

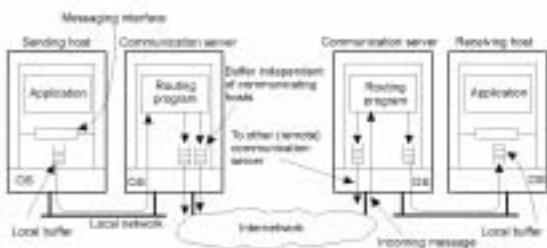
Inter-process Communication

Yong Guan
3216 Coover
Tel: (515) 294-8378
Email: guan@ee.iastate.edu
February 26, 2004

Readings for Today's Lecture

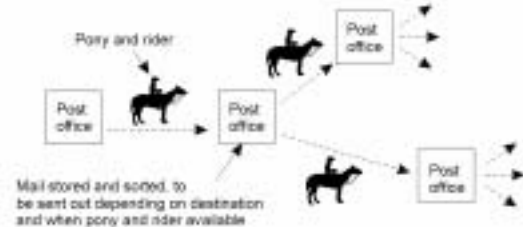
- References
 - Chapter 2 of "Distributed Systems: Principles and Paradigms"

Persistence and Synchronicity in Communication (1)



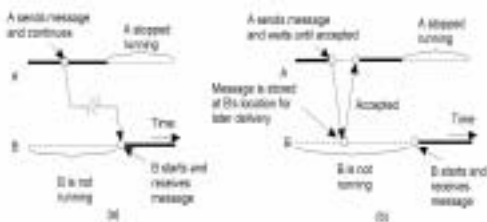
- ◆ General organization of a communication system in which hosts are connected through a network

Persistence and Synchronicity in Communication (2)



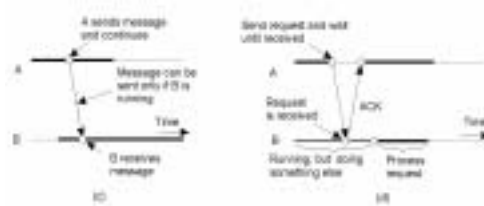
- ◆ Persistent communication of letters back in the days of the Pony Express.

Persistence and Synchronicity in Communication (3)

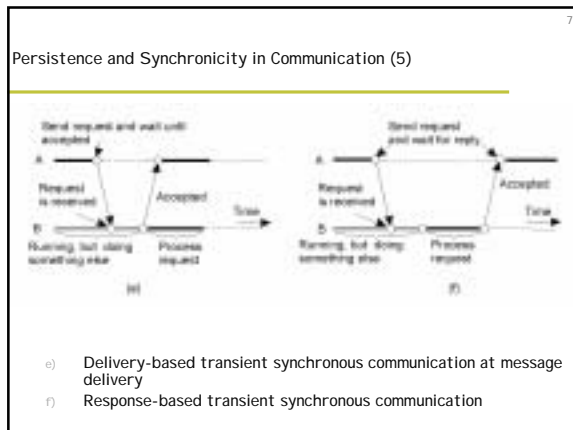


- a) Persistent asynchronous communication
- b) Persistent synchronous communication

Persistence and Synchronicity in Communication (4)



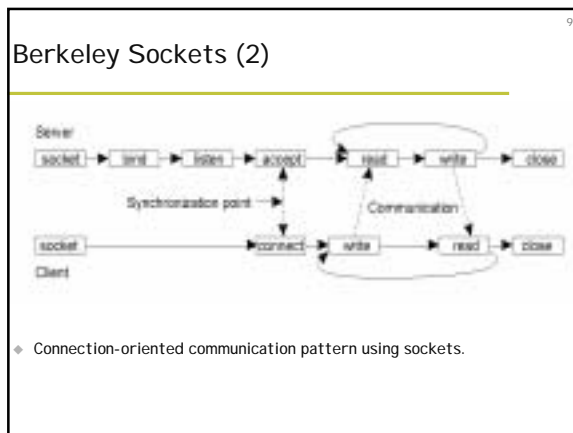
- c) Transient asynchronous communication
- d) Receipt-based transient synchronous communication



Berkeley Sockets (1)

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

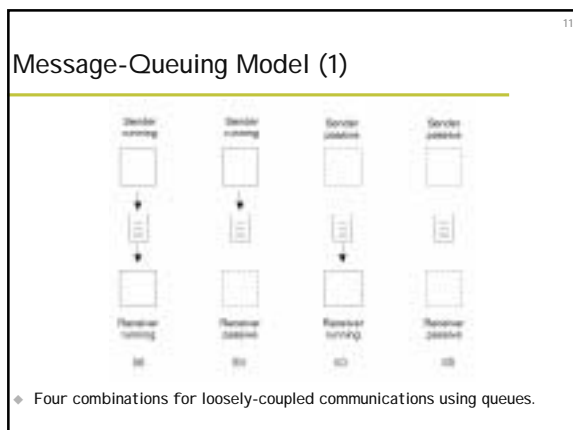
Socket primitives for TCP/IP.



The Message-Passing Interface (MPI)

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_issend	Pass reference to outgoing message, and continue
MPI_lssend	Pass reference to outgoing message, and wait until receipt starts
MPI_recv	Receive a message: block if there are none
MPI_lrecv	Check if there is an incoming message, but do not block

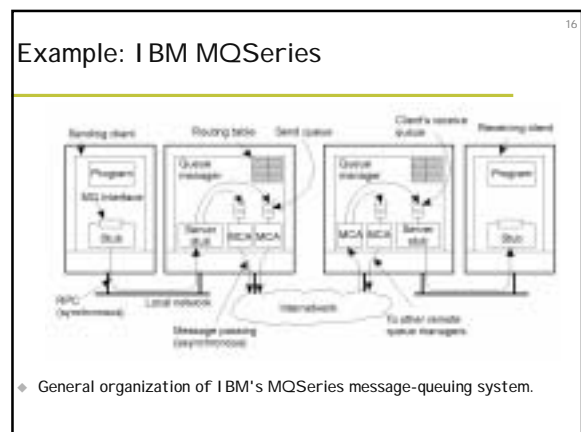
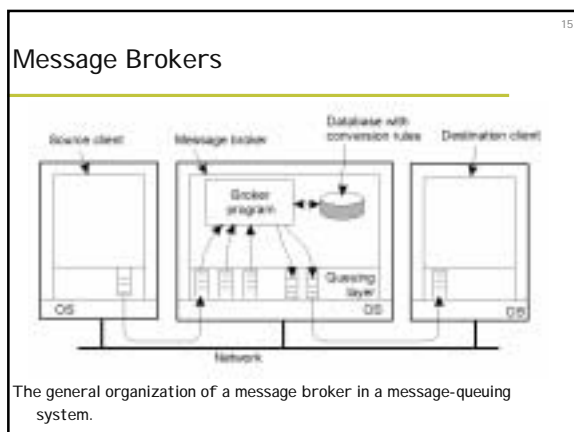
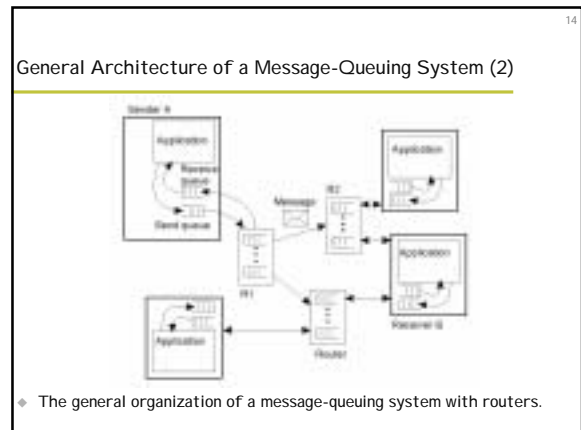
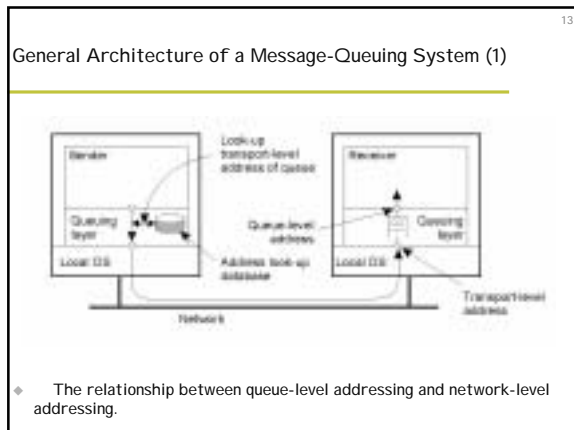
◆ Some of the most intuitive message-passing primitives of MPI.



Message-Queuing Model (2)

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.

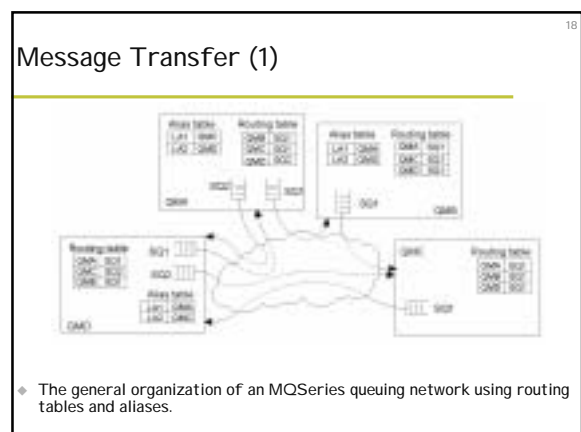
◆ Basic interface to a queue in a message-queuing system.



Channels

Attribute	Description
Transport type	Determines the transport protocol to be used
FIFO delivery	Indicates that messages are to be delivered in the order they are sent
Message length	Maximum length of a single message
Setup retry count	Specifies maximum number of retries to start up the remote MCA
Delivery retries	Maximum times MCA will try to put received message into queue

◆ Some attributes associated with message channel agents.

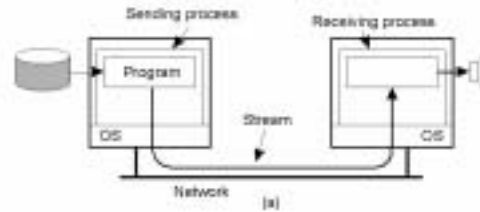


Message Transfer (2)

Primitive	Description
MQOpen	Open a (possibly remote) queue
MQClose	Close a queue
MQput	Put a message into an opened queue
MQget	Get a message from a (local) queue

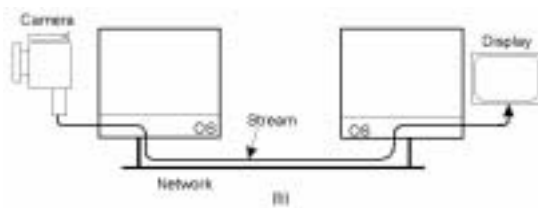
- ◆ Primitives available in an IBM MQSeries MQI

Data Stream (1)



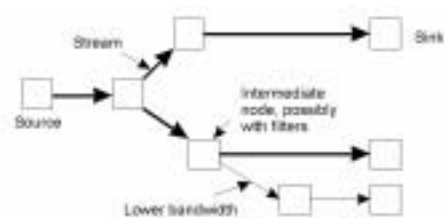
- ◆ Setting up a stream between two processes across a network.

Data Stream (2)



- ◆ Setting up a stream directly between two devices.

Data Stream (3)



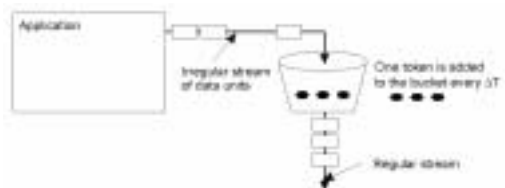
- ◆ An example of multicasting a stream to several receivers.

Specifying QoS (1)

Characteristics of the Input	Service Required
<ul style="list-style-type: none"> ◆ maximum data unit size (bytes) ◆ Token bucket rate (bytes/sec) ◆ Token bucket size (bytes) ◆ Maximum transmission rate (bytes/sec) 	<ul style="list-style-type: none"> ◆ Loss sensitivity (bytes) ◆ Loss interval (μsec) ◆ Burst loss sensitivity (data units) ◆ Minimum delay noticed (μsec) ◆ Maximum delay variation (μsec) ◆ Quality of guarantee

- ◆ A flow specification.

Specifying QoS (2)



- ◆ The principle of a token bucket algorithm.

