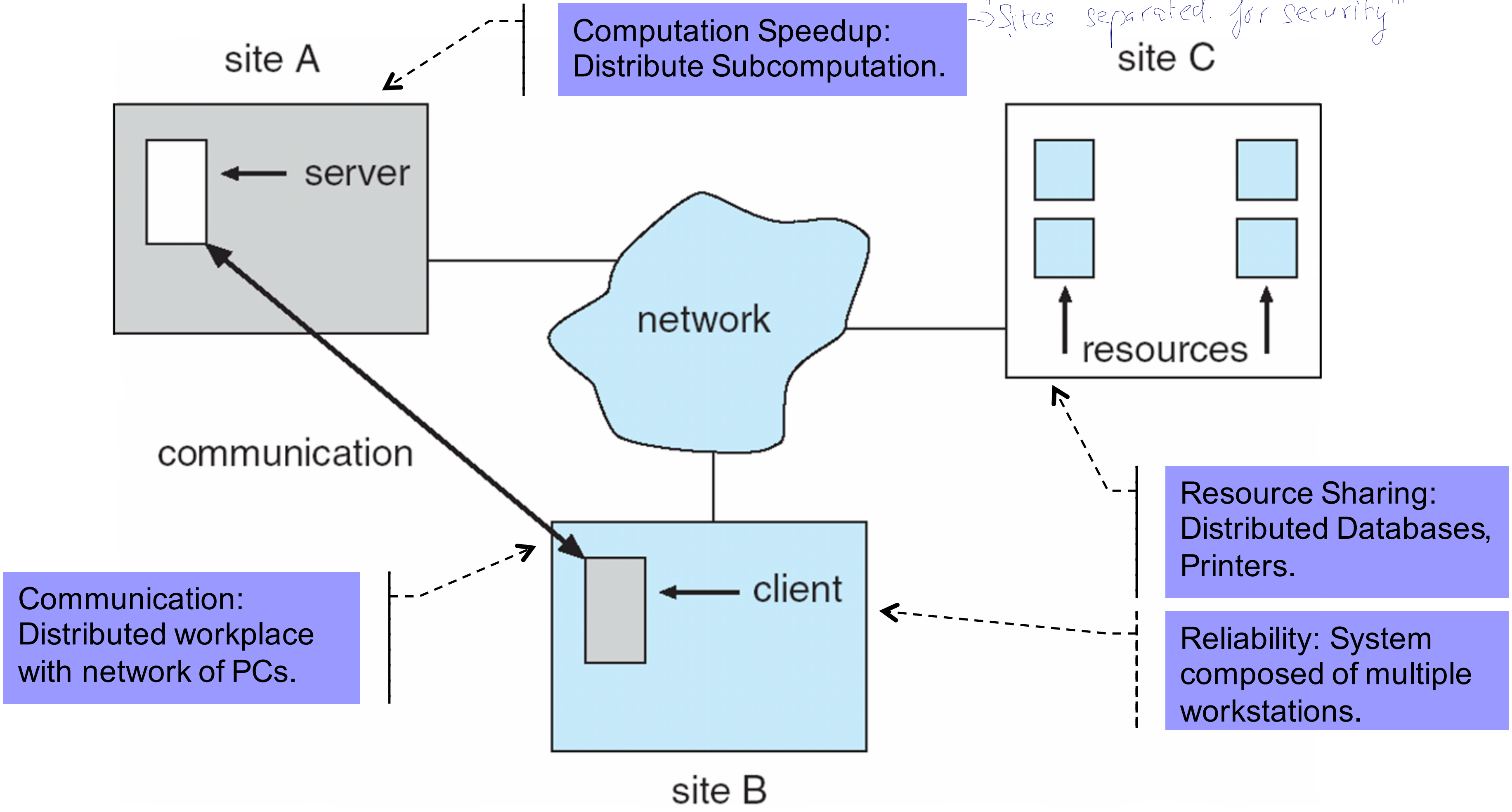
- Motivation: Four major reasons for Distributed Systems:
  - □ Resource Sharing:
    - Sharing and printing files at remote sites.
    - Processing information in a distributed database.
    - Using remote specialized hardware devices.
  - □ Computation speedup:
    - Computation partioned into subcomputations that can run concurrently.
    - Load Sharing: Site overloaded with jobs, jobs moved to lightly loaded sites.
  - □ Reliability:
    - System composed of autonomous installations, failure of one site does not affect the rest.
    - Detect and recover from site failure, function transfer, reintegrate failed site.
  - □ Communication:
    - Message passing provides functions of a stand-alone system expanded to a distributed system, using file Xfer, remote login, mail, and RPC calls.
    - Project work does not have to local, but distributed using network of workstations: Expanded facilities.

Distributed System Structures

- Distributed System

Question: A distributed system is a collection of loosely coupled processors interconnected by a com network From the point of view of a specific ... which is FALSE for distributed systems

→ "Sites separated for security"



Computation Speedup: Distribute Subcomputation.

Resource Sharing: Distributed Databases, Printers.

Communication: Distributed workplace with network of PCs.

Reliability: System composed of multiple workstations.

Distributed System Structures

- Types of Network-Based Operating Systems
- Network Operating Systems: Users aware of networked systems access the remote resources by Login or transfer files from remote to local system.
  - □ Remote logging into the appropriate remote machine:
    - telnet, ssh Utilities.
    - Remote shell process commands on behalf of user.
  - □ Remote Desktop (Microsoft Windows):
    - Access desktop GUI interface remotely.
  - □ Transferring data from remote machines to local machines:
    - File Transfer Protocol (FTP) mechanism.
    - Secure Copy (scp) using ssh utility for authentication and encryption.
    - Secure FTP (sftp) using ssh utility for authentication and encryption.
  - □ User must change paradigms:
    - User must know FTP, scp, sftp, telnet, or ssh command sets.
    - Windows user using ssh or telnet to UNIX system must know UNIX commands.

not transparent

→ unlike distributed OS system (next slide)

Distributed System Structures

- Types of Network-Based Operating Systems
- Distributed Operating Systems
- Users not aware of multiplicity of machines:
  - Access to remote resources similar to access to local resources.
  - Data and process migration to another site controlled by distributed OS.
- Data Migration:
  - Transfer data by transferring entire file (useful for large files).
  - Transferring only portions of the file necessary for the immediate task.
  - Data translation, between character sets or integer bit ordering.
- Computation Migration:
  - Transfer the computation, rather than the data, across the system.
  - Time to transfer data longer than execution of the job.
  - Using RPC to execute predefined procedure at remote site and return results.
  - Using messages between system to request remote process execution and return results.

Distributed System Structures

- Types of Network-Based Operating Systems
- Distributed Operating Systems (Cont)
- Process Migration: Process or parts of process executed at different sites due to:
  - □ Load Balancing: Process or subprocesses distributed workload.
  - □ Computation Speedup: Concurrently execution of subprocesses.
  - □ Hardware Preference: Specialized processor (array processor operating on arrays or vectors of data).
  - □ Software Preference: Software available at select site.
  - □ Data Access: Large amounts of data being used in computation, more efficient to run remotely than transfer the data.

via messaging or RPC

- Two techniques to move processes in network.
  - □ OS hides migration: Computer system achieving load balancing and computational speedup among homogeneous systems, no user input.
  - □ User specify migration: Require user to specify process migration to satisfy hardware or software preference.

Distributed System Structures

- Types of Network-Based Operating Systems
- Distributed Operating Systems (Cont)
- World Wide Web:
  - WWW has many aspects of a distributed-computing environment.
  - Data Migration between Web Server and Web Client.
  - Computation Migration:
    - Web Client trigger database operation on Web Server.
  - Process Migration:
    - Java Applets sent from Web Server to Web Client for execution.

Teacher

Q.n: not behavior of Distributed OS.

ans: (c) Desktop Migration used by OS to transfer GUI, etc...

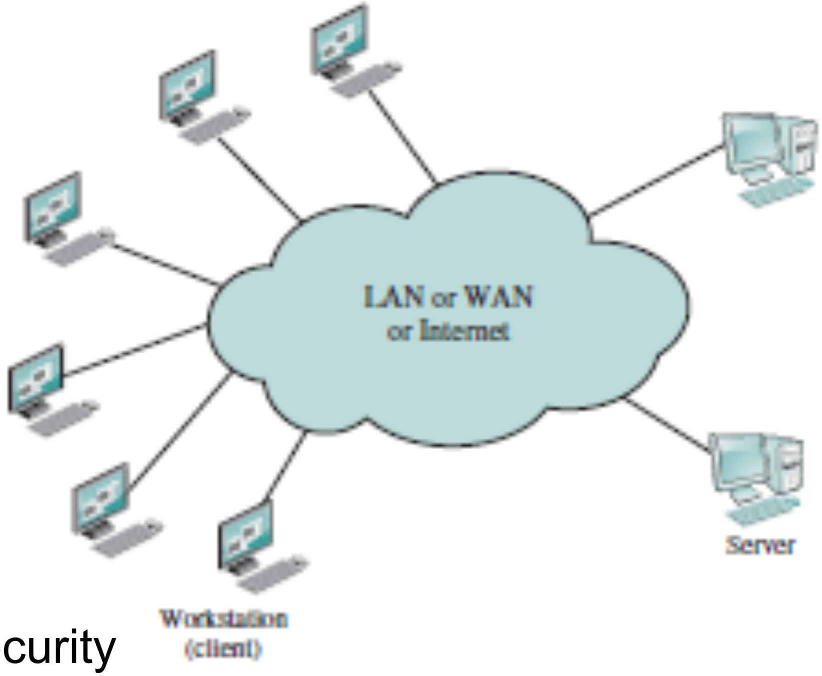
↑

is not mentioned in the slides.

Distributed System Structures

- Types of Network-Based Operating Systems
- Client / Server Computing: Distinct Characteristics:
  - □ Heavy reliance on user-friendly applications to the client system.
  - □ Server provides set of shared services to the clients.
  - □ Most common is database server, allowing many clients to share access to same database.
  - □ Allows Management to maintain control of the computing and information systems.
  - □ User organizations, application, and equipment vendors must provide interoperable products and equipment.
  - □ Networking fundamental to operations. Network Management and Network Security have a high priority.

there is some type of interface; GUI



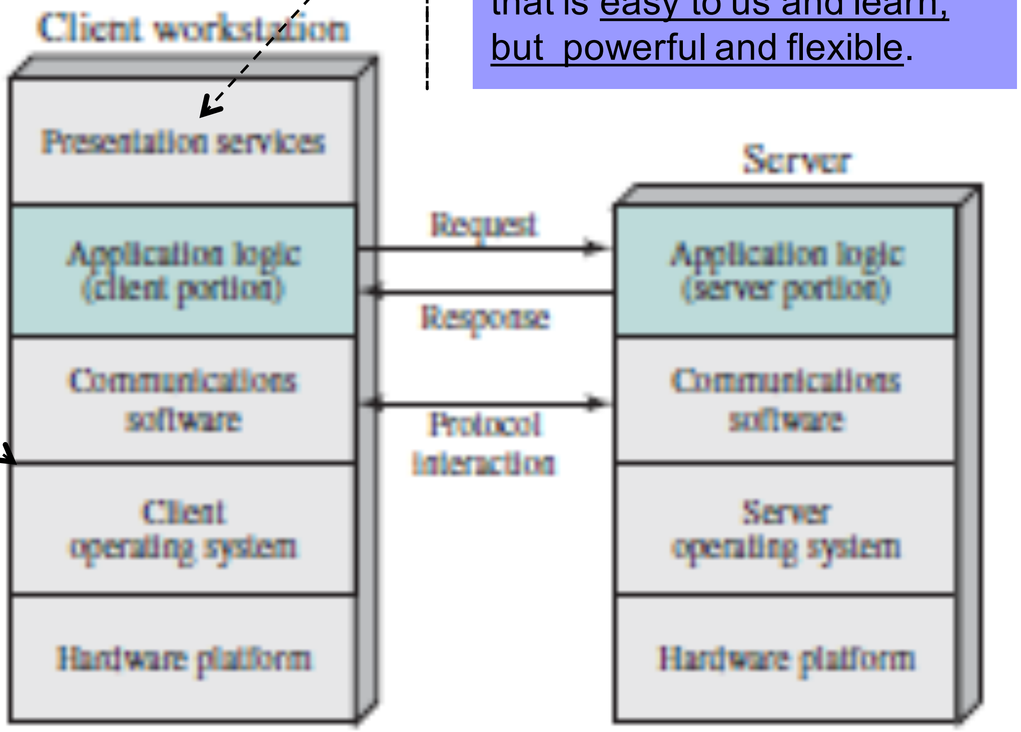
Distributed System Structures

- Types of Network-Based Operating Systems
- Client / Server Computing:
  - Key feature is allocation of application-level tasks between Client and Server.

Client and Server must share same communication protocol and support same application.

Different Operating Systems or different Client / Server Systems.

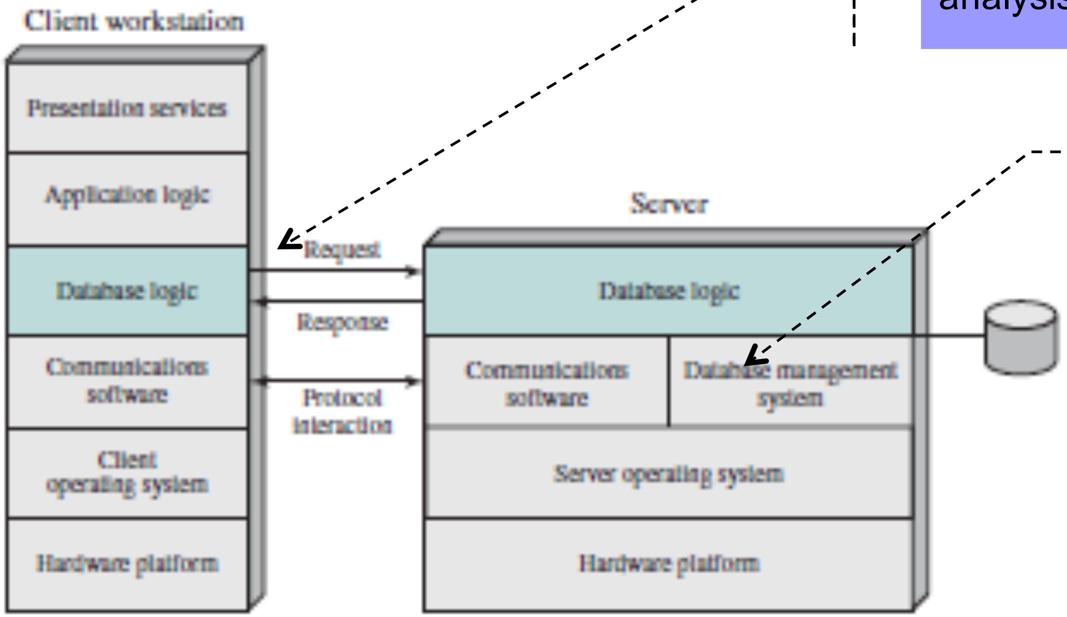
Emphasis on providing a Graphic User Interface (GUI) that is easy to us and learn, but powerful and flexible.



does not matter

Distributed System Structures

- Types of Network-Based Operating Systems
- Client / Server Computing: Client-based Processing
  - Most common, Relational Database.
  - Server responsible for maintaining the database.
  - SQL (Structured Query Language) ties Client and Server: Enables Client to make requests for access to the Server's Database.



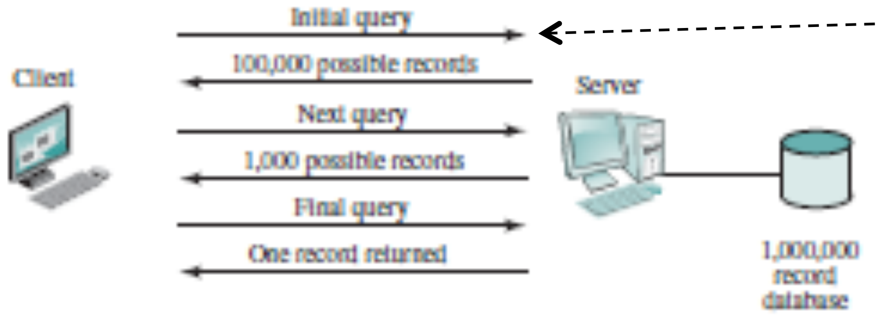
Client is responsible for processing or data analysis.

Server is responsible for managing the Database. The process of sorting and searching the database requires large disk with high-speed CPU and high-speed I/O architecture.

Cost of high-end equipment focused at the Server.

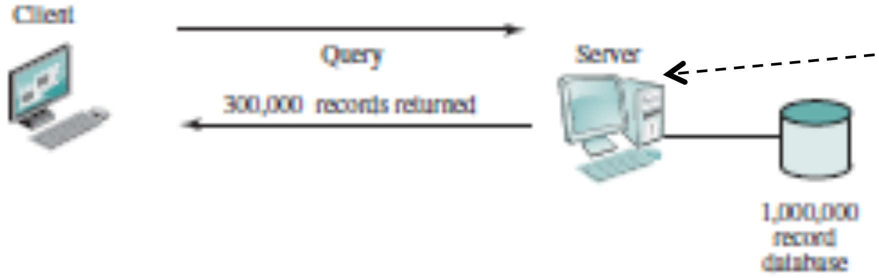
Distributed System Structures

- Types of Network-Based Operating Systems
- Client / Server Computing: Client-based Processing



(a) Desirable client/server use

Client requests involves multiple steps.



(b) Misused client/server

Possible problem when large number of records are returned.

Large traffic burden to transfer results from Server to Client.

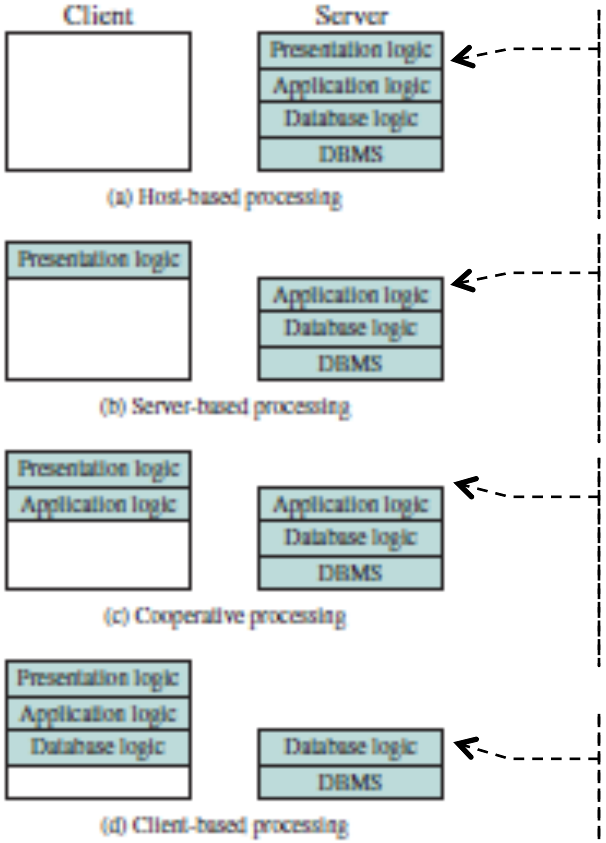
Solution would require Server to have some or all the Application Logic.

Distributed System Structures

TQn: Which is server based processing?
Ans: Client responsible for GUI, server do all the work.

- Types of Network-Based Operating Systems
- Client / Server Computing: Client/Server Application Classes

Eg: command line /ssh.



(a) Host-based processing

(b) Server-based processing

(c) Cooperative processing

(d) Client-based processing

Host-based Processing: Traditional mainframe environment where all processing is done at the host (Server).
User interface is a dumb terminal or PC acting as terminal emulator.

Server-based Processing: Basic Client/Server configuration where Client provides us-friendly GUI. All processing is done at the Server and the databases and applications can easily be maintained on Server.

Cooperative Processing: Application processing performed at the Client and the Server. More complex to setup and maintain, but offers better user performance and better networking. Sybase PowerBuilder and Gupta Corp SQL.

Client-based Processing: Client handles all application processing. Most common Client/Server approach. Enables user to employ applications tailored to local needs.
Server handles data validation routines and database logic function.

Distributed System Structures

- Types of Network-Based Operating Systems
- Client / Server Computing: Client/Server Application
- Fat Client Model: Cooperative Processing and Client-based Processing.
  - Sybase Inc's PowerBuilder and Gupta Corp's SQL Windows.
  - Advantages: Uses desktop processing power and less bottleneck at the server because offloaded app processing.
  - Disadvantages: High volume traffic need high-capacity LANs, and difficult to maintain, upgrade, or replace apps distributed over hundreds of desktops.

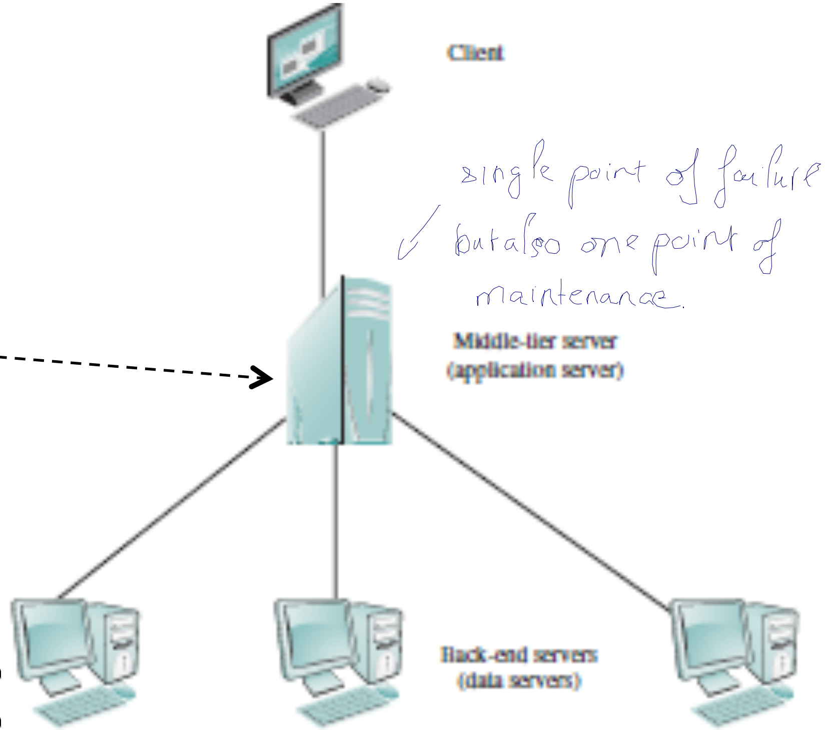
client has more functions

- Thin Client Model: Server-based Processing.
  - Traditional host-centric approach, usually the migration path from mainframe to a distributed computing environment.
  - Server performs app processing and will be the bottleneck.

Distributed System Structures

- Types of Network-Based Operating Systems
- Client / Server Computing: Three-Tier Architecture
  - Application Software distributed among:
    - User Machine: Thin Client.
    - Middle-tier Server: Gateways between Thin Client and variety of backend Database Servers. Convert protocols and map one type of database query to another.
    - Backend Server: Database Application.

Middle-tier system acts as both client and server.



single point of failure

✓ but also one point of maintenance.

Distributed System Structures

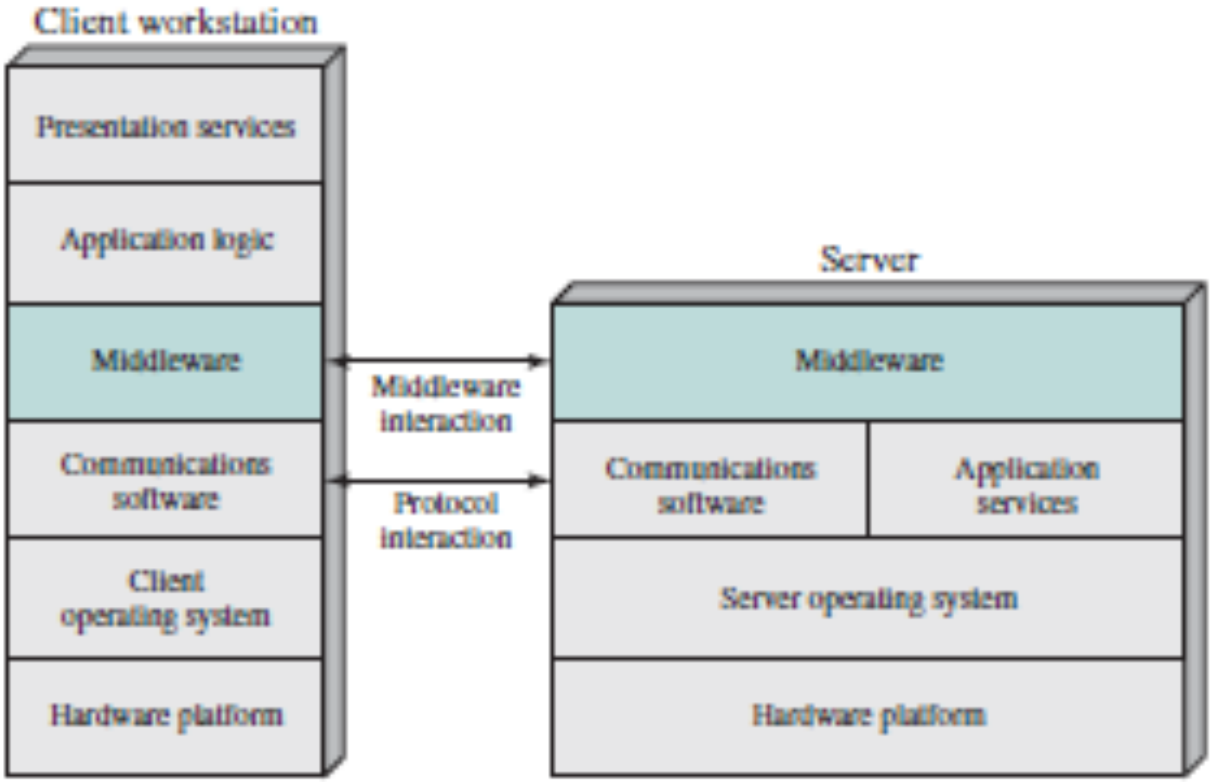
- Types of Network-Based Operating Systems
- Client / Server Computing: Middleware
- Standards to integrate multivendor and enterprise-wide Client/Server configuration.
  - □ Developers need set of tools that provide a uniform means and style of access to system resources.
  - □ Middleware:
    - Standard programming interface between the client application and network communications software and Operating Systems.
    - Allow access to variety of services on servers without being concerned about differences among servers.
    - Client does not need to know data is stored in a SQL database or in an Oracle database and the commands to access the database.
  - □ Clients (Users) will use standardized interfaces (Middleware) and assemble the equipment from different vendors that support the interface.

Distributed System Structures

- Types of Network-Based Operating Systems
- Client / Server Computing: Middleware

User writes their own logic which talks to the middleware.

Middleware: Hide the differences between Operating Systems and Network Protocols.



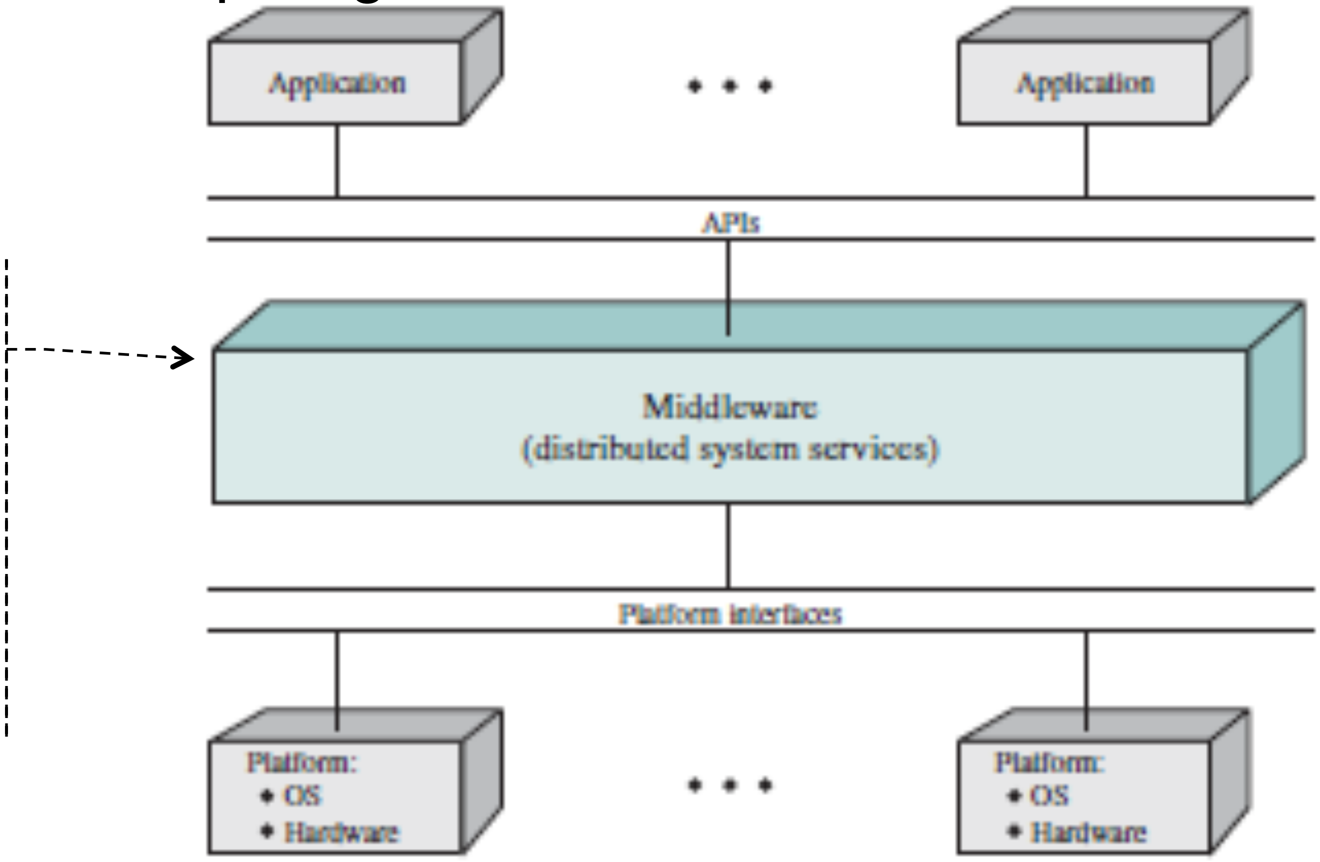
Middleware in Client / Server Architecture

Distributed System Structures

TQn. best describe Middleware

- Types of Network-Based Operating Systems
- Client / Server Computing: Middleware

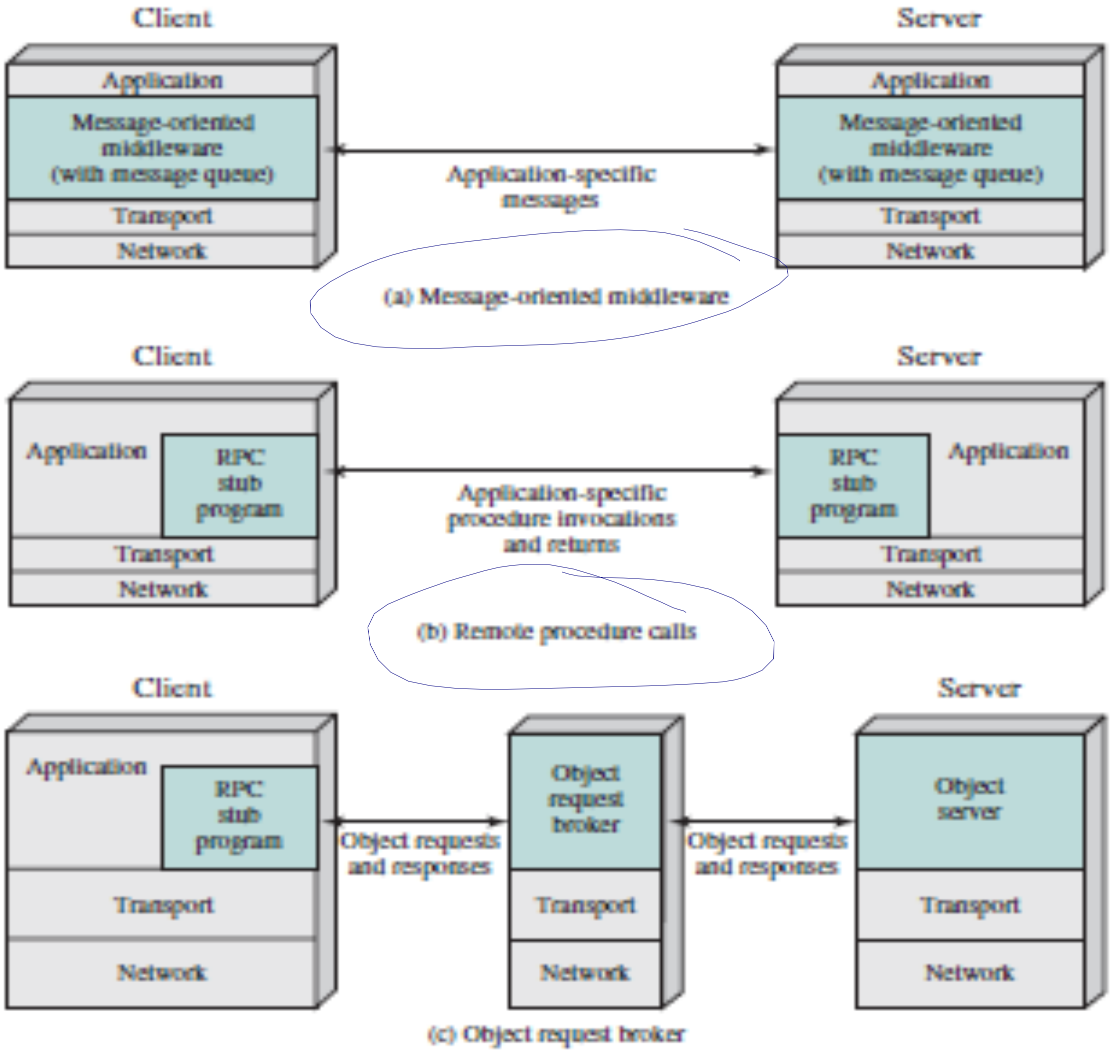
All applications operate over a uniform Application Programming Interface (API). The Middleware, which cuts across all client and server platforms, is responsible for routing client requests to the appropriate servers.



Logical View of Middleware

Distributed System Structures

- Types of Network-Based Operating Systems
- Client / Server Computing: Distributed Message Passing
  - □ Computers in Distributed Processing System are isolated Computer Systems.
  - □ InterProcess Communication technique must rely on Message Passing.



Client
Server
Application
Message-oriented middleware (with message queue)
Transport
Network
Application-specific messages
(a) Message-oriented middleware
Client
Server
Application
RPC stub program
Transport
Network
Application-specific procedure invocations and returns
(b) Remote procedure calls
Client
Server
Application
RPC stub program
Object requests and responses
Object request broker
Object server
Transport
Network
(c) Object request broker

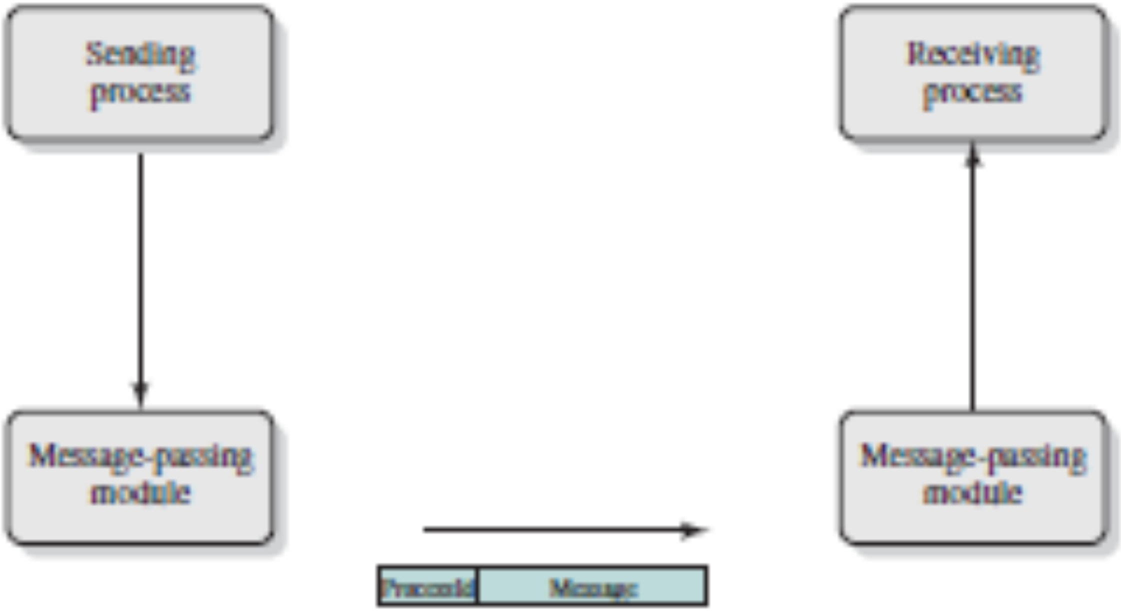
Copyright @ 2009 Pearson Education, Inc.

Distributed System Structures

- Types of Network-Based Operating Systems
- Client / Server Computing: Distributed Message Passing
    - Application-specific Messages:
        - Message contains Primitives and Parameters: Primitives specifies the function to be performed and Parameters are used to pass data and control information.
        - Reliable Message-passing Facility uses reliable transport protocol (TCP) or performs similar logic itself.
        - Unreliable Message-passing Facility simply sends the message into network and does not report success or failure.
        - Blocking Primitive does not return control until message is transmitted (unreliable service) or response received (reliable service).
        - Nonblocking Primitive allows process to continue to run and the process is informed by interrupt or it can poll for status periodically.

Distributed System Structures

- Types of Network-Based Operating Systems
- Client / Server Computing: Distributed Message Passing
    - Application-specific Messages: (Cont)



Basic Message-Passing Primitives

Distributed System Structures

- Types of Network-Based Operating Systems
- Client / Server Computing: Distributed Message Passing
  - Application-specific Procedure Invocations and Returns: Remote Procedure Calls (RPC):
    - □ Allow programs on different systems to interact using simple procedure call/return semantics: the Procedure Call is used for access to remote services.
    - □ Uses reliable, blocking message passing.
    - □ The procedure call is already documented as a operation and parameters (easy error checking).
    - □ Since interface is already defined, the communication code for an application can be generated automatically.

make a func call.
→ translated to RPC.
message
→ send to receiver

Copyright @ 2009 Pearson Education, Inc.

Distributed System Structures

- Types of Network-Based Operating Systems
- Client / Server Computing: Distributed Message Passing
  - Application-specific Procedure Invocations and Returns: Remote Procedure Calls (RPC): (Cont)
    - □ Allow programs on different systems to interact using simple procedure call/return semantics: the Procedure Call is used for access to remote services.
    - □ Uses reliable, blocking message passing.
    - □ The procedure call is already documented as a operation and parameters (easy error checking).

      CALL P (X, Y)

      P = Procedure name, X = Passed arguments, Y = Returned values.
    - □ Since interface is already defined, the communication code for an application can be generated automatically.

Distributed System Structures

- Types of Network-Based Operating Systems
- Client / Server Computing: Distributed Message Passing
  - Application-specific Procedure Invocations and Returns: Remote Procedure Calls (RPC): (Cont)
    - □ Client / Server Binding:

      Nonpersistent Binding: Logical connection is established between two processes during RPC and connection dismantled when completed.

      Persistent Binding: Connection is set up for RPC sustained after completed and can be reused for future RPCs. Connection is terminated after period of no activity.
    - □ Synchronous RPC: Similar to concept of blocking messages. Behavior is predictable, but does not make use of parallelism. Linux uses separate thread for each RPC for parallelism.
    - □ Asynchronous RPC: Similar to concept of nonblocking messages. Client execution proceed in parallel with server invocation.

      Client must check periodically for completed actions.

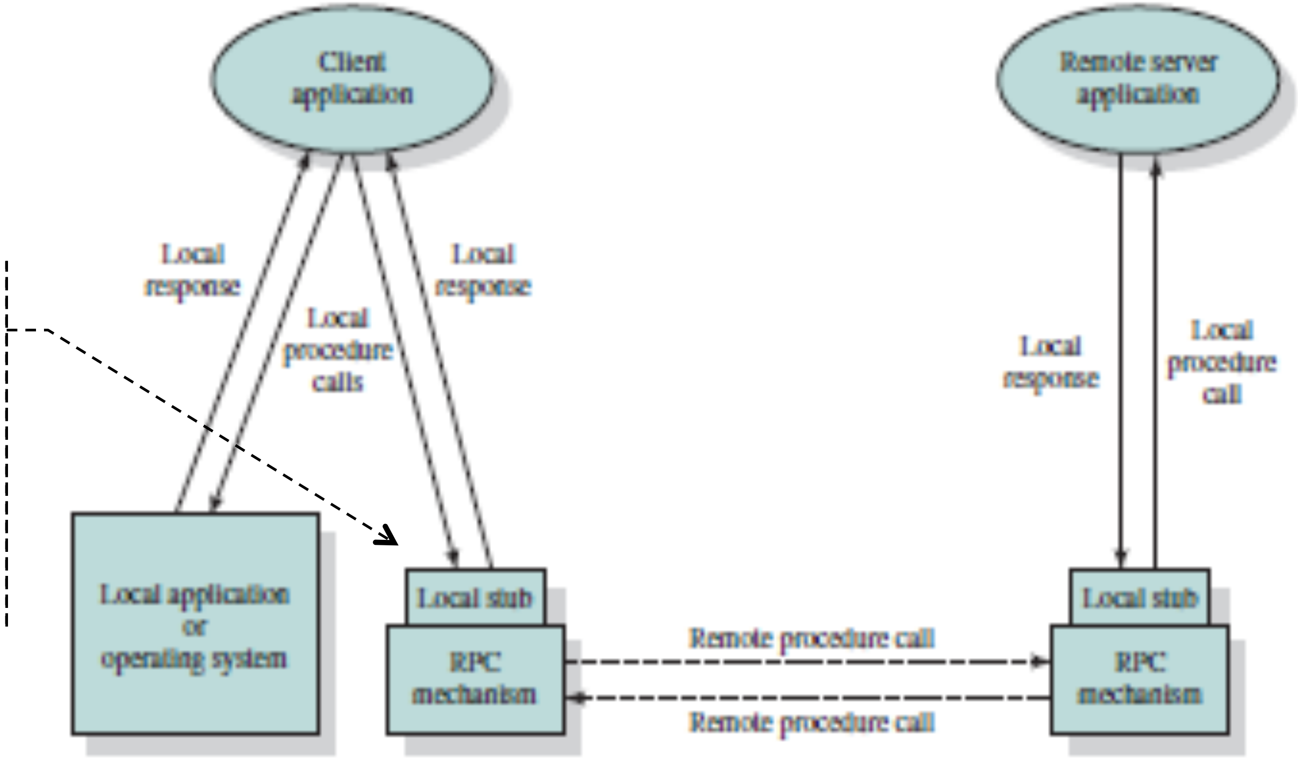
      Client can mix with synchronous RPC to wait for last action.

Distributed System Structures

I&n - RPC related.

- Types of Network-Based Operating Systems
- Client / Server Computing: Distributed Message Passing

Variation of Message-oriented model is the Remote Procedure Call (RPC).
Currently the common method for encapsulating communication in a distributed system.



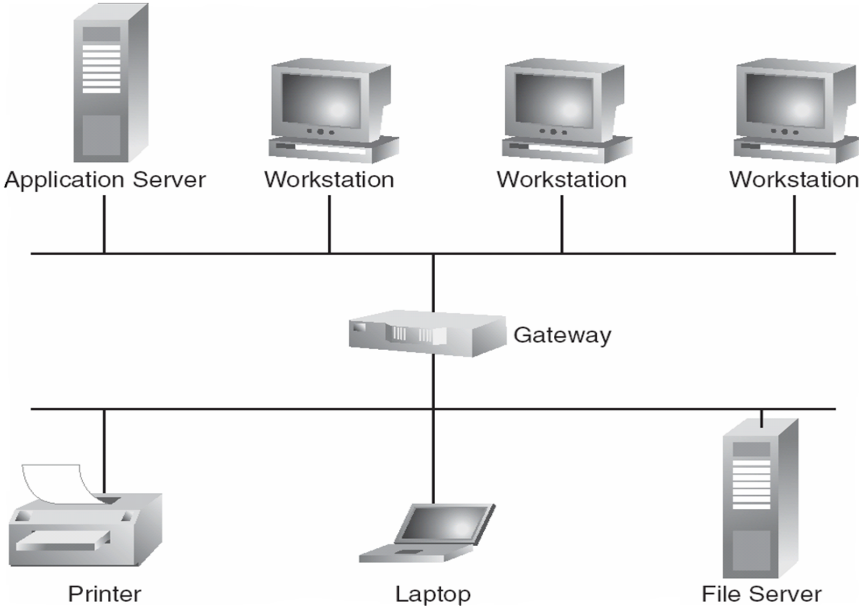
Remote Procedure Call Mechanism

Distributed System Structures

- Network Structure: LAN and WAN
- Local-Area Networks (LAN)
  - □ Local Area Network connecting Computer Systems using twisted-pair or fiber-optic cabling.
    - Ethernet: Multiaccess bus, IEEE 802.3 standard (1/10/100/1000/10000 gbps).
      - □ Simple, Reliable, Cost
    - Token Ring/FDDI: Token-based (4 or 16/100 mbps).
      - □ Deterministic, Distance, Throughput under heavy load.

post & pass the token for next device to post.
all devices have chance to talk

Cons: require special hardware.
- risk of losing token.



Distributed System Structures

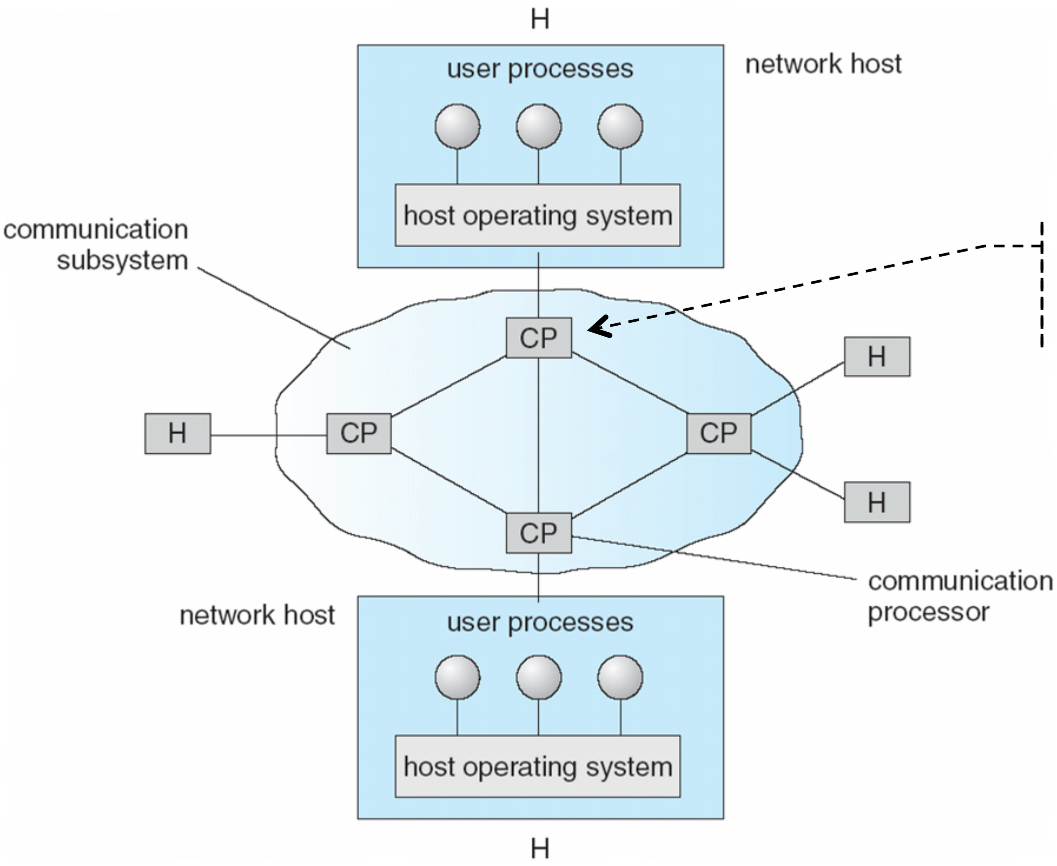
- Network Structure: LAN and WAN
- Wide-Area Networks (WAN)
  - Arpanet, 1968, packet switching network funded by Dod as academic research project for use by projects in university and research laboratories.
    - The intent was to sharing hardware and software among wide community of users.
    - Connecting universities allowed Linux to develop among university students and engineers.
  - National Science Foundation established networking sites and used autonomous processors, routers, to peer with Arpanet and form the worldwide network.
    - Connects between Regional Networks use telephone-system services.
    - T1 lines (1.5), T3 lines (45 Mbps), OC3 (155Mbps) . . . 10 Gbps and higher.
  - Point-to-point (PPP Protocol) connections over modem connections allow home computers (leased from a phone company) to connect to Internet.

Distributed System Structures

TQn false statement about TokenRing

Ans: all statement are true (but it may be different in exam)

- Network Structure: LAN and WAN
- Wide-Area Networks



Communications Processors or Routers in Wide-Area Network.

Distributed System Structures

- Network Topology
- Sites in the system can be physically connected in a variety of ways; they are compared with respect to the following criteria:
  - **Installation cost** - How expensive is it to link the various sites in the system?
  - **Communication cost** - How long does it take to send a message from site *A* to site *B*?
  - **Reliability** - If a link or a site in the system fails, can the remaining sites still communicate with each other?
- The various topologies are depicted as graphs whose nodes correspond to sites
  - An edge from node *A* to node *B* corresponds to a direct connection between the two sites.

Distributed System Structures

- Network Topology

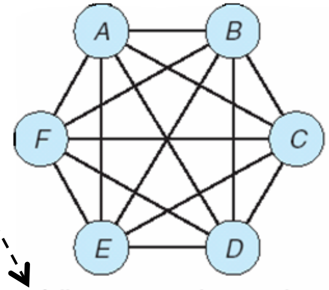
Fully Connected Network: Each site directly connected to every other site.
Installation cost high, since number of links grows as square of number of sites.

Partially Connected Network: Direct link to some, but not all pairs of sites. Installation cost lower. Communication cost high, since sites must route through other sites.
System can be partitioned, when a site fails.

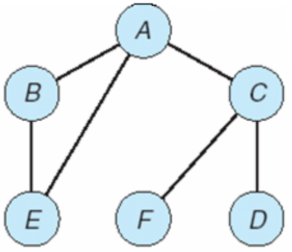
Tree-Structured: Installation cost low. Communication cost low. System can be partitioned, when a site fails.

Star Network: Installation cost low. Communication cost low. System can be partitioned, but only one site. Failure of central site disconnects ALL sites.

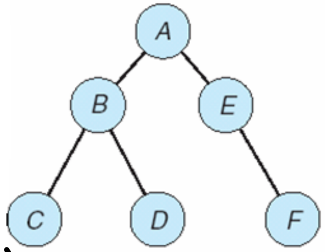
Ring Network: Installation cost low. Communication cost high. System can be partitioned, when a two site fails.



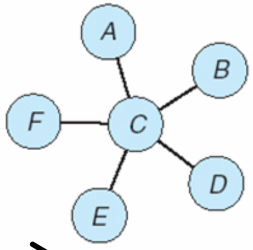
fully connected network



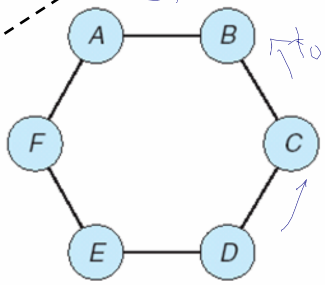
partially connected network



tree-structured network



star network



ring network

token

more reliable

Distributed System Structures

- Communication Structure
- Internal workings of Communication Network.
- The design of a Communication Network must address five basic issues:
  - □ **Naming and name resolution** - How do two processes locate each other to communicate?
  - □ **Routing strategies** - How are messages sent through the network?
  - □ **Connection strategies** - How do two processes send a sequence of messages?
  - □ **Contention -** The network is a shared resource, so how do we resolve conflicting demands for its use?

Distributed System Structures

- Communication Structure
- Naming and Name Resolution
  - Within Computer System, each host has a Process Identifier: In the Internet, the Identifier is the IP Address.
  - Identify Processes by: <Host-Name, Identifier> pair

    Host-Name is alphnumeric name.

    Identifier is unique number (IP Address) .
  - Data File containing Host-Name and IP Address of all Hosts.
    - Became unmanageable as Internet grew.
    - Linux: /etc/hosts

```
127.0.0.1                 localhost
192.168.1.100             student1.svuca.edu
192.168.1.101             student2.svuca.edu
```

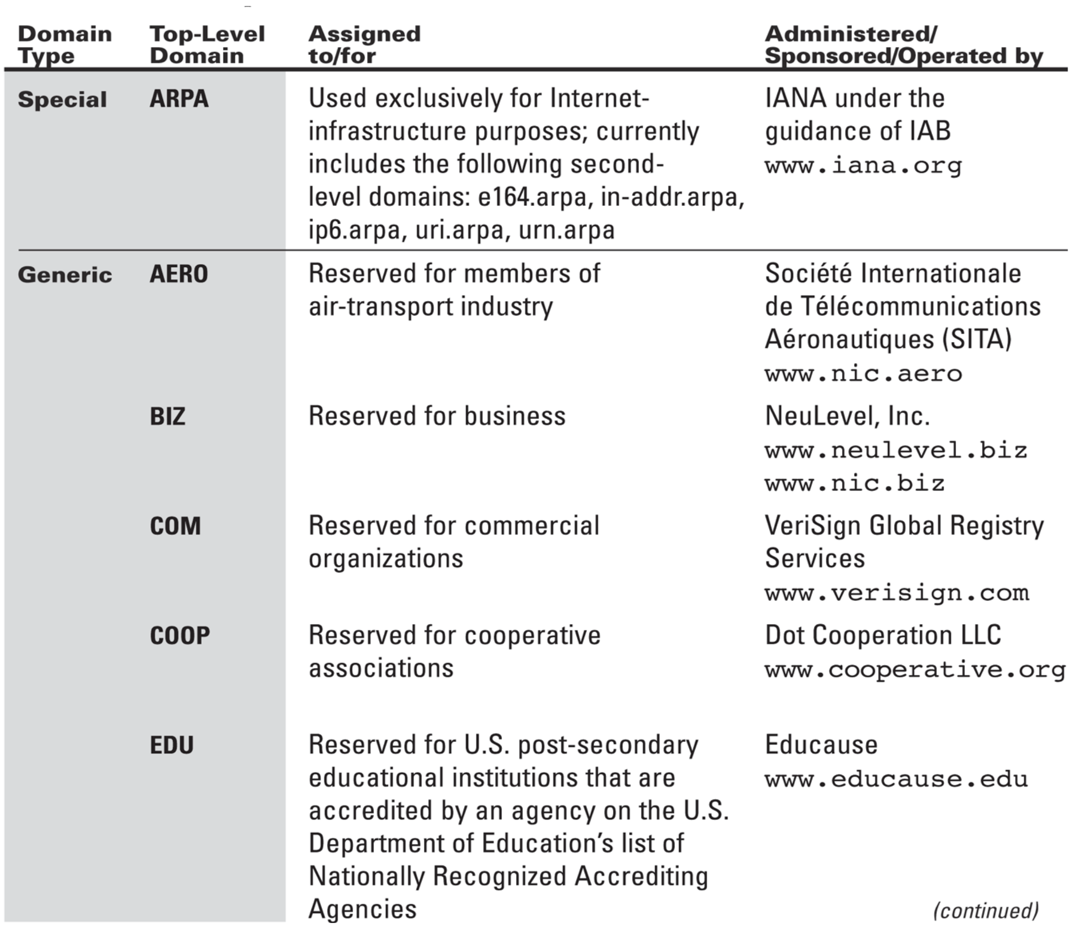
  - **Domain name service (DNS):**
    - Specifies the naming structure of the hosts, as well as name to address resolution (Internet).

Distributed System Structures

- Communication Structure
- Naming and Name Resolution
- Host-Name
  - Management of IP addresses easier.
  - Unique for a host on the Internet.
  - Remains the same, even if the numeric IP address changes.
- Format: hostname.domain_name
  - FQDN (Fully Qualified Domain Name) is host name attached to the domain name with a period between them.
  - Domain Name assigned by Internet Network Information Center (NIC).
  - Top-Level Domain is right-most string.
  - Organization is string to the left of the right-most period.
  - Another string assigned to the left of the Organization if "Organization.Top-Level-Domain" is used.
  - Three levels of Top-Level Domains: special, generic, and country-code.

Distributed System Structures

- Communication Structure
- Naming and Name Resolution

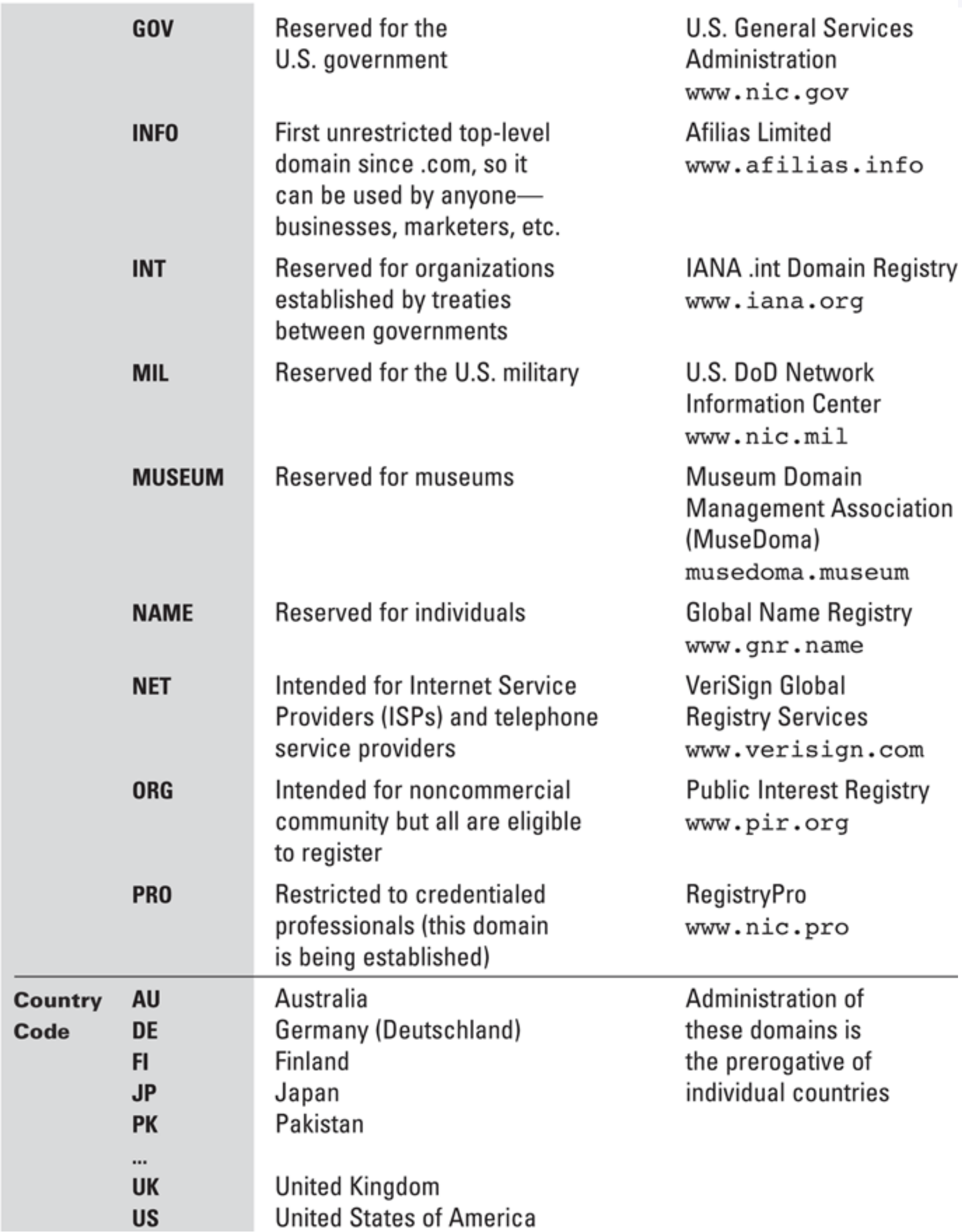


Domain Type	Top-Level Domain	Assigned to/for	Administered/ Sponsored/Operated by
Special	ARPA	Used exclusively for Internet-infrastructure purposes; currently includes the following second-level domains: e164.arpa, in-addr.arpa, ip6.arpa, uri.arpa, urn.arpa	IANA under the guidance of IAB www.iana.org
Generic	AERO	Reserved for members of air-transport industry	Société Internationale de Télécommunications Aéronautiques (SITA) www.nic.aero
	BIZ	Reserved for business	NeuLevel, Inc. www.neulevel.biz www.nic.biz
	COM	Reserved for commercial organizations	VeriSign Global Registry Services www.verisign.com
	COOP	Reserved for cooperative associations	Dot Cooperation LLC www.cooperative.org
	EDU	Reserved for U.S. post-secondary educational institutions that are accredited by an agency on the U.S. Department of Education's list of Nationally Recognized Accrediting Agencies	Educause www.educause.edu

(continued)

Distributed System Structures

- Communication Structure
- Naming and Name Resolution



	GOV	Reserved for the U.S. government	U.S. General Services Administration `www.nic.gov`
	INFO	First unrestricted top-level domain since .com, so it can be used by anyone—businesses, marketers, etc.	Afilias Limited `www.afilias.info`
	INT	Reserved for organizations established by treaties between governments	IANA .int Domain Registry `www.iana.org`
	MIL	Reserved for the U.S. military	U.S. DoD Network Information Center `www.nic.mil`
	MUSEUM	Reserved for museums	Museum Domain Management Association (MuseDoma) `musedoma.museum`
	NAME	Reserved for individuals	Global Name Registry `www.gnr.name`
	NET	Intended for Internet Service Providers (ISPs) and telephone service providers	VeriSign Global Registry Services `www.verisign.com`
	ORG	Intended for noncommercial community but all are eligible to register	Public Interest Registry `www.pir.org`
	PRO	Restricted to credentialed professionals (this domain is being established)	RegistryPro `www.nic.pro`
Country Code	**AU**	Australia	Administration of these domains is the prerogative of individual countries
	DE	Germany (Deutschland)	
	FI	Finland	
	JP	Japan	
	PK	Pakistan	
	...		
	UK	United Kingdom	
	US	United States of America	

Distributed System Structures

TQn: FQDN addess?

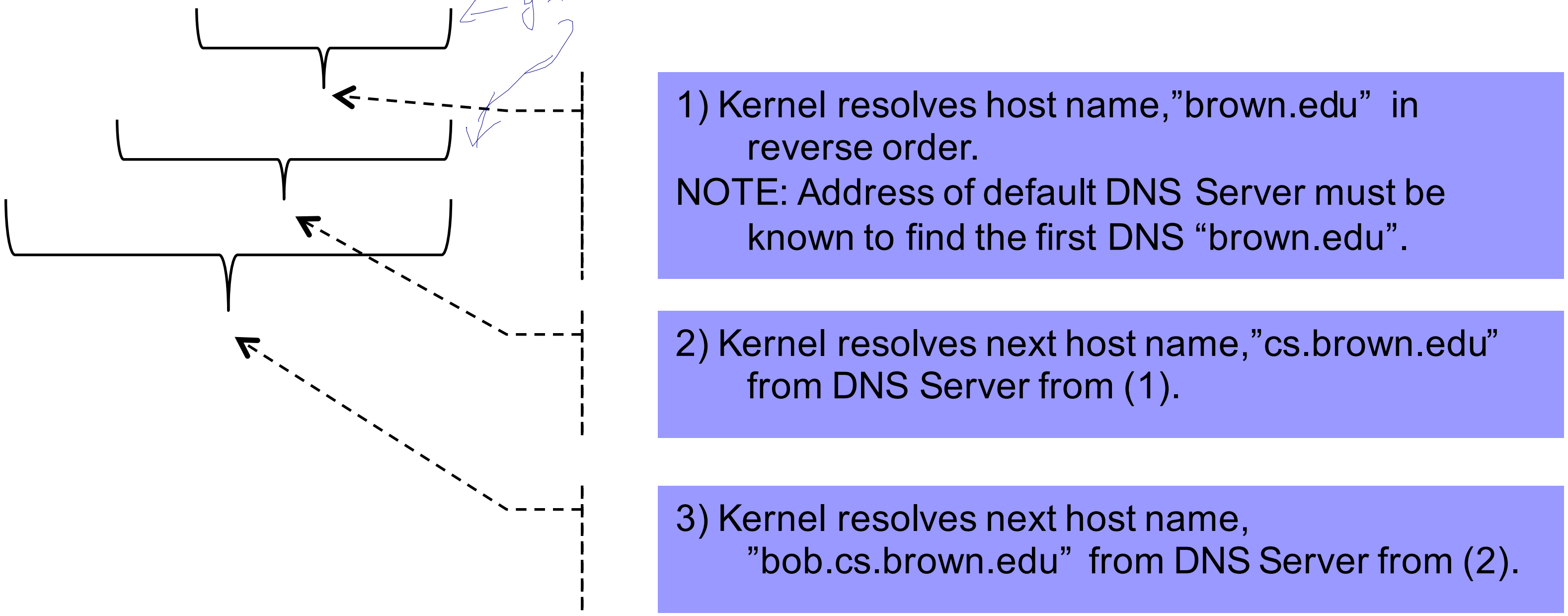
ans: sau@svuca.edu (surprised!)

not svuca.edu

- Communication Structure
- Naming and Name Resolution
  - □ Resolve from most specific to the most general part.

bob.cs.brown.edu

got to this domain DNS server



1) Kernel resolves host name,”brown.edu” in reverse order.
NOTE: Address of default DNS Server must be known to find the first DNS “brown.edu”.

2) Kernel resolves next host name,”cs.brown.edu” from DNS Server from (1).

3) Kernel resolves next host name, ”bob.cs.brown.edu” from DNS Server from (2).

Distributed System Structures

- Communication Structure
- Routing Strategies
  - What in Operating System determines how are messages sent from Site A to Site B?
  - Routing Table indicates the paths that can be used to send a message to other sites.
  - Routing schemes: Fixed Routing, Virtual Routing, and Dynamic Routing.
- **Fixed routing:** A path from *A* to *B* is specified in advance; path changes only if a hardware failure disables it.
  - The shortest path chosen, communication costs are minimized.
  - Fixed routing cannot adapt to load changes.
  - Ensures that messages delivered in the order in which they were sent.

Distributed System Structures

- Communication Structure
- Routing Strategies

packet in sequence. (different from dynamic routing?)

- **Virtual circuit:** A path from *A* to *B* is fixed for the duration of one session. Different sessions involving messages from *A* to *B* may have different paths.
  - □ Partial remedy to adapting to load changes.
  - □ Session can be a file transfer or a remote-login session to another site.
  - □ At start of session, adapts to load changes by avoiding routing messages on heavily used path.
  - □ Ensures that messages will be delivered in the order in which they were sent.

Distributed System Structures

Qn: which routing is more adaptable to load changes?

Ans: Dynamic routing.

- Communication Structure
- Routing Strategies
- **Dynamic routing:** The path used to send a message from site *A* to site *B* is chosen when a message is sent.
  - □ Usually a site sends a message to another site on the link least used at that particular time.
  - □ Adapts to load changes by avoiding routing messages on heavily used path.
  - □ Router is host with routing software or special device.
    - Routing Protocol between Routers to inform each other of network changes and update routing tables automatically.
  - □ Messages may arrive out of order:
    - Routing decision dynamic, separate messages may be assigned different paths.
    - Remedied by appending a sequence number to each message.

different than fixed or virtual circuit routing. →

Distributed System Structures

- Communication Structure
- Packet Strategies
  - □ Messages vary in length, but to simply Communication Structure, communication is implemented with fixed-length Messages: Packets, Frames, or Datagrams.
  - □ Unreliable Message:
    - Connectionless Oriented (UDP), sender has no guarantee that the packet reached its destination.
  - □ Reliable Message:
    - Connection Oriented (TCP), acknowledgement packet from destination indicating that the packet has reached its destination.

Distributed System Structures

Tqn: which switching: msg with var length. can take different path in the network?

Ans: packet switching

- Communication Structure
- Connection Strategies:
  - □ Three common schemes to connect sites.
- **Circuit switching** - A permanent physical link is established for the duration of the communication (i.e., telephone system).
- **Message switching** - A temporary link is established for the duration of one message transfer (i.e., post-office mailing system).
- **Packet switching** - Messages of variable length are divided into fixed-length packets which are sent to the destination.
  - □ Each packet may take a different path through the network.
  - □ The packets must be reassembled into messages as they arrive.
- Circuit switching requires setup time, but incurs less overhead for shipping each message, and may waste network bandwidth.
  - □ Message and packet switching require less setup time, but incur more overhead per message.

Distributed System Structures

- Communication Structure
- Contention
  - Ring or Multi-Access Bus Network must resolve simultaneous transmission request by multiple sites.
- **CSMA/CD**: Carrier Sense with Multiple Access (CSMA); Collision Detection (CD)
  - A site determines whether another message is currently being transmitted over that link. If two or more sites begin transmitting at exactly the same time, then they will register a CD and will stop transmitting.
  - When the system is very busy, many collisions may occur, and thus performance may be degraded
- CSMA/CD is used successfully in the Ethernet system, the most common network system.

Distributed System Structures

- Communication Structure
- Contention
- **Token Passing:** A unique message type, known as a token, continuously circulates in the system (usually a ring structure).
  - □ A site to transmit information must wait until the token arrives.
  - □ Site converts token into data frame for transmission. Once site receives its own data frame, it is converted to token and retransmitted.
  - □ Scheme is used by some IBM and HP/Apollo systems
- **Message Slots:** Time-Division Multiplexing and fixed-length message slots continuously circulate in the system (a ring structure).
  - □ Since a slot can contain only fixed-sized messages, a single logical message may have to be broken down into a number of smaller packets, each of which is sent in a separate slot.
  - □ This scheme has been adopted in the experimental Cambridge Digital Communication Ring.

Distributed System Structures

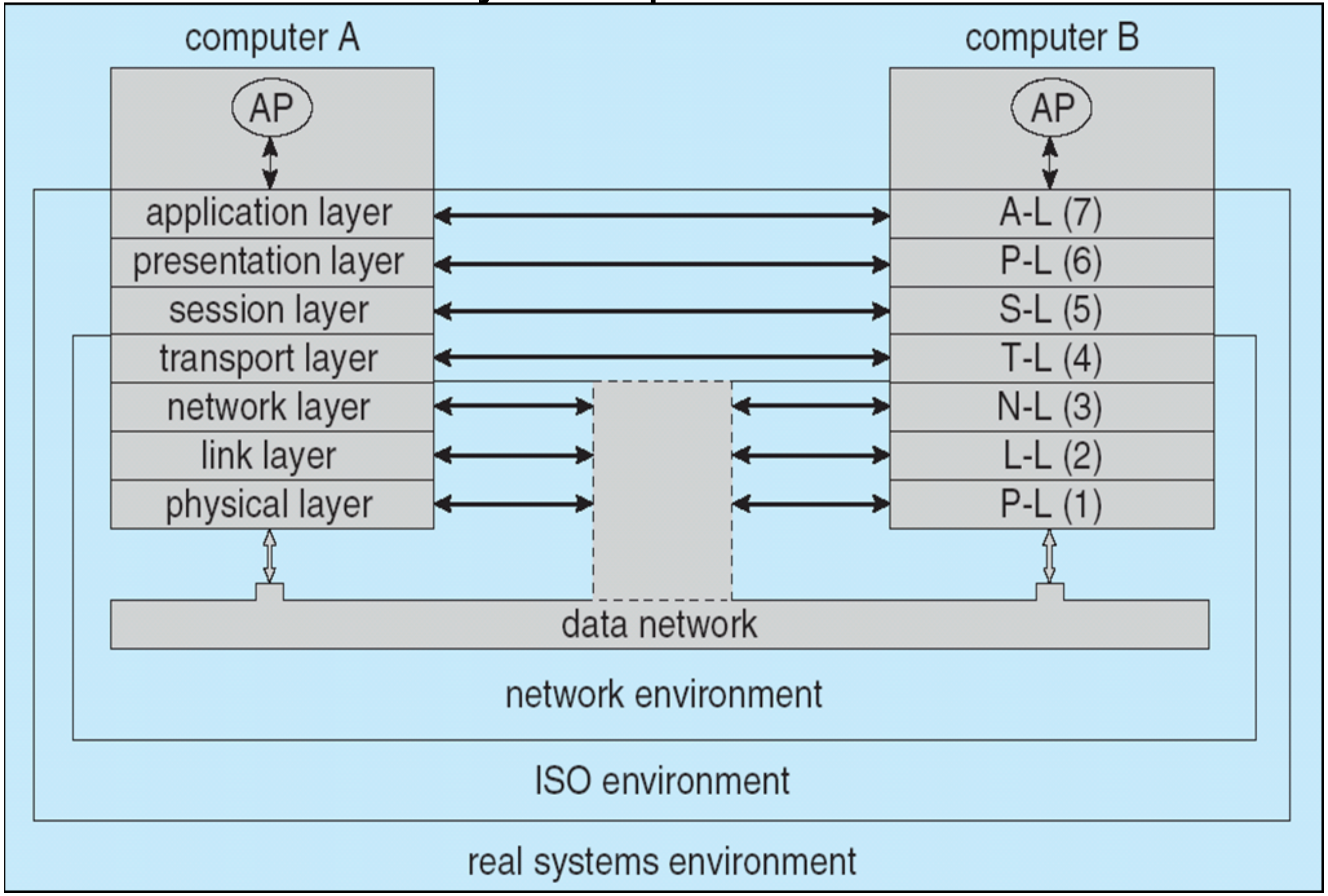
- Communication Protocols
- **ISO (International Standards Organization)** standardizes the internal function of a communication system by partitioning into abstraction layers: Open Systems Interconnection (OSI) Project.
  - □ The communication network is partitioned into multiple layers.
  - □ Each layer, one system communicates with same layer other system.
- **Physical layer:** Handles the mechanical and electrical details of the physical transmission of a bit stream.
- **Data-link layer:** Handles the *frames*, or fixed-length parts of packets, including any error detection and recovery that occurred in the physical layer.
- **Network layer:** Provides connections and routes packets in the communication network, including handling the address of outgoing packets, decoding the address of incoming packets, and maintaining routing information for proper response to changing load levels.

Distributed System Structures

- Communication Protocols
- **Transport layer:** Responsible for low-level network access and for message transfer between clients, including partitioning messages into packets, maintaining packet order, controlling flow, and generating physical addresses.
- **Session layer:** Implements sessions, or process-to-process communications protocols.
- **Presentation layer:** Resolves the differences in formats among the various sites in the network, including character conversions, and half duplex/full duplex (echoing).
- **Application layer:** Interacts directly with the users' deals with file transfer, remote-login protocols and electronic mail, as well as schemas for distributed databases.

Distributed System Structures

- Communication Protocols
  - Logical communications between two Computer Systems, with the three lowest-level layers implemented in hardware.



Distributed System Structures

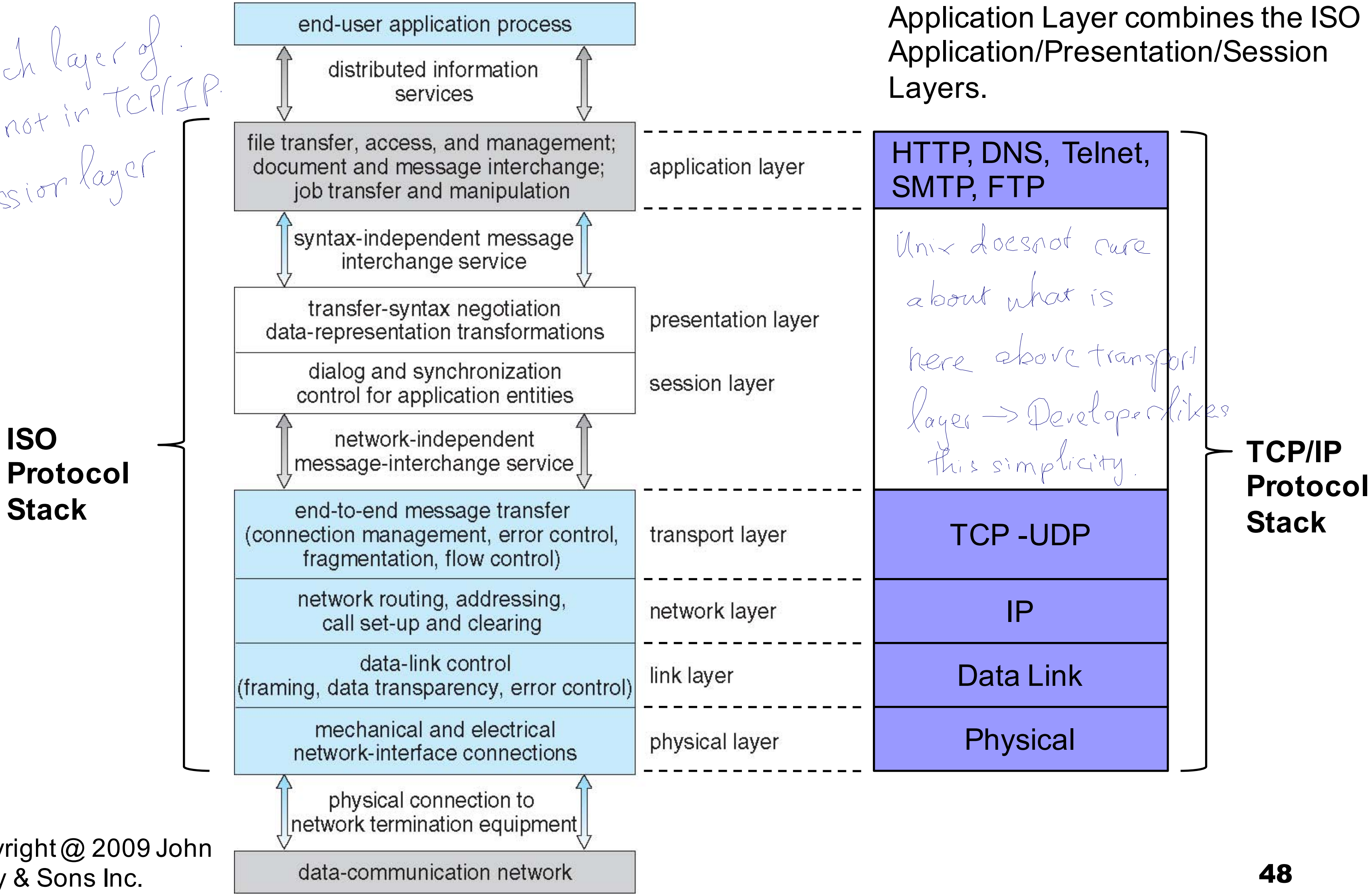
- Communication Protocols

TQn: which layer of ISO is not in TCP/IP.

Ans: Session layer

TCP/IP has fewer layers than does the ISO Model.

Application Layer combines the ISO Application/Presentation/Session Layers.



end-user application process
distributed information services
file transfer, access, and management; document and message interchange; job transfer and manipulation
syntax-independent message interchange service
transfer-syntax negotiation data-representation transformations
dialog and synchronization control for application entities
network-independent message-interchange service
end-to-end message transfer (connection management, error control, fragmentation, flow control)
network routing, addressing, call set-up and clearing
data-link control (framing, data transparency, error control)
mechanical and electrical network-interface connections
physical connection to network termination equipment
data-communication network
application layer
presentation layer
session layer
transport layer
network layer
link layer
physical layer
HTTP, DNS, Telnet, SMTP, FTP
TCP -UDP
IP
Data Link
Physical
ISO Protocol Stack
TCP/IP Protocol Stack

Unix doesnot care about what is here above transport layer → Developer likes this simplicity.

Distributed System Structures

- Robustness
  - Detection of Link Failure, Site Failure, or Lost Messages.
    - Handshake Procedures (“Are-You-Up?”) or Ping.
    - Time-out, still unable to determine exactly which scenerio.
  - Scenerio:
    - Assume Site A and Site B have established a link
      - At fixed intervals, each site will exchange an *I-am-up* message indicating that they are up and running
    - If Site A does not receive a message within the fixed interval, it assumes either (a) the other site is not up or (b) the message was lost
    - Site A can now send an *Are-you-up?* message to Site B
    - If Site A does not receive a reply, it can repeat the message or try an alternate route to Site B

Distributed System Structures

- Robustness
  - Scenerio:
    - If Site A does not ultimately receive a reply from Site B, it concludes some type of failure has occurred
    - Types of failures:
      - Site B is down
      - The direct link between A and B is down
      - The alternate link from A to B is down
      - The message has been lost
    - However, Site A cannot determine exactly **why** the failure has occurred

Distributed System Structures

- Reconfiguration
  - Initiate procedure to reconfigure and continue normal mode of operation.
  - Scenerio:
    - When Site A determines a failure has occurred, it must reconfigure the system:

      1. If the link from A to B has failed, this must be broadcast to every site in the system.

      2. If a site has failed, every other site must also be notified indicating that the services offered by the failed site are no longer available.

    - When the link or the site becomes available again, this information must again be broadcast to all other sites.

Distributed System Structures

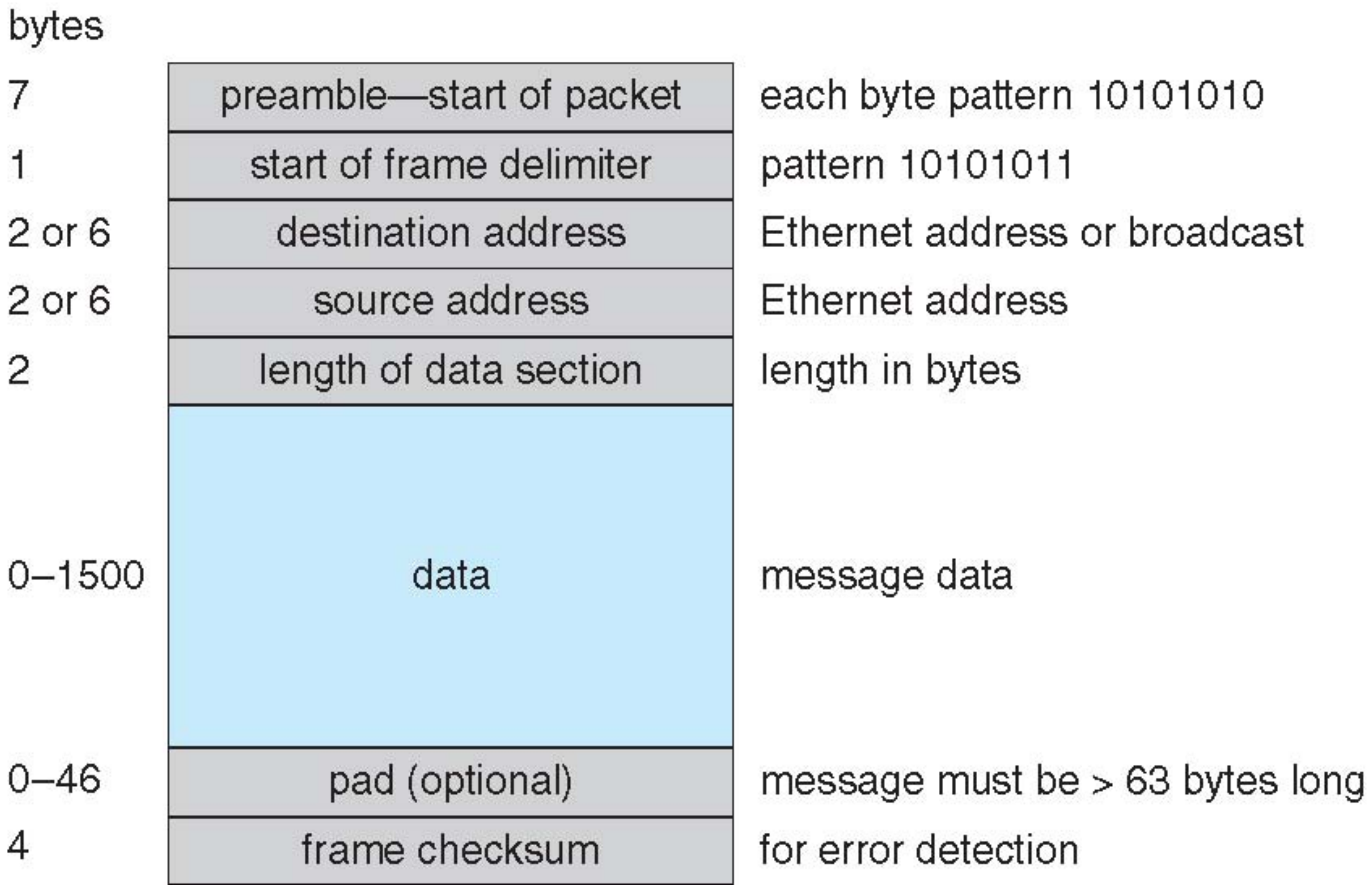
- Design Issues
- Transparency: The distributed system should appear as a conventional, centralized system to the user
- Fault tolerance: The distributed system should continue to function in the face of failure main reason we want distributed systems, a big +
- Scalability: As demands increase, the system should easily accept the addition of new resources to accommodate the increased demand
  - Central control schemes and central resources should not be used for scalable or fault-tolerant systems.
  - Ideal configuration is a functionally symmetric configuration; all the component machines have an equal role in the operation of the system, and each machine has some degree of autonomy.
- Clusters: A collection of semi-autonomous machines that acts as a single system

Distributed System Structures

- Example: Networking
- The transmission of a network packet between hosts on an Ethernet network.
  - Every host has a unique IP address and a corresponding Ethernet (MAC) address.
  - Communication requires both addresses.
  - Domain Name Service (DNS) can be used to acquire IP addresses.
  - Address Resolution Protocol (ARP) is used to map MAC addresses to IP addresses.
  - If the hosts are on the same network, ARP can be used
    - If the hosts are on different networks, the sending host will send the packet to a *router* which routes the packet to the destination network.

Distributed System Structures

- Example: Networking
- Ethernet Packet



bytes		
7	preamble—start of packet	each byte pattern 10101010
1	start of frame delimiter	pattern 10101011
2 or 6	destination address	Ethernet address or broadcast
2 or 6	source address	Ethernet address
2	length of data section	length in bytes
0–1500	data	message data
0–46	pad (optional)	message must be > 63 bytes long
4	frame checksum	for error detection

Distributed System Structures

Iqn. which behavior NOT belong to Network OS?

Ans: Network OS : need to Know <u>which res</u> to look for!

- Summary
- A distributed system is a collection of processors that do not share main memory or Real-Time clock.
- Processor connected through a communication network configured as a tree, a star, a ring, or a multi-access bus.
- A distributed system provides the user with access to the resources the system provides:
  - Data Migration, Computation Migration, or Process Migration.
- Network Layer Model specifies Protocol Stacks
  - Naming System, DNS, must be used to translate from Host name to a network address.
  - Address Resolution Protocol, ARP, translates Network Number to a Network Device .
- A distributed system need to be fault tolerate.