Communication: Layered Protocols, Lower-Level Protocols, Transport Protocols, Higher-Level Protocols, Remote Procedure Call: Basic RPC Operation, Parameter Passing. Extended RPC Models, Remote Object

Invocation: Distributed Objects, Binding a Client to an Object; Static verses Dynamic Remote Method Invocations, Parameter Passing, Message Oriented. Communication: Persistence and synchronicity in Communication, Message Oriented Transient Communication, and Message-Oriented’ Persistent Communication, Stream Oriented Communication: Support for Continuous Media, Streams and Quality of Service, Stream Synchronization.

**PROTOCOLS**

Protocol= Set of Rules for how computers communicate with each other.

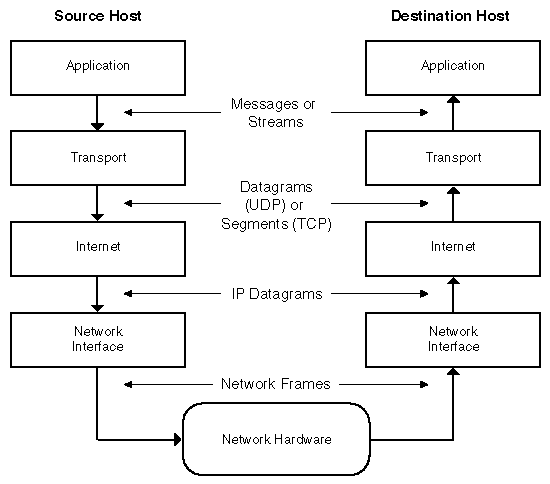
|  |  |
| --- | --- |
| **PROTOCOLS TYPES** | |
| **1. Lower level Protocols (Device to Device).**  The lowest protocol always deals with "low-level", physical interaction of the hardware. Every higher layer adds more features.  **a. IP (Internet Protocol)** (The Address of the Machine)).  **b. TCP (Transmission Control Protocol) (**Proof of Delivery, rules or reassembling partitioned messages)).  c. Implementation in physical layer and data link layer of the stack. Group data bits into frames and adds a pattern called checksums at either end of frame. | **2. Higher Level Protocols (Program to Program)**  a. FTP: File Transfer protocol.  b. SMTP: simple mail transfer protocol.  c. HTTP: Hyper Text Transfer Protocol.  d. Network Layer chooses best path from sender to receiver by routing. |

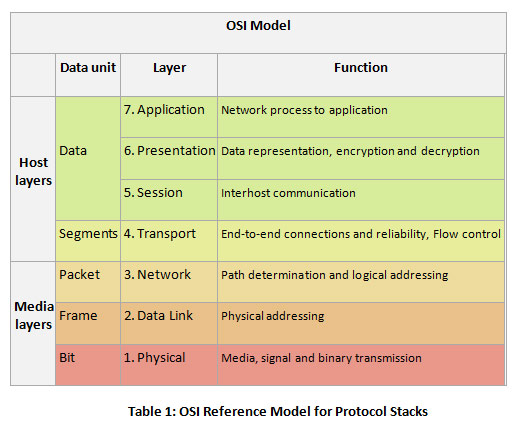
**Layers in the OSI model, they can be grouped into three areas:**

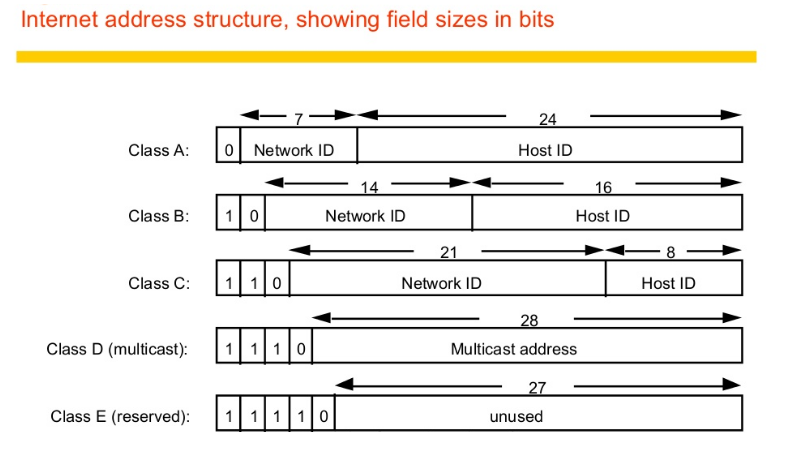
* **High-level Protocols (layers 5, 6 and 7  -  Session, Presentation, and Application)**- how the data is presented, displayed, and summarized for the user  -  and in the reverse direction, how the user prepared data is assembled into meaningful data structures (high-level protocols).
* **Medium-level Protocols (Layers 3 and 4 - Network and Transport)**- how the data is assembled into packets and frames and how error checking and flow control is implemented - and in the reverse direction, how the received packets and frames are assembled into structures such as files and databases (medium-level protocols)
* **Low-level Protocols (Layers 1 and 2 - Physical and Data Link)** - how the data is converted into electrical pulses of one's and zero's (bits) and sent across cables or the physical medium, and in the reverse direction, how the electrical pulses are taken off the cable and converted to ones and zeros.

<http://www.infocellar.com/networks/osi-model.htm>

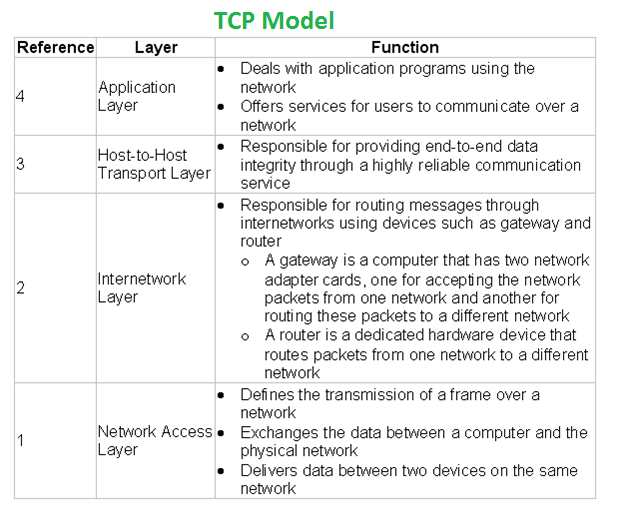
<https://en.wikipedia.org/wiki/List_of_network_protocols_(OSI_model)>

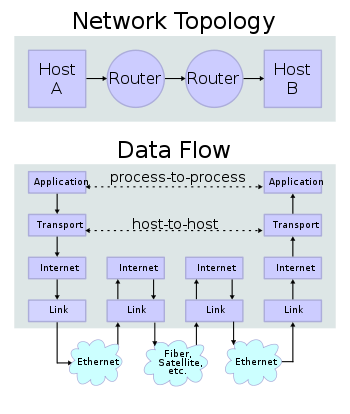


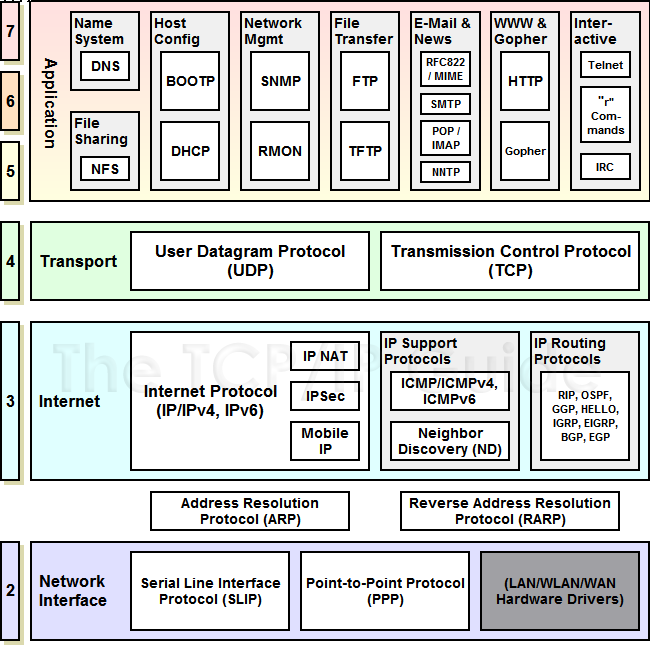


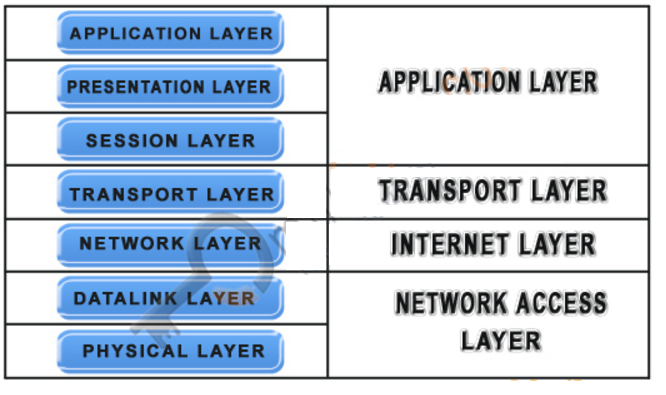


|  |  |
| --- | --- |
| **OSI** | **TCP** |
| **Application Layer** | **Application Layer** |
| **Presentation Layer** |
| **Session Layer** |
| **Transport Layer** | **Transport Layer** |
| **Network Layer** | **Internet Layer** |
| **Data link Layer** | **Network Access Layer** |
| **Physical Layer** |



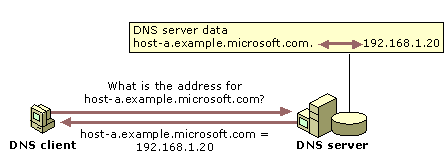




****

**Domain Name System (DNS)**

It is used for naming computers and network services that is organized into a hierarchy of **domains**. **DNS** naming is used in TCP/IP networks, such as the Internet, to locate computers and services through user-friendly **names**.



**Network File System** (**NFS**)

It **allows a user on a client computer to access files over a computer network** much like local storage is accessed.

**Bootstrap Protocol (BOOTP)**

It **automatically assigns an IP address to network devices from a configuration server.** The **BOOTP** was originally defined in RFC 951.

**Dynamic Host Configuration Protocol (DHCP)**

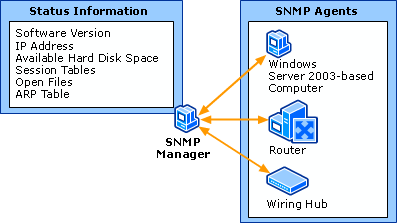
**It** **automatically assign an IP address to a computer** from a defined range of numbers (i.e., a scope) configured for a given network.

**Differences between BOOTP and DHCP**

* BOOTP supports a **limited number of client configuration parameters called vendor extensions,** while DHCP supports a larger and extensible set of client configuration parameters called options.
* BOOTP uses a two-phase bootstrap configuration process in which clients contact BOOTP servers to perform address determination and boot file name selection, and clients contact Trivial File Transfer Protocol (TFTP) servers to perform file transfer of their boot image. DHCP uses a single-phase boot configuration process whereby a DHCP client negotiates with a DHCP server to determine its IP address and obtain any other initial configuration details it needs for network operation.
* BOOTP clients do not rebind or renew configuration with the BOOTP server except when the system restarts, while DHCP clients do not require a system restart to rebind or renew configuration with the DHCP server. Instead, clients automatically enter the Rebinding state at set timed intervals to renew their leased address allocation with the DHCP server. This process occurs in the background and is transparent to the user.

**Simple Network Management Protocol (SNMP)**

It is used for **collecting** information from, and **configuring**, network devices, such as servers, printers, hubs, switches, and routers on an Internet Protocol (IP) network.



**Remote Monitoring (RMON)**

It is a standard specification that **facilitates the monitoring of network operational activities** through the **use of remote devices known as monitors or probes.** **RMON** assists network administrators (NA) with efficient network infrastructure control and management.

**File Transfer Protocol (FTP)**

It is used to transfer computer files between a client and server on a computer network.

**Trivial File Transfer Protocol (TFTP)**

It is used for transferring files that is simpler to use than the File Transfer Protocol (FTP) but less capable. It is used where user authentication and directory visibility are not required.

**Request for Comments (RFC)**

It is a type of publication from the Internet Engineering Task Force (IETF) and the Internet Society (ISOC), the principal technical development and standards-setting bodies for the Internet.

**Simple Mail Transfer Protocol (SMTP)**

It is used for electronic mail (email) transmission.

**Post Office Protocol** (**POP**)

It is used by local e-mail clients to retrieve e-mail from a remote server over a TCP/IP connection.

**IMAP (Internet Message Access Protocol)**

It is a standard email protocol that stores email messages on a mail server, but allows the end user to view and manipulate the messages as though they were stored locally on the end user's computing device(s).

**Network News Transfer Protocol (NNTP)**

It is an application protocol used for transporting Usenet news articles (netnews) between news servers and for reading and posting articles by end user client applications.

**Hypertext Transfer Protocol (HTTP)**

It is an application protocol for distributed, collaborative, hypermedia information systems. **HTTP** is the foundation of data communication for the World Wide Web. Hypertext is structured text that uses logical links (hyperlinks) between nodes containing text.



**Gopher protocol (TCP/IP application layer)**

It is designed for distributing, searching, and retrieving documents over the Internet.

**Telnet (TCP/IP protocol)**

It is a user command for accessing remote computers. Through **Telnet**, an administrator or another user can access someone else's computer remotely.

**Internet Relay Chat Protocol (IRCP)**

It is an application layer protocol that facilitates communication in the form of text. The chat process works on a client/server networking model. IRC clients are computer programs that a user can install on their system.

**User Datagram Protocol** (**UDP**)

It is simplest Transport Layer communication protocol available of the TCP/IP **protocol** suite. It involves minimum amount of communication mechanism. **UDP** is said to be an unreliable transport **protocol** but it uses IP services which provides best effort delivery mechanism.

**Transmission Control Protocol** (**TCP**)

It is a core **protocol** of the Internet protocol suite. It originated in the initial network implementation in which it complemented the Internet Protocol (IP). Therefore, the entire suite is commonly referred to as **TCP**/IP.

**IP Nat**

It enables private **IP** networks that use unregistered **IP** addresses to connect to the Internet. **NAT** operates on a router, usually connecting two networks together, and translates the private (not globally unique) addresses in the internal network into legal addresses, before packets are forwarded to another network.

**Private Network--🡪Use unregistered IP Address---🡪Internet.**

**Routing Information Protocol** (**RIP**)

It is one of the oldest distance-vector routing **protocols** which employ the hop count as a routing metric. **RIP** prevents routing loops by implementing limit on the number of hops allowed in a path from source to destination.

Each rip router maintains a routing table which is a list of all the destinations networks it knows how to reach along with the distance to that destination.

**Open Shortest Path First** (**OSPF**)

It uses a link state routing (LSR) algorithm and falls into the group of interior routing **protocols**, operating within a single autonomous system (AS). It is defined as **OSPF** Version 2 in RFC 2328 (1998) for IPv4.

**Address resolution protocol** (**ARP(IP----Hardware Address)**)

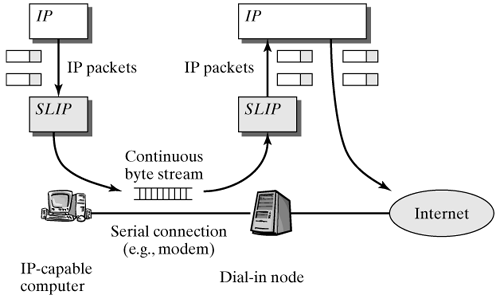
It is a **protocol** used to map IP network addresses to the hardware addresses used by a data link **protocol**. It below the network layer as a part of the interface between the OSI network and OSI link layer.

**Reverse Address Resolution Protocol** (**RARP (Hardware Address ----IP)**)

It is an obsolete computer networking **protocol** used by a client computer to request its Internet **Protocol**(IPv4) address from a computer network, when all it has available is its link layer or hardware address, such as a MAC address.

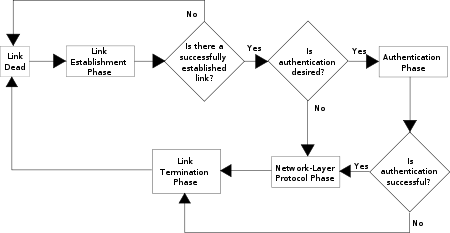
**Serial Line Internet Protocol** (**SLIP**)

It is an encapsulation of the Internet Protocol designed to work over serial ports and modem connections. It is documented in RFC 1055.



**Point-to-Point Protocol** (**PPP**)

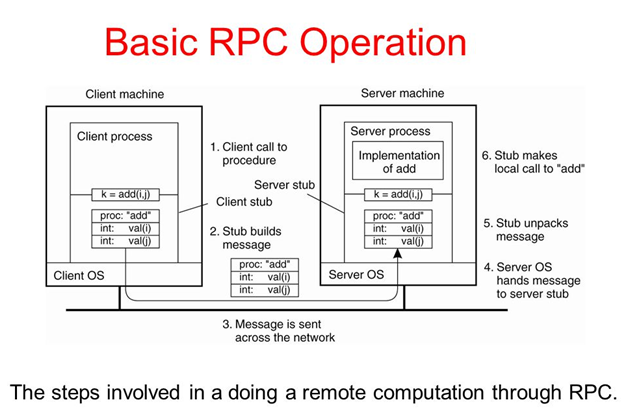
It is used to establish a direct connection between two nodes. It can provide connection authentication, transmission encryption (using ECP, RFC 1968), and compression.

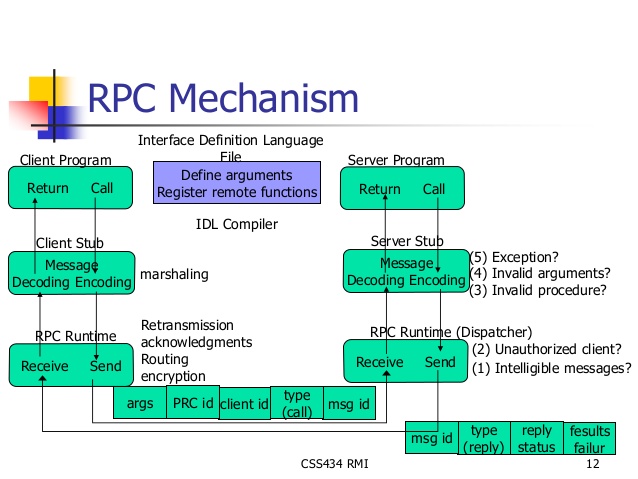


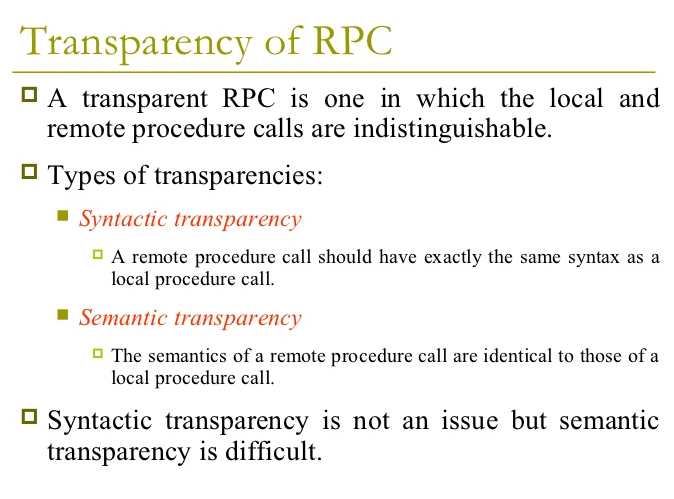
**Remote Procedure Call** (**RPC**)

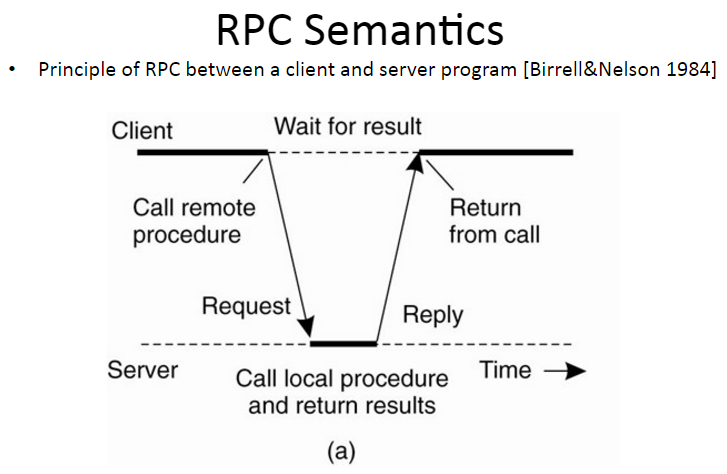
It is a protocol that one program can use to request a service from a program located in another computer in a network without having to understand network details. (A procedure call or function call or a subroutine call.) **RPC** uses the client/server model.

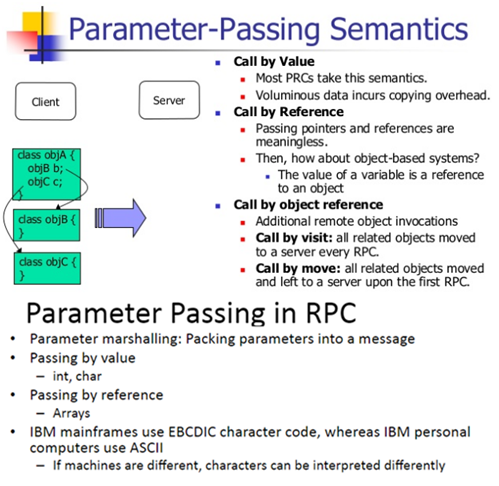
|  |  |
| --- | --- |
| RPC Architecture | Figure 2. Protocols in relation to the Internet layering scheme. |

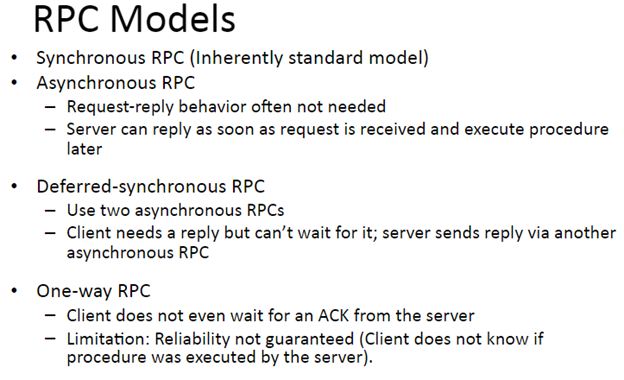


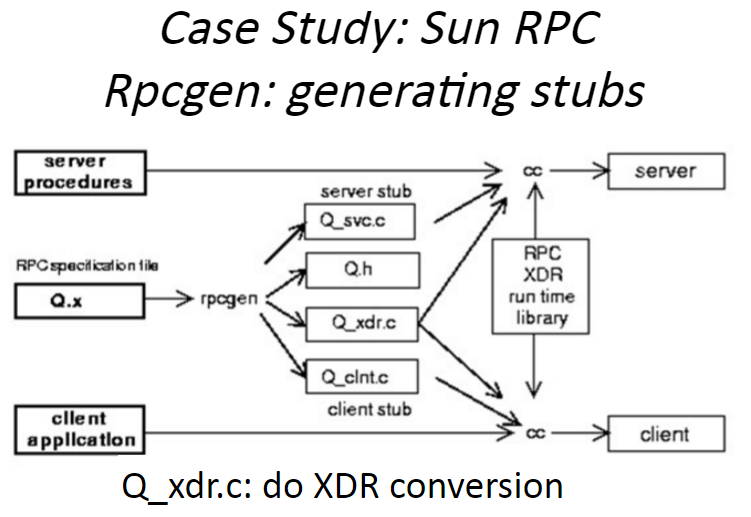
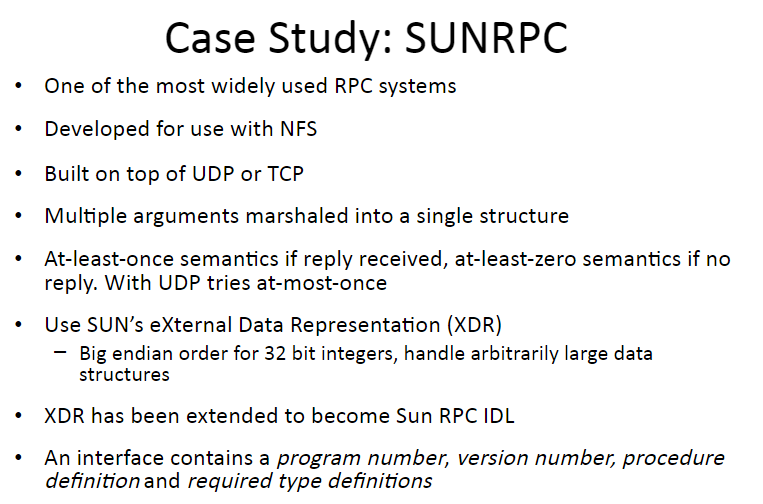
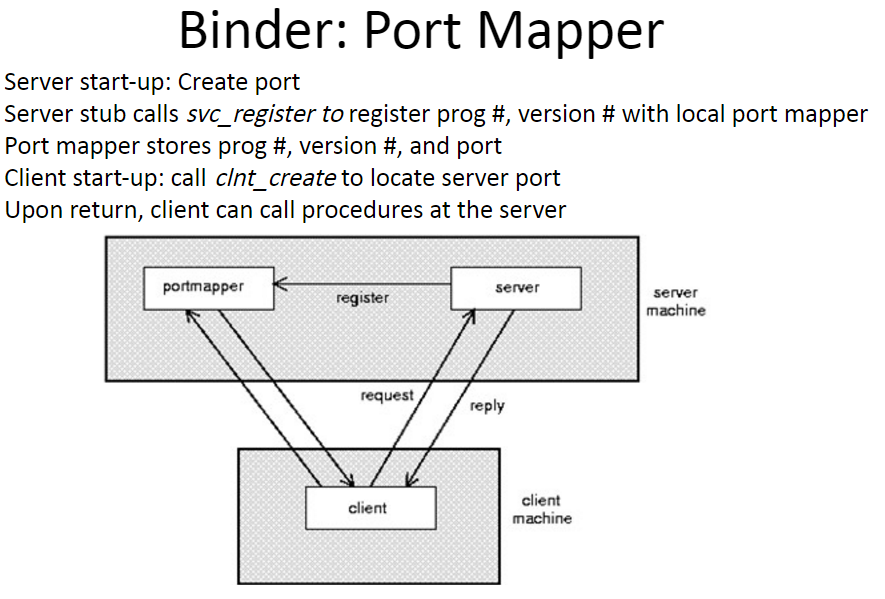
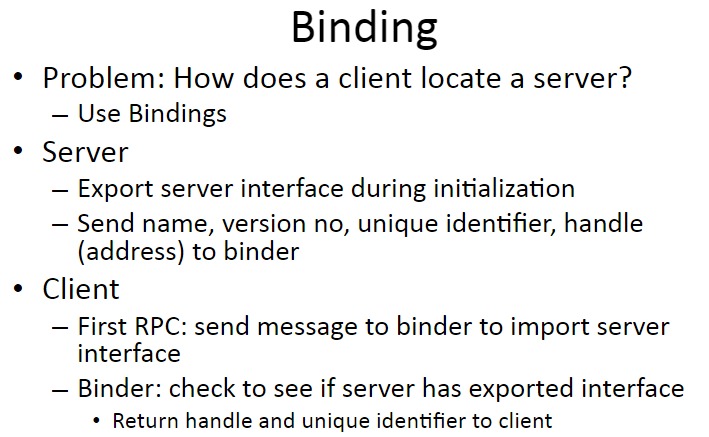
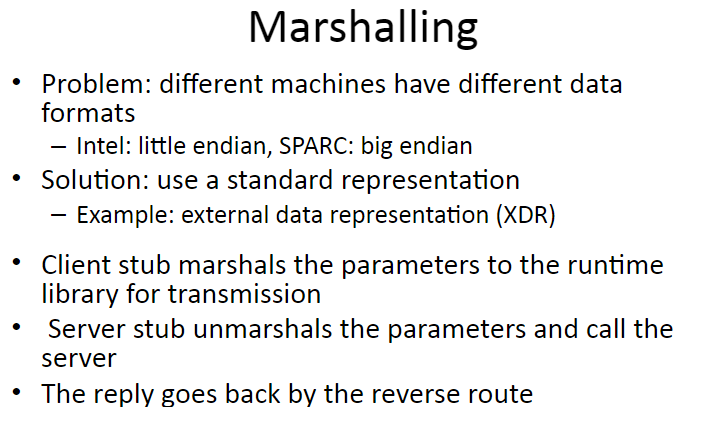
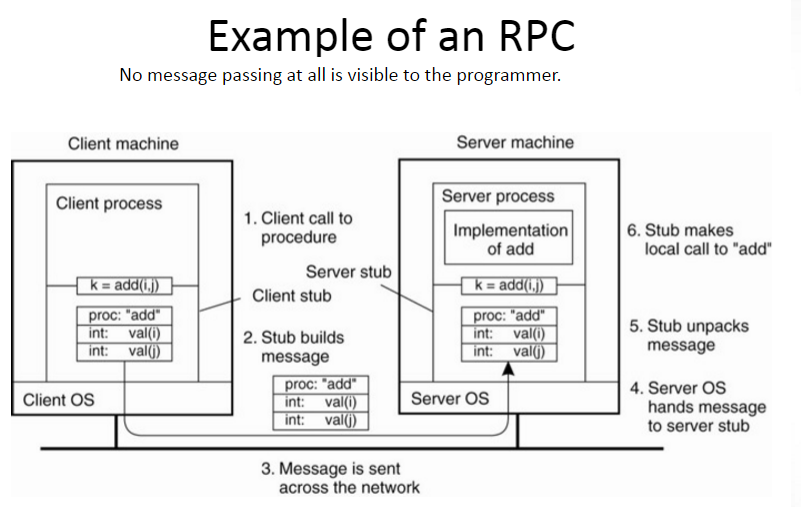
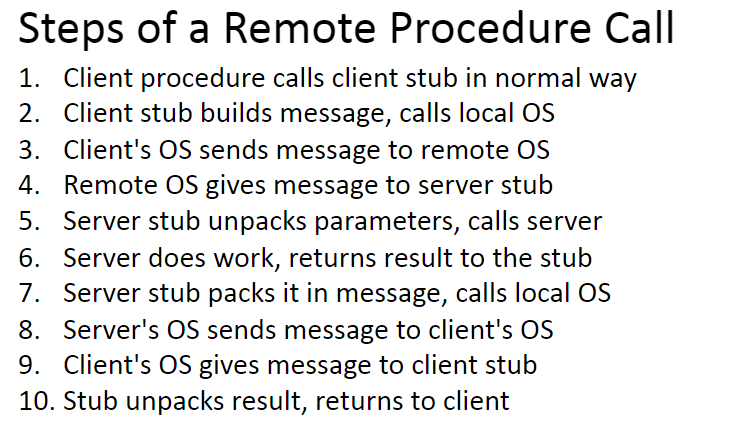
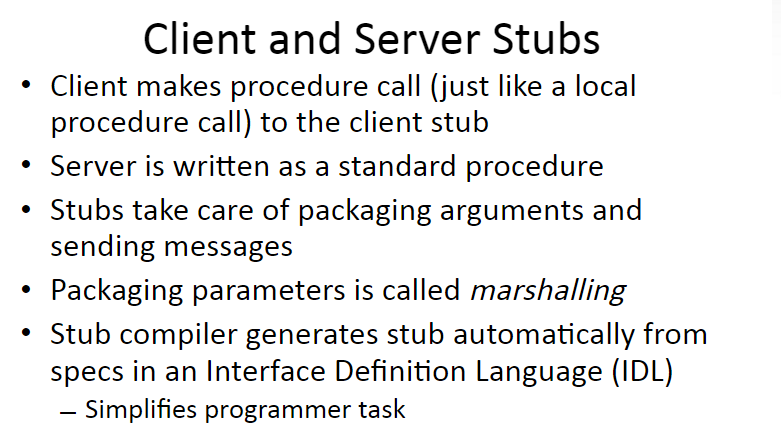
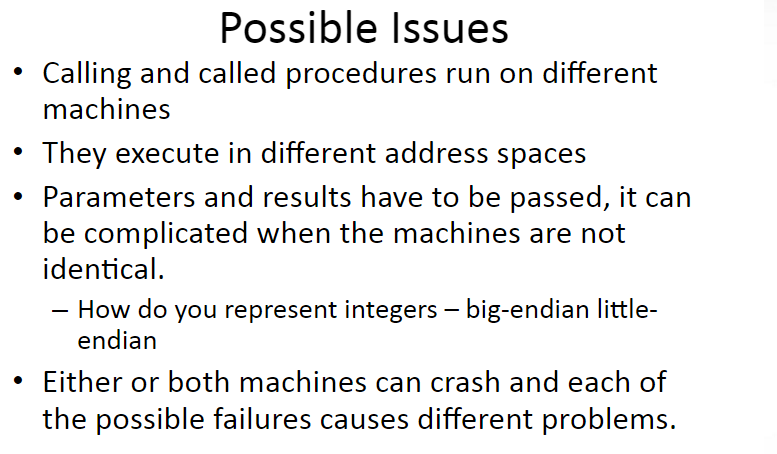
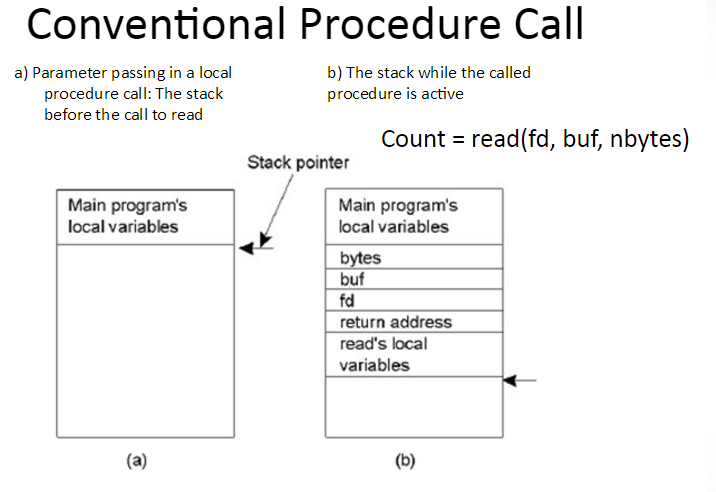
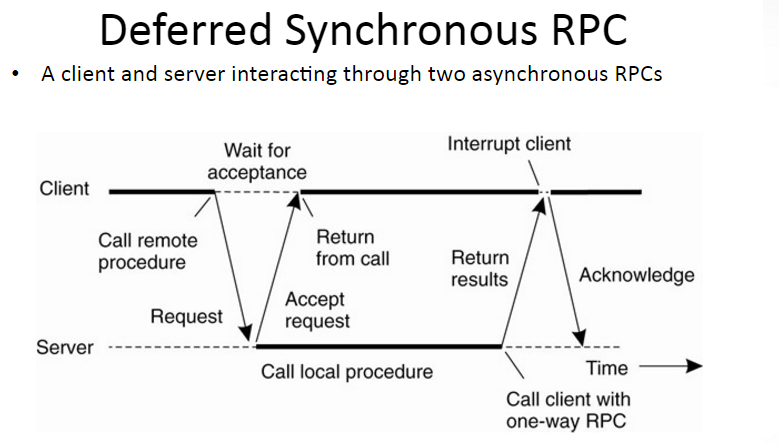
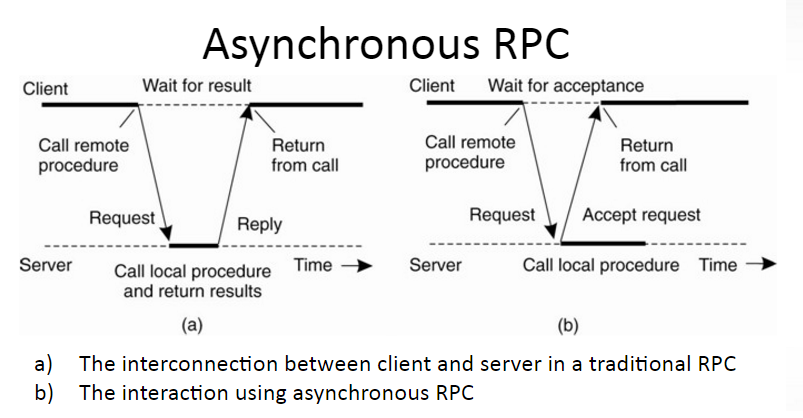












**Remote Object Invocation**

RMI (**Remote Method Invocation**) is a way that a programmer, using the Java programming language and development environment, can write **object**-oriented programming in which **object**s on different computers can interact in a distributed network.

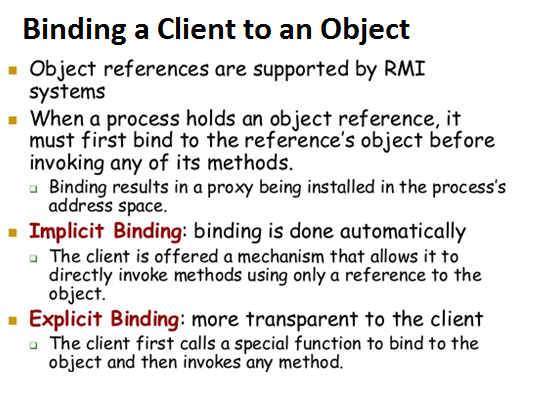
**Distributed objects** are objects (in the sense of [object-oriented programming](https://en.wikipedia.org/wiki/Object-oriented_programming)) that are distributed across different [address spaces](https://en.wikipedia.org/wiki/Address_space), either in multiple [computers](https://en.wikipedia.org/wiki/Computer) connected via a [network](https://en.wikipedia.org/wiki/Computer_network) or even in different [processes](https://en.wikipedia.org/wiki/Process_(computing)) on the same computer, but which work together by sharing data and invoking methods.

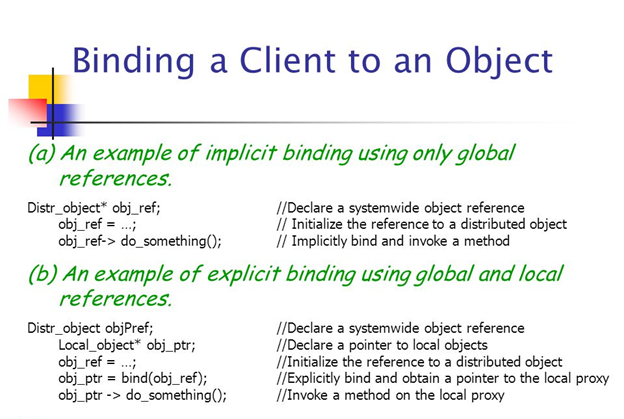
The main method of [distributed object communication](https://en.wikipedia.org/wiki/Distributed_object_communication) is with [remote method invocation](https://en.wikipedia.org/wiki/Remote_method_invocation), generally by message-passing: one object sends a message to another object in a remote machine or process to perform some task. The results are sent back to the calling object.

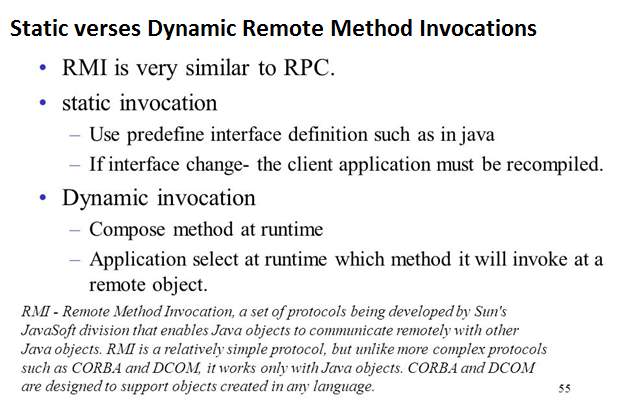
## Local vs. Distributed Objects

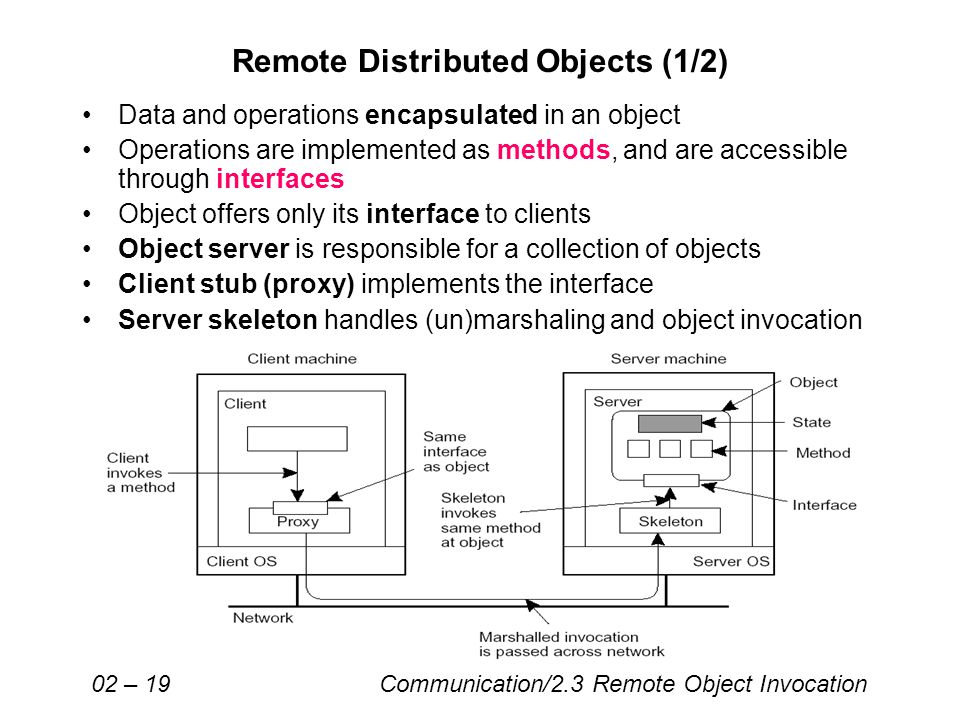
Local and distributed objects differ in many respects. Here are some of them:

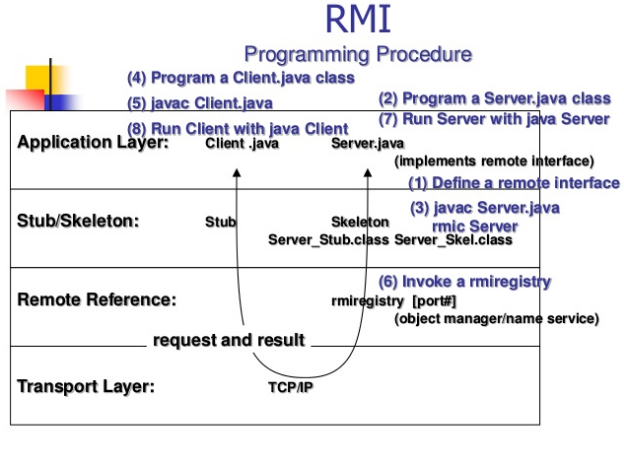
1. **Life cycle** : Creation, migration and deletion of distributed objects is different from local objects
2. **Reference** : Remote references to distributed objects are more complex than simple pointers to memory addresses
3. **Request Latency** : A distributed object request is orders of magnitude slower than local method invocation
4. **Object Activation** : Distributed objects may not always be available to serve an object request at any point in time
5. **Parallelism**: Distributed objects may be executed in parallel.
6. **Communication** : There are different communication primitives available for distributed objects requests
7. **Failure**: Distributed objects have far more points of failure than typical local objects.
8. **Security**: Distribution makes them vulnerable to attack.

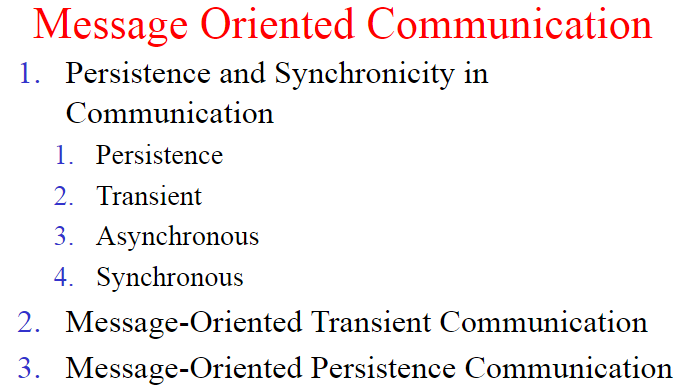


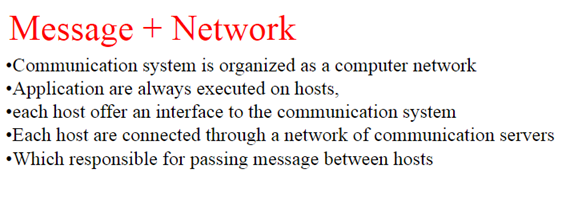


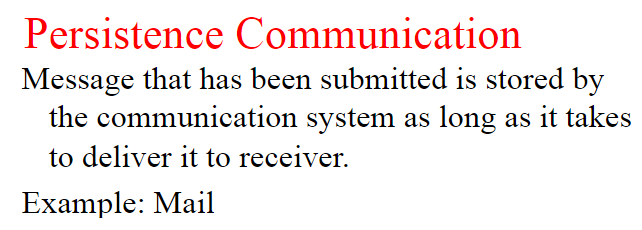


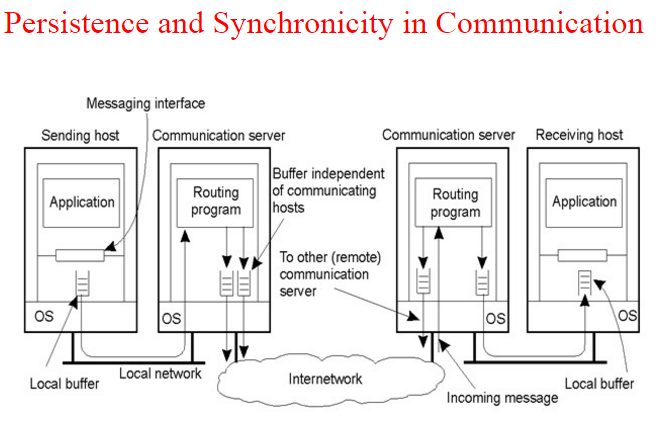


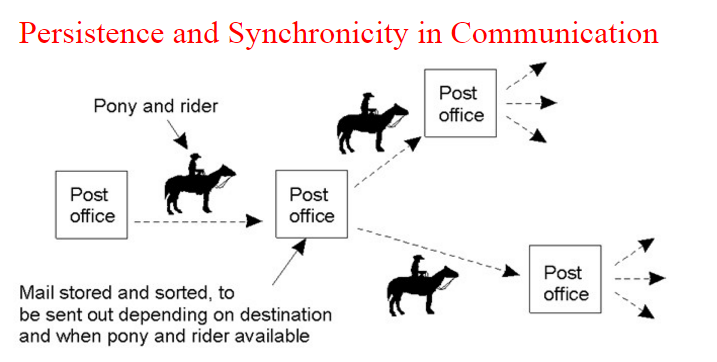


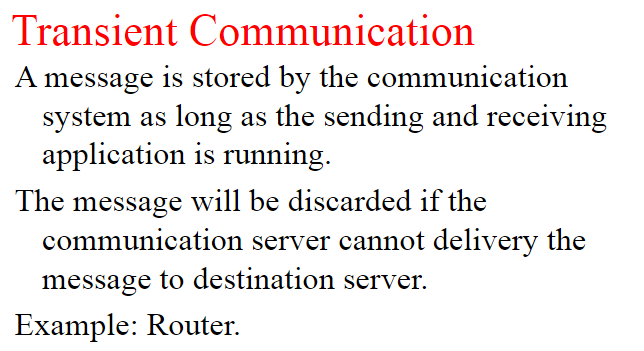


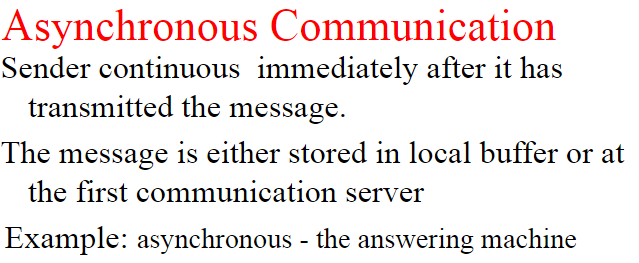


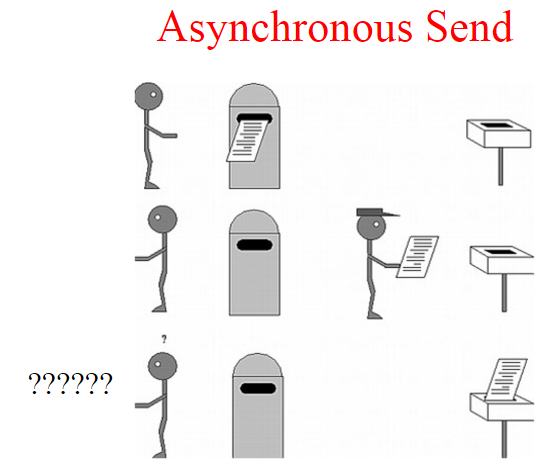


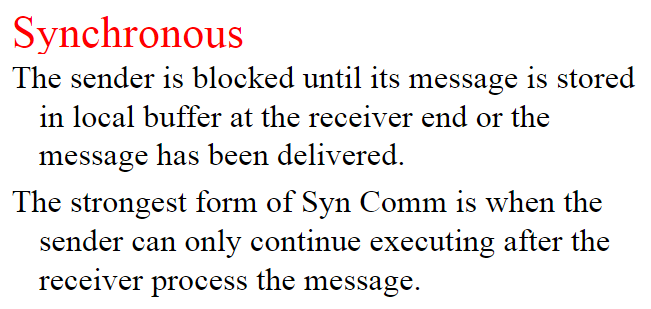


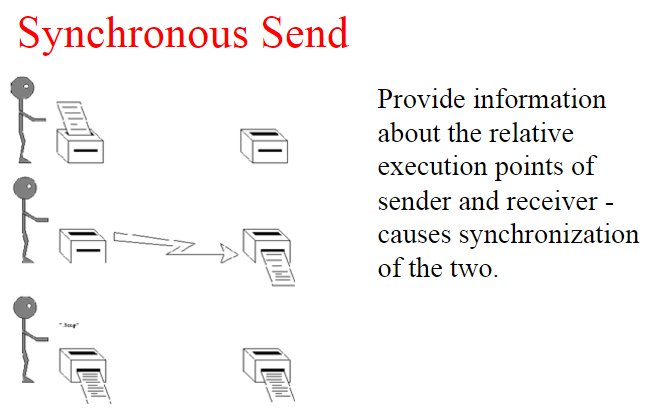


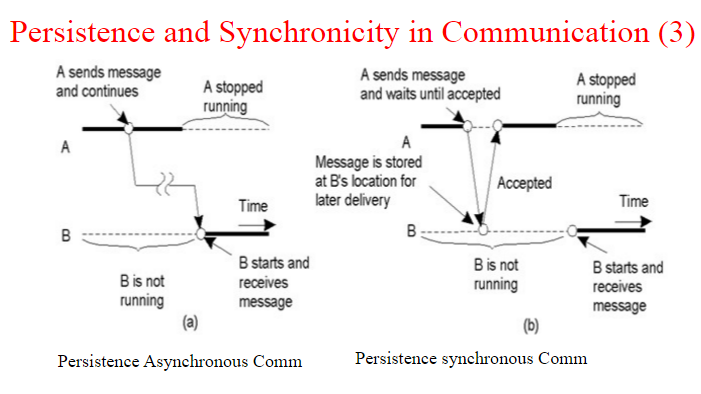


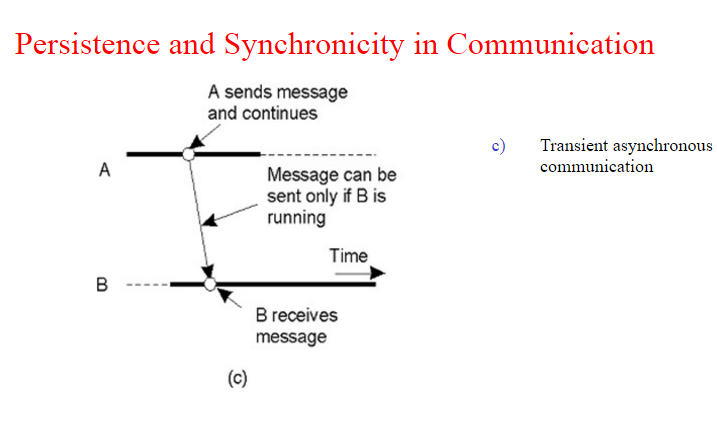


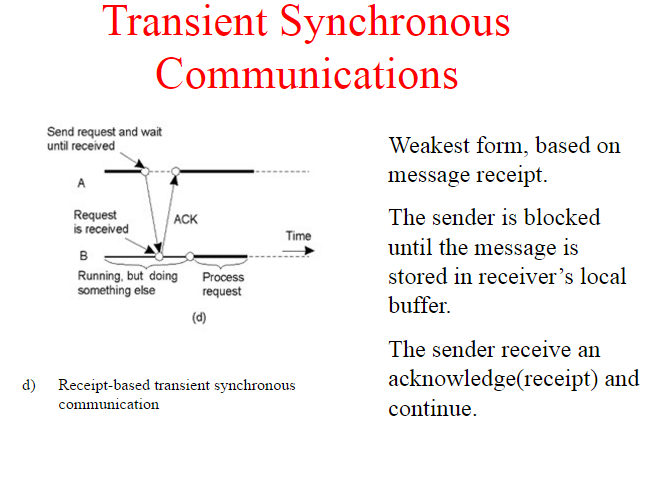


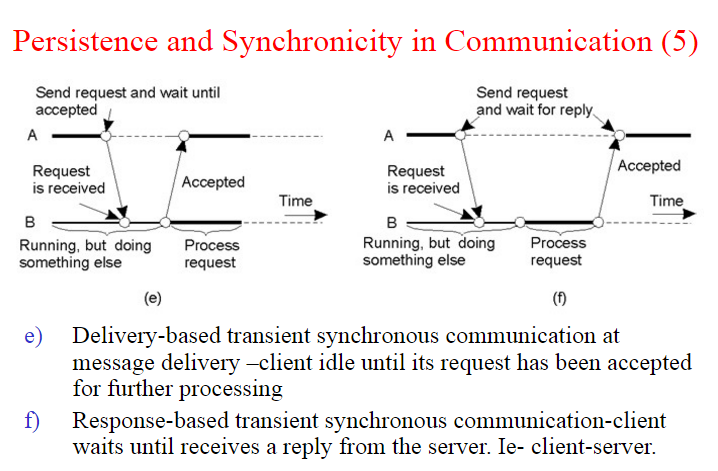


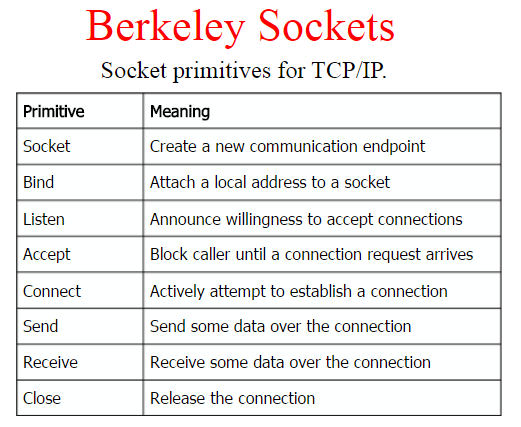


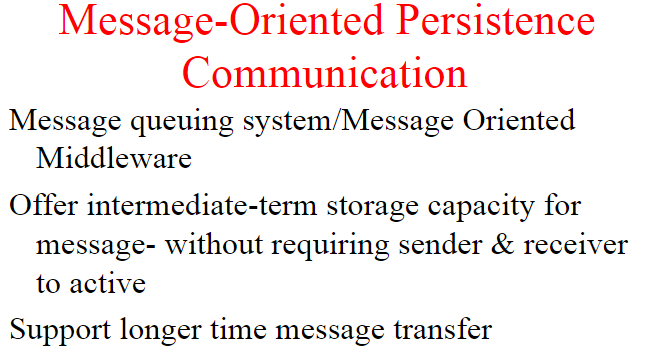


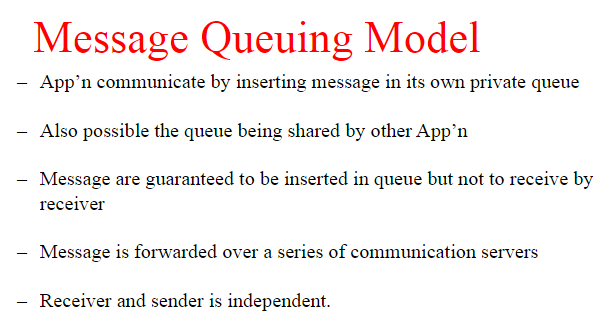


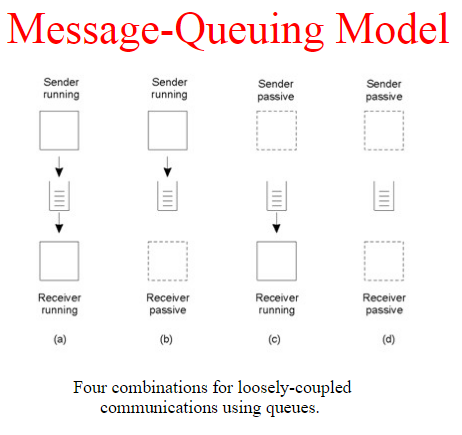


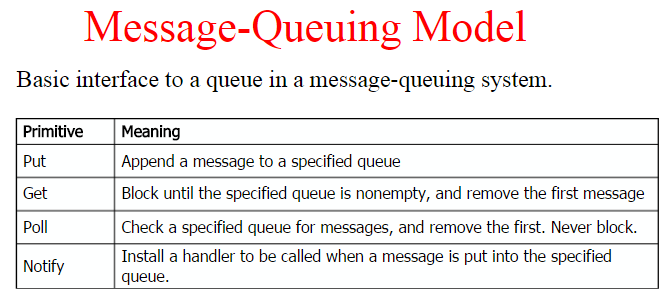


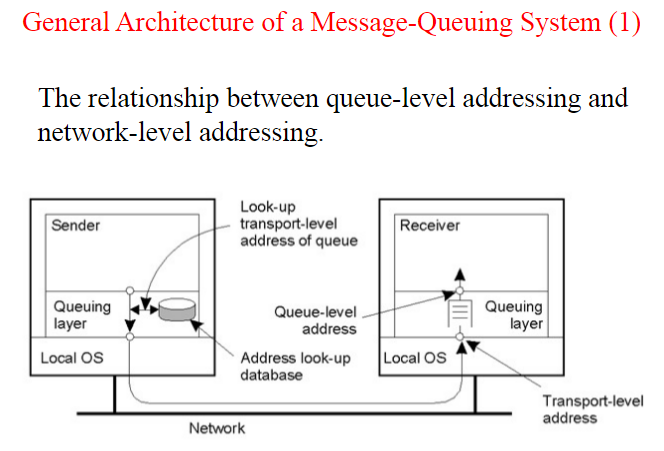


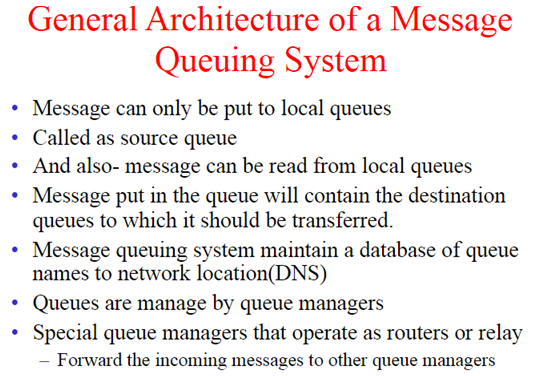


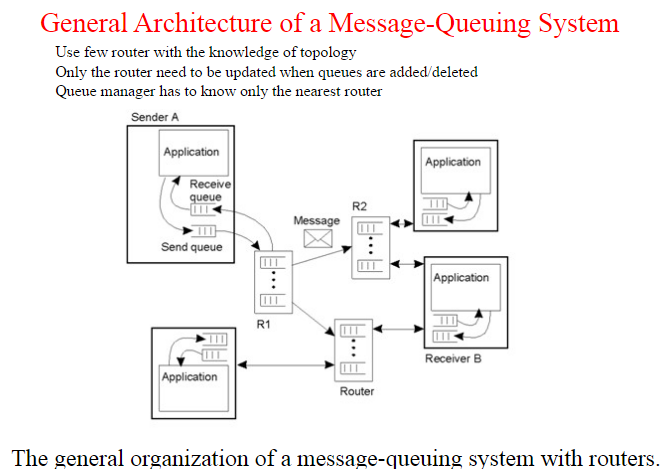


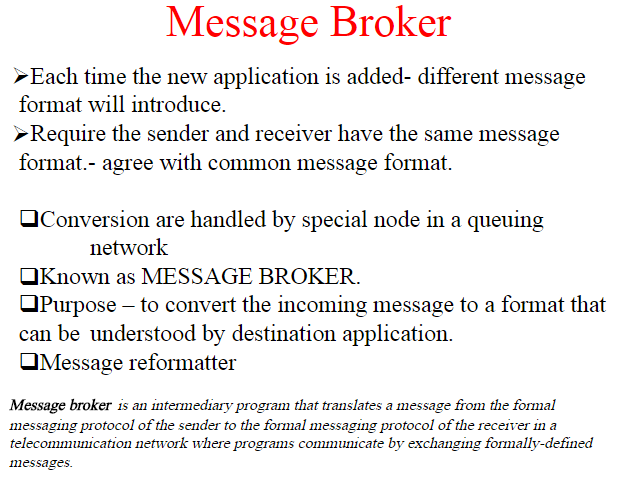


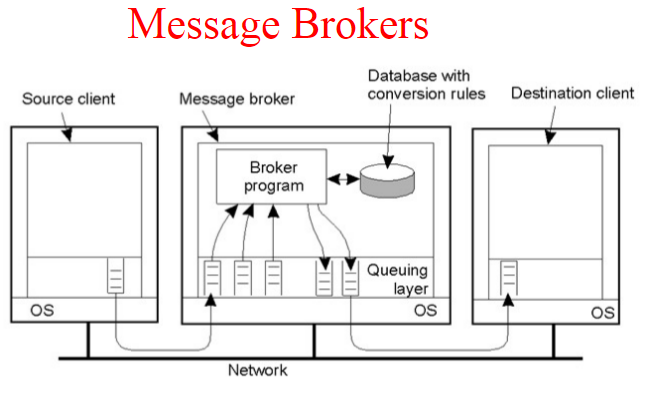


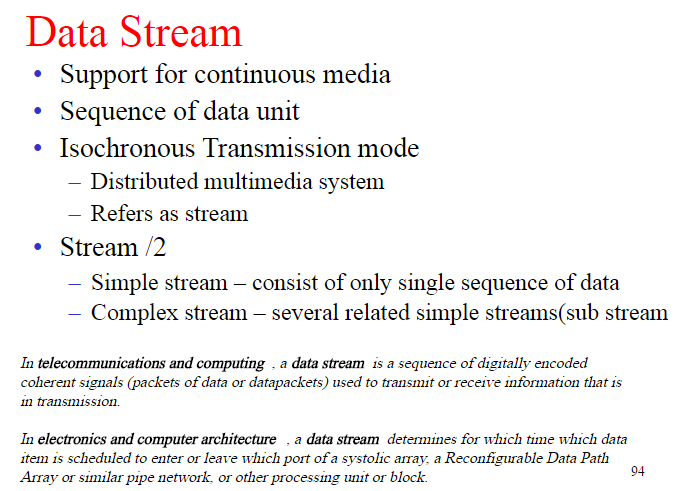


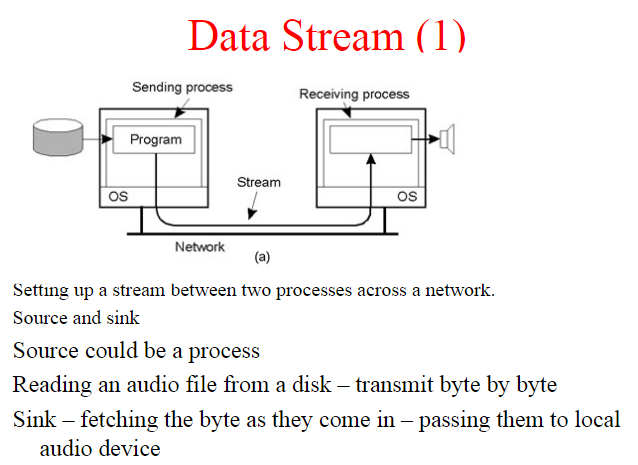


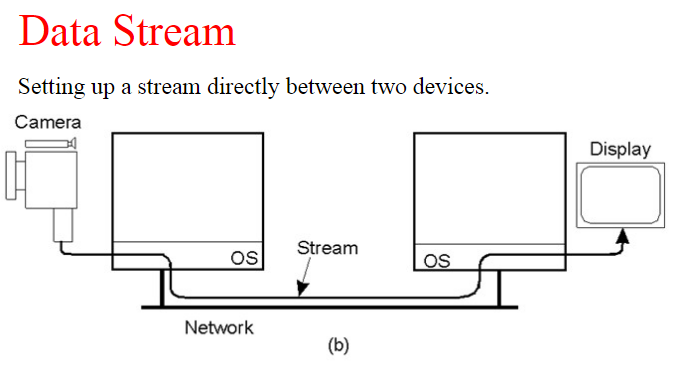


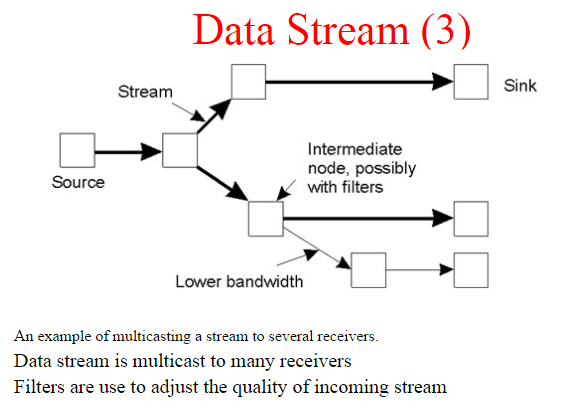


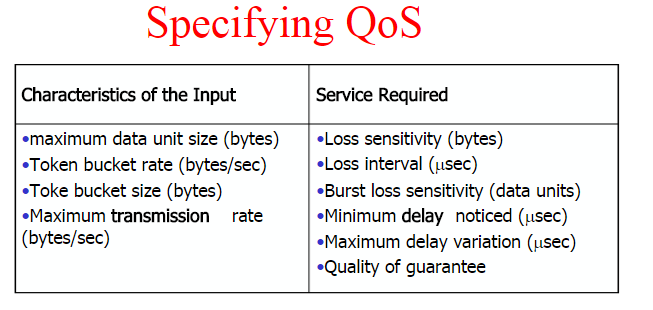


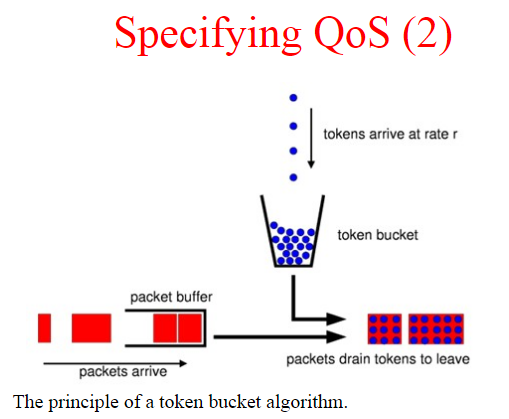


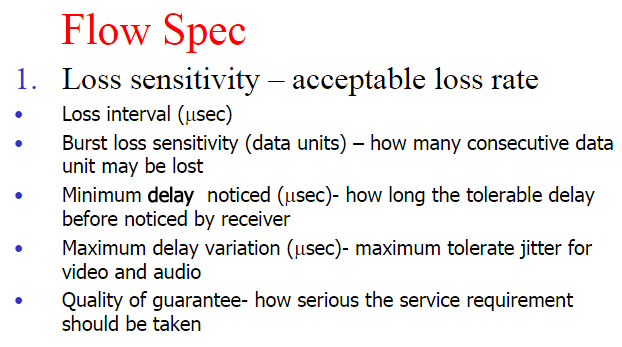


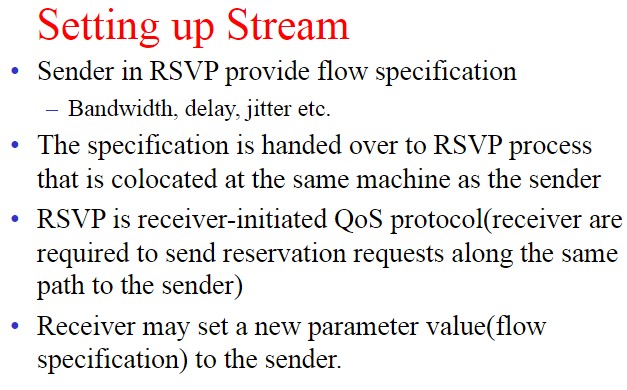


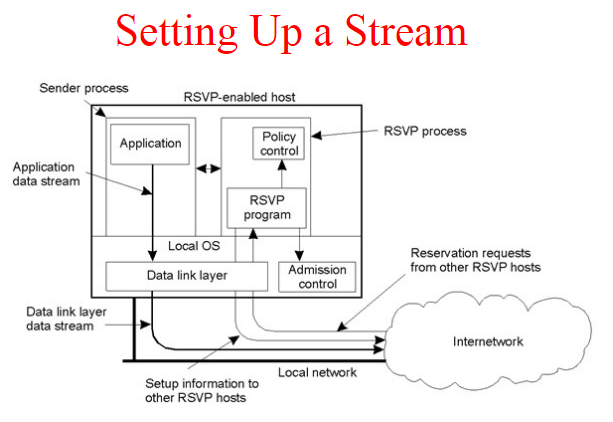


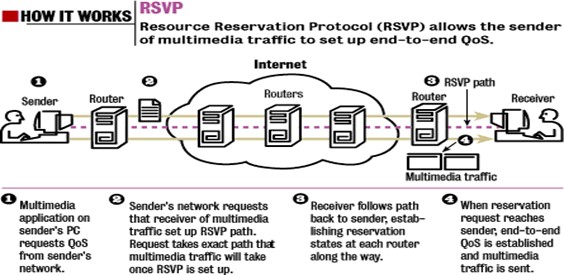


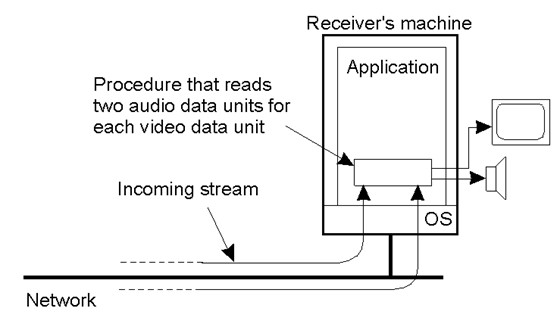


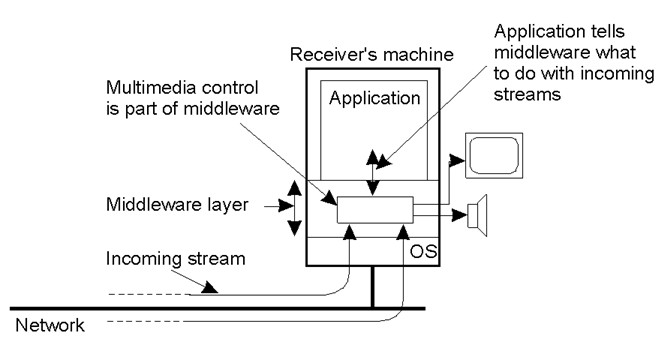












**Complete**

<http://slideplayer.com/slide/5179691/>