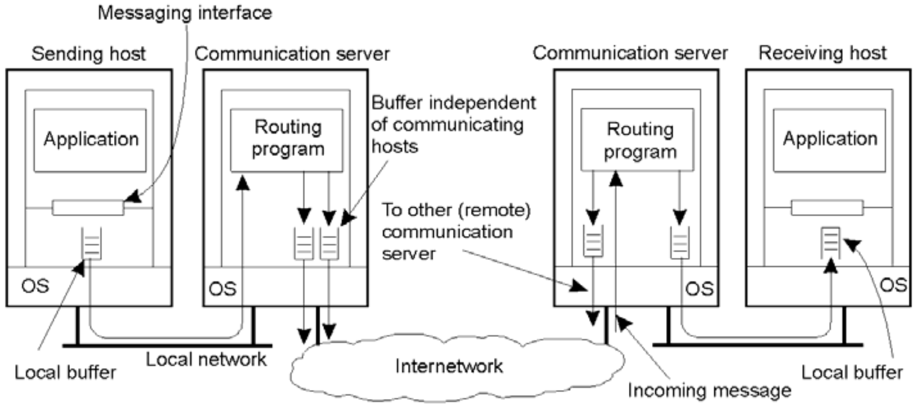
Message-oriented Middleware

Distributed Software Systems

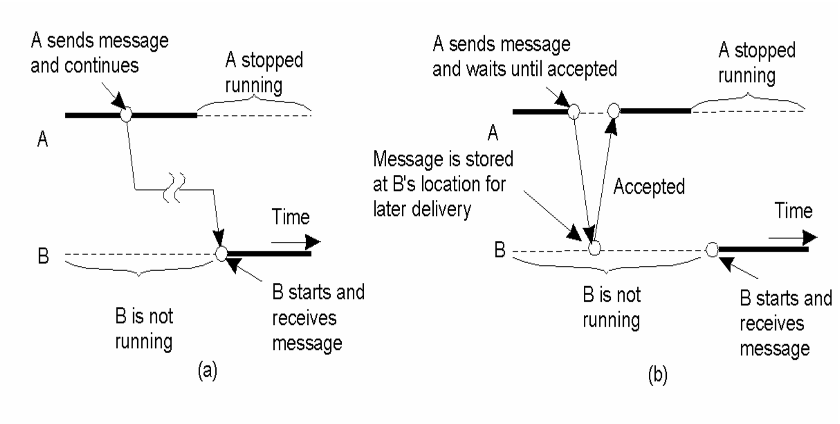
Message-oriented middleware



Transient vs Persistent communication

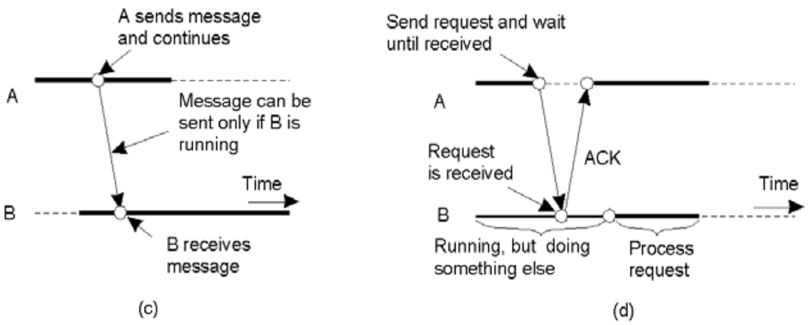
- RPC and RMI (by default) support communication between two processes that are executing at the same time
  - transient communication
- Typically, the client is blocked until the RPC/RMI returns
  - synchronous communication
- Not suitable for middleware that integrates applications in widely dispersed and large-scale distributed systems
  - Message-oriented middleware

Persistence and Synchronicity in Communication (1)


A sends message and continues
A stopped running
A
B
Time
B is not running
B starts and receives message
(a)
A sends message and waits until accepted
A stopped running
A
Message is stored at B's location for later delivery
Accepted
Time
B
B is not running
B starts and receives message
(b)

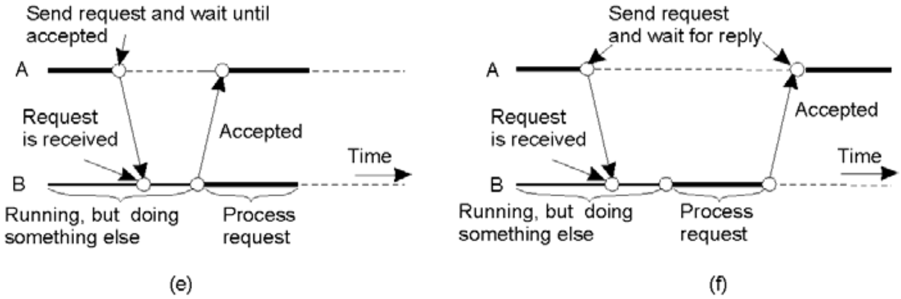
a) Persistent asynchronous communication
b) Persistent synchronous communication

Persistence and Synchronicity in Communication (2)



c) Transient asynchronous communication
d) Receipt-based transient synchronous communication

Persistence and Synchronicity in Communication (3)



e) Delivery-based transient synchronous communication at message delivery
f) Response-based transient synchronous communication

Message-oriented transient communication (1)

- Socket primitives for TCP/IP.

Primitive	**Meaning**
Socket	Create a new communication endpoint
Bind	Attach a local address to a socket
Listen	Announce willingness to accept connections
Accept	Block caller until a connection request arrives
Connect	Actively attempt to establish a connection
Send	Send some data over the connection
Receive	Receive some data over the connection
Close	Release the connection

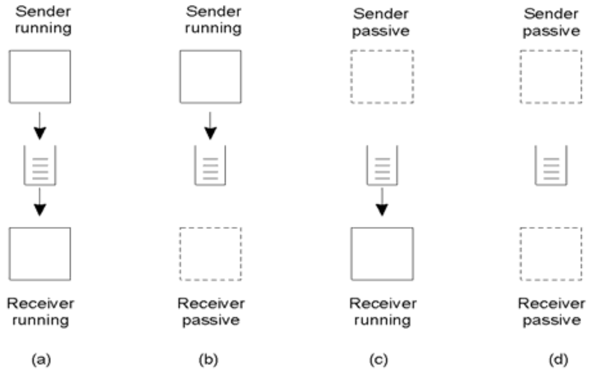
Message-oriented Transient communication (2)

- The Message-Passing Interface (MPI)
- Used for developing message-passing parallel applications

Primitive	**Meaning**
MPI_bsend	Append outgoing message to a local send buffer
MPI_send	Send a message and wait until copied to local or remote buffer
MPI_ssend	Send a message and wait until receipt starts
MPI_sendrecv	Send a message and wait for reply
MPI_isend	Pass reference to outgoing message, and continue
MPI_issend	Pass reference to outgoing message, and wait until receipt starts
MPI_recv	Receive a message; block if there are none
MPI_irecv	Check if there is an incoming message, but do not block

Message-Queuing Model (1)

- Four combinations for loosely-coupled communications using queues.



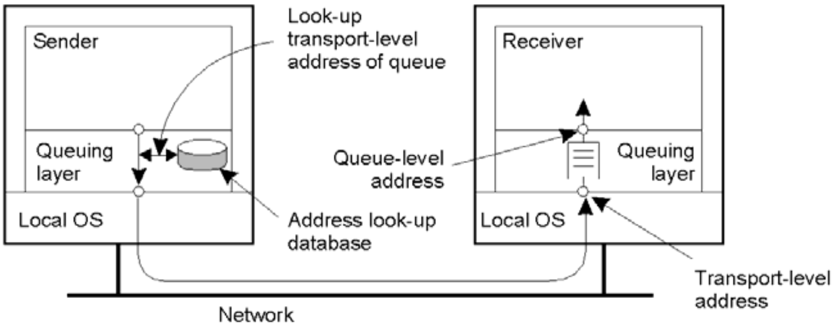
Message-Queuing Model (2)

- Basic interface to a queue in a message-queuing system.

Primitive	Meaning
Put	Append a message to a specified queue
Get	Block until the specified queue is nonempty, and remove the first message
Poll	Check a specified queue for messages, and remove the first. Never block.
Notify	Install a handler to be called when a message is put into the specified queue.

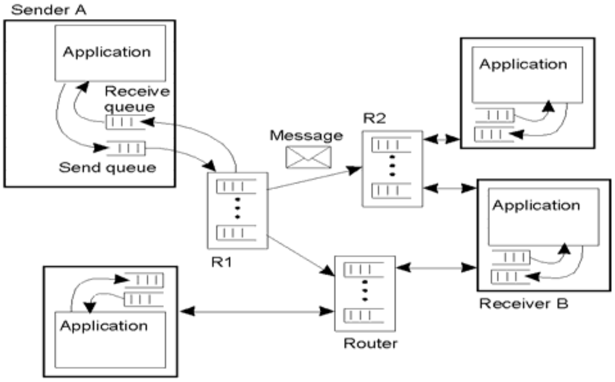
General Architecture of a Message-Queuing System (1)

- The relationship between queue-level addressing and network-level addressing.



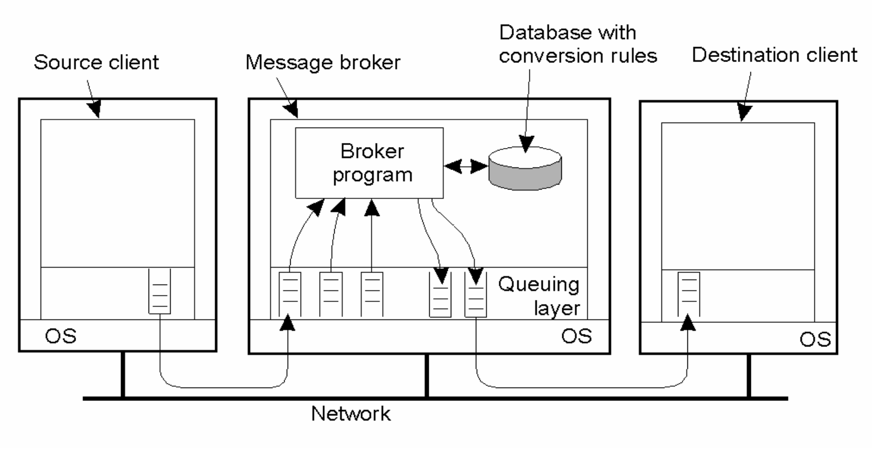
General Architecture of a Message-Queuing System (2)

- The general organization of a message-queuing system with routers.



Message Brokers

- The general organization of a message broker in a message-queuing system.

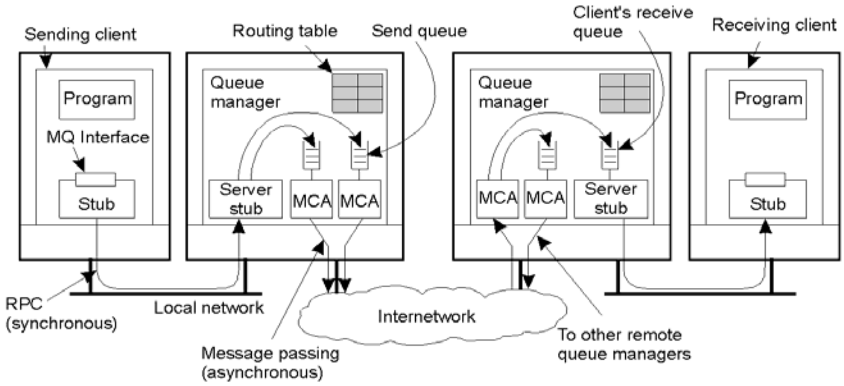


Message-queuing systems vs Email systems

- The underlying architecture of message-queuing systems is very similar to that for email services
- The difference is that email systems primarily provide support for end users
- General message-queuing systems enable persistent communication between processes regardless of what the process is doing
  - leads to different requirements, e.g., guaranteed message delivery, message priorities, logging facilities, load balancing, fault tolerance, etc.

Example: IBM MQSeries

- General organization of IBM's MQSeries message-queuing system.


Sending client
Routing table
Send queue
Client's receive queue
Receiving client
Program
MQ Interface
Stub
Queue manager
Server stub
MCA
MCA
Queue manager
MCA
MCA
Server stub
Program
Stub
RPC (synchronous)
Local network
Internetwork
Message passing (asynchronous)
To other remote queue managers