ECE151 – Lecture 3

Chapter 2

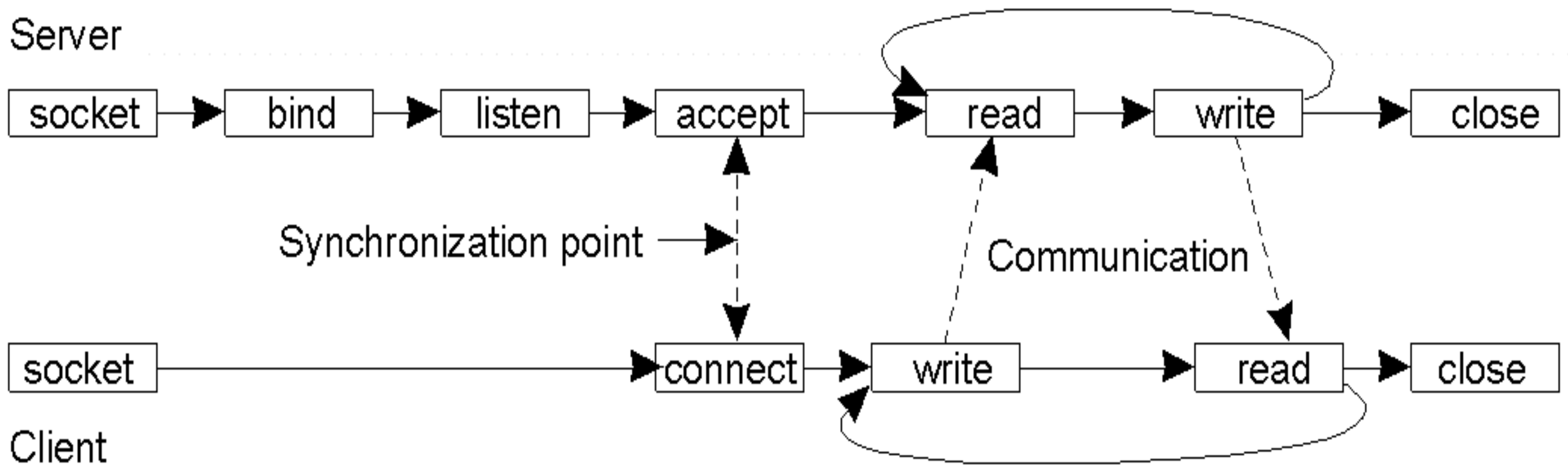
Communication

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.

Berkeley Sockets (2)



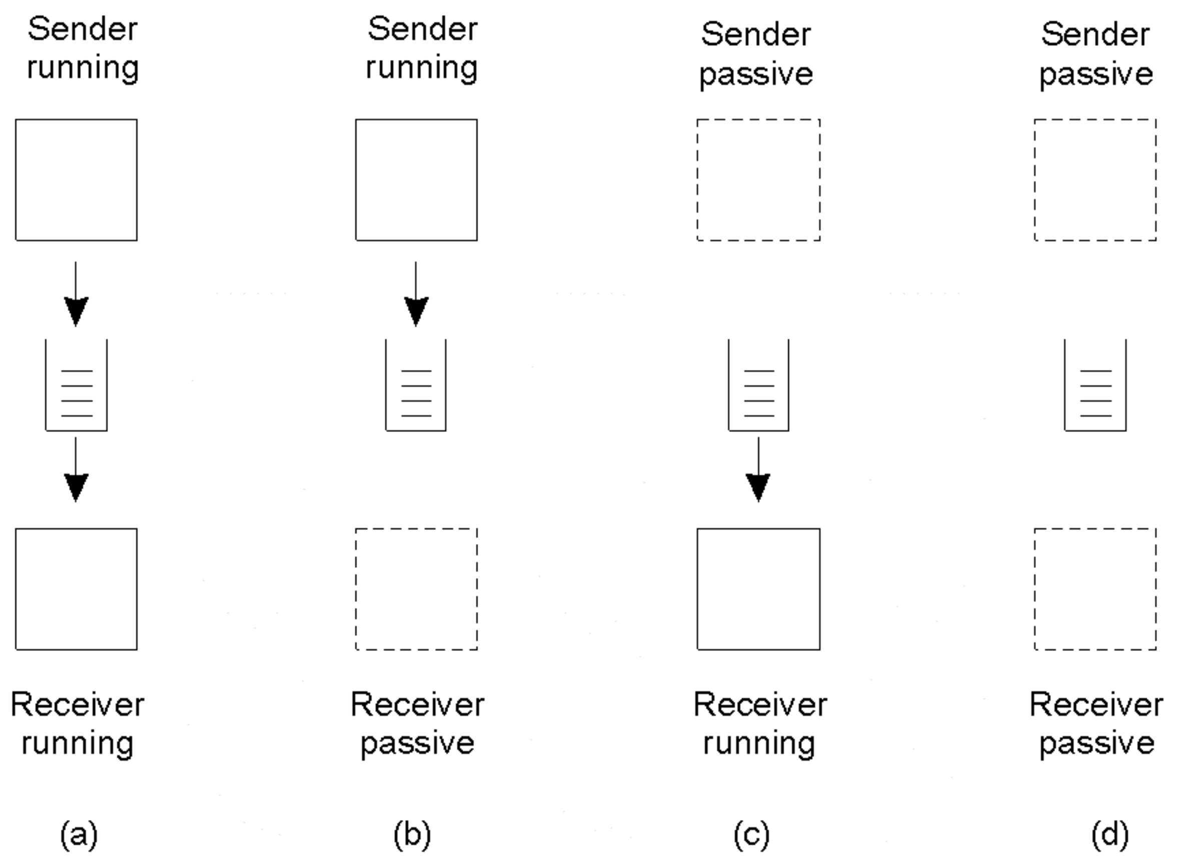
Connection-oriented communication pattern using sockets.

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_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

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

Message-Queuing Model (1)



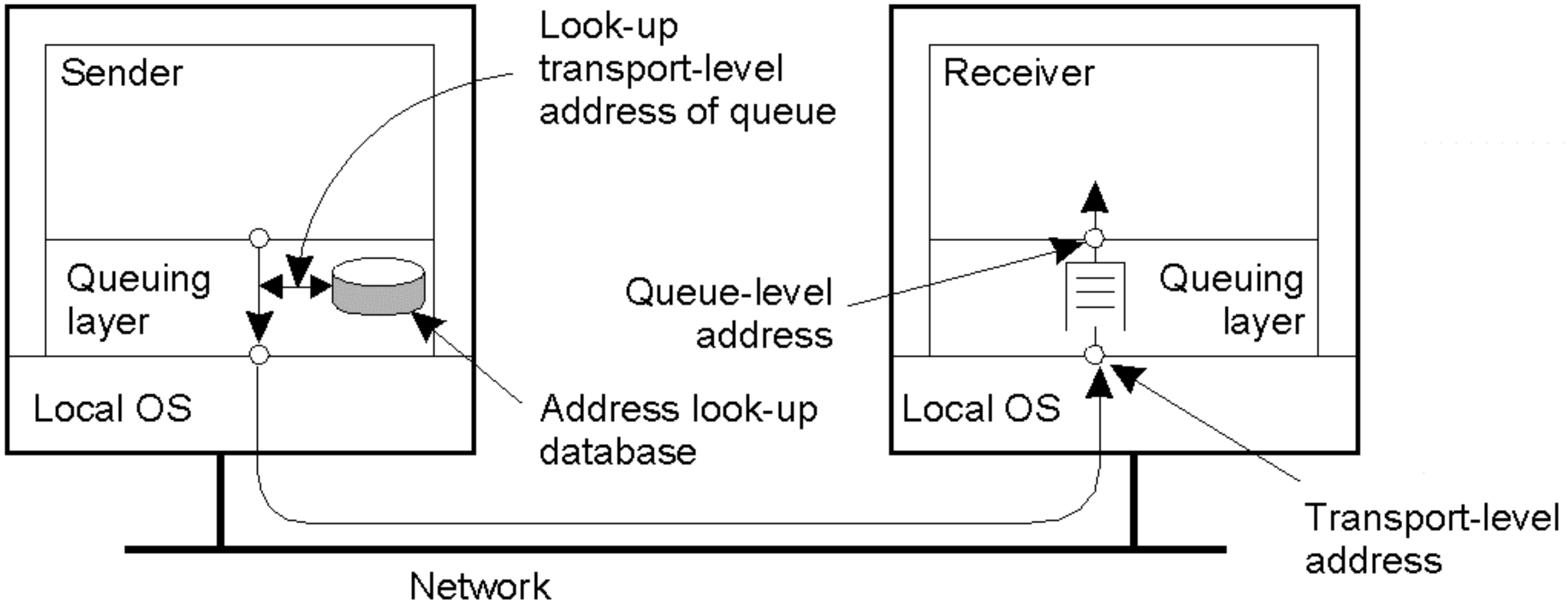
Four combinations for loosely-coupled communications using queues.

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.

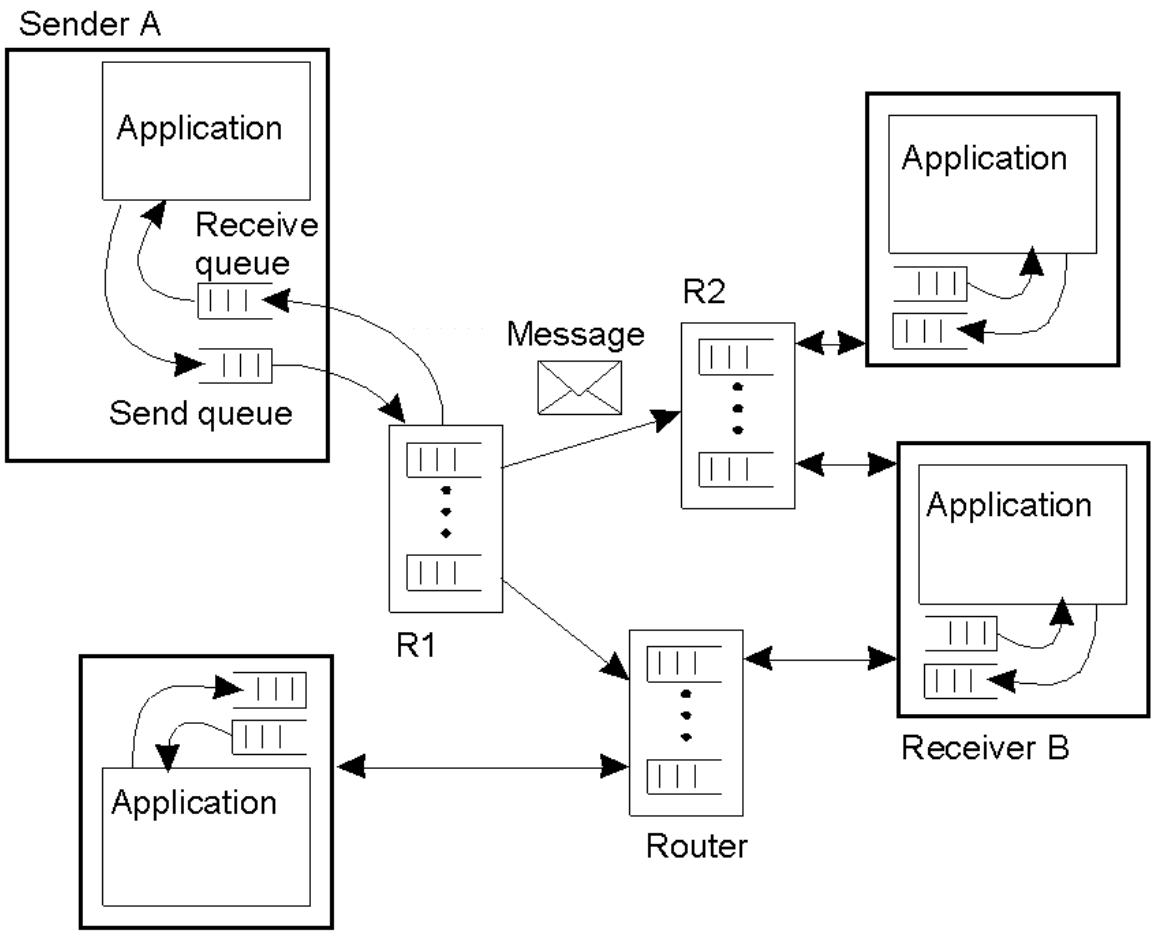
Basic interface to a queue in a message-queuing system.

General Architecture of a Message-Queuing System (1)



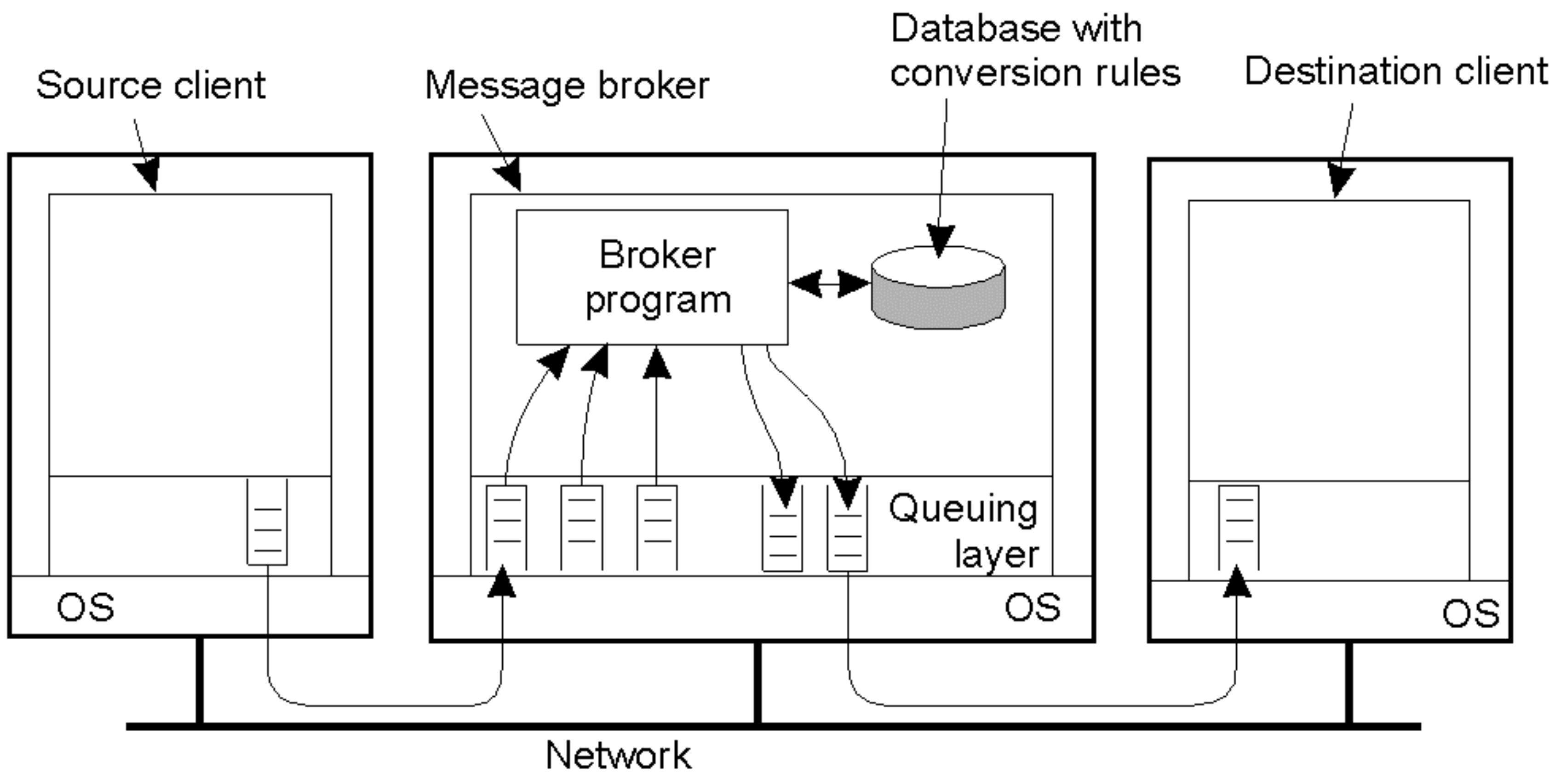
The relationship between queue-level addressing and network-level addressing.

General Architecture of a Message-Queuing System (2)



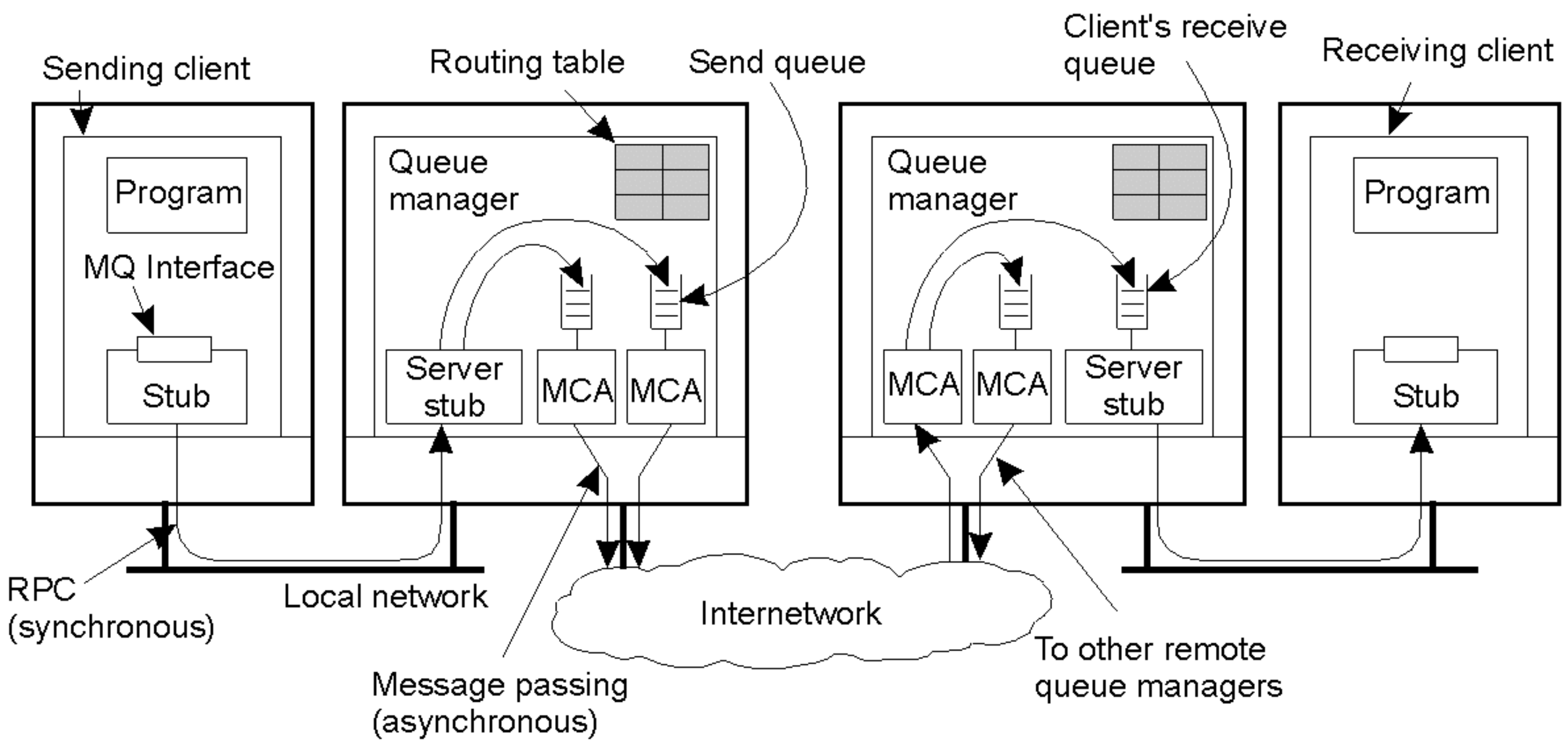
The general organization of a message-queuing system with routers.

Message Brokers



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

Example: IBM MQSeries



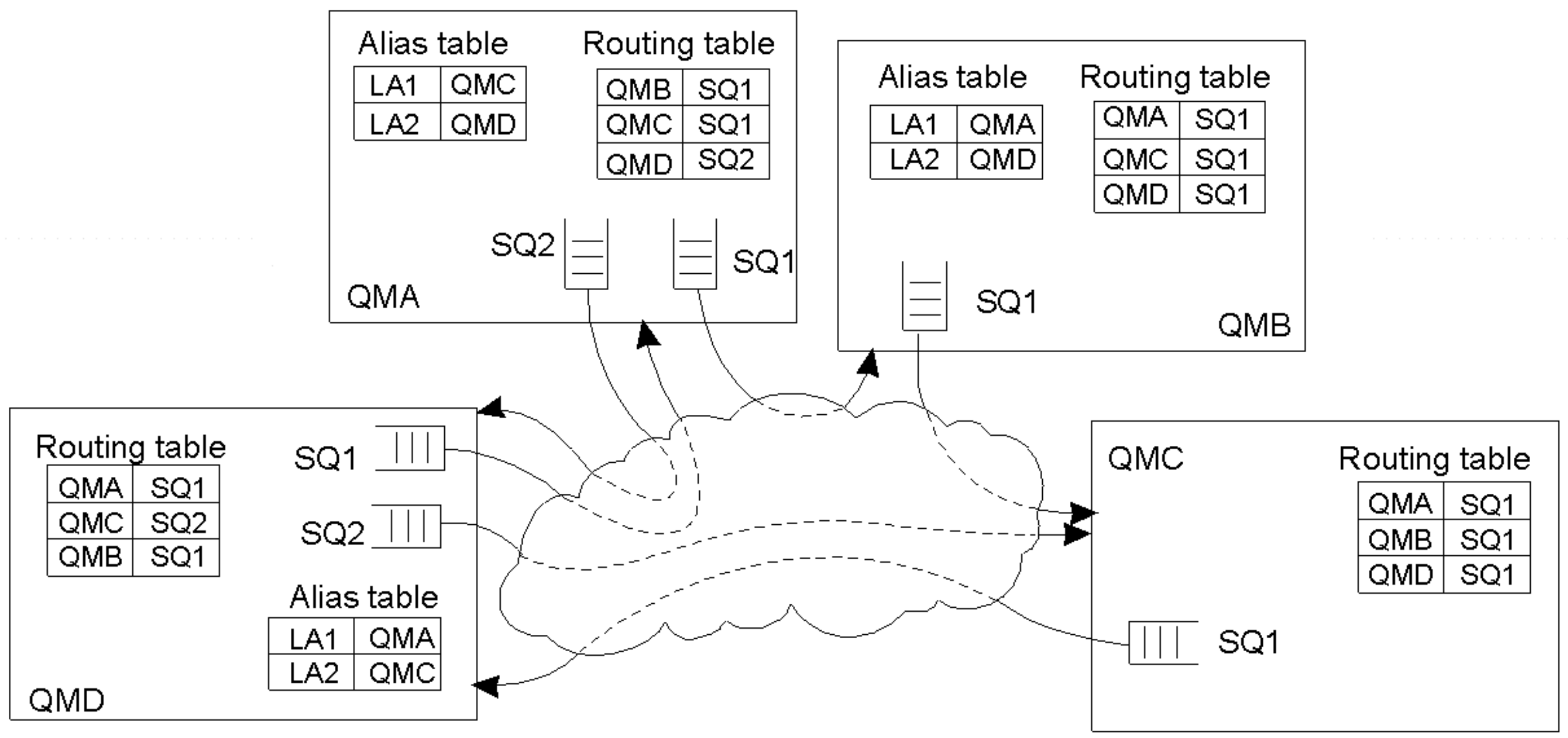
General organization of IBM's MQSeries message-queuing system.

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 (1)



QMA

Alias table

LA1	QMC
LA2	QMD

Routing table

QMB	SQ1
QMC	SQ1
QMD	SQ2

QMB

Alias table

LA1	QMA
LA2	QMD

Routing table

QMA	SQ1
QMC	SQ1
QMD	SQ1

QMD

Routing table

QMA	SQ1
QMC	SQ2
QMB	SQ1

Alias table

LA1	QMA
LA2	QMC

QMC

Routing table

QMA	SQ1
QMB	SQ1
QMD	SQ1

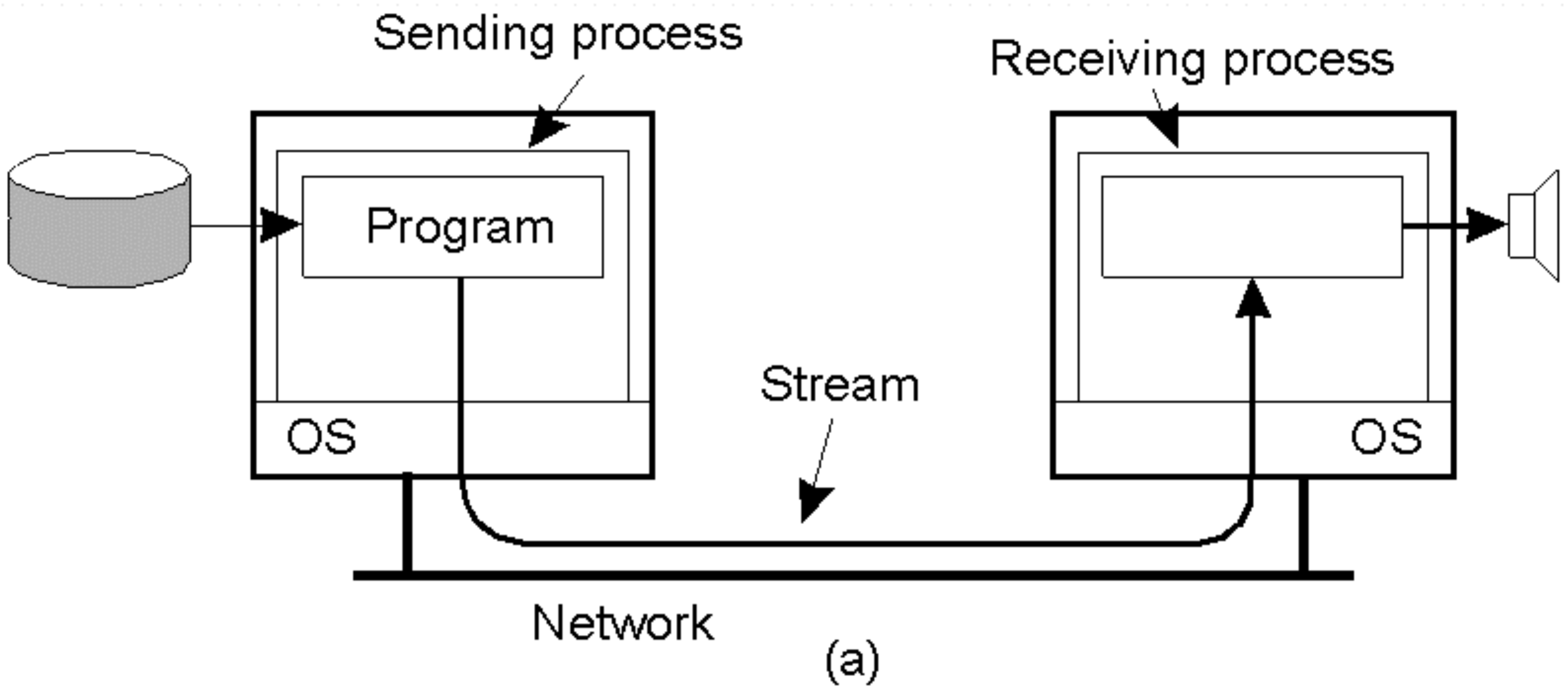
The general organization of an MQSeries queuing network using routing tables and aliases.

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

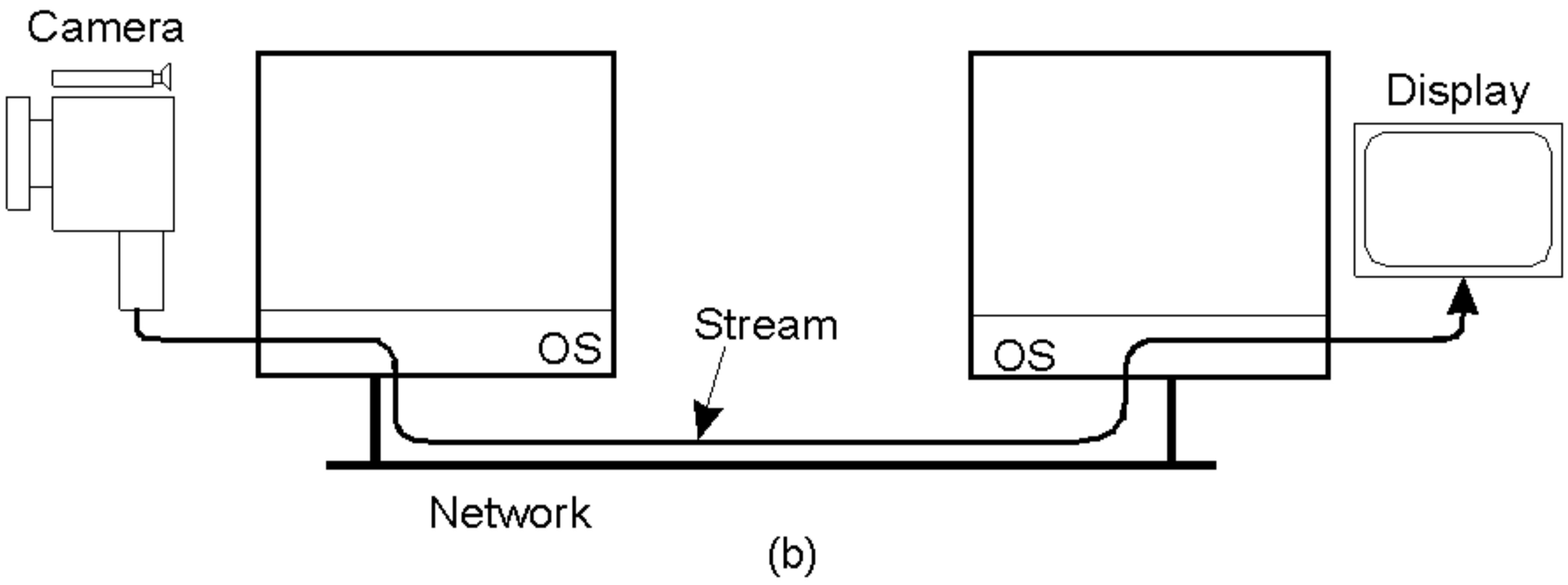
Primitives available in an IBM MQSeries MQI

Data Stream (1)



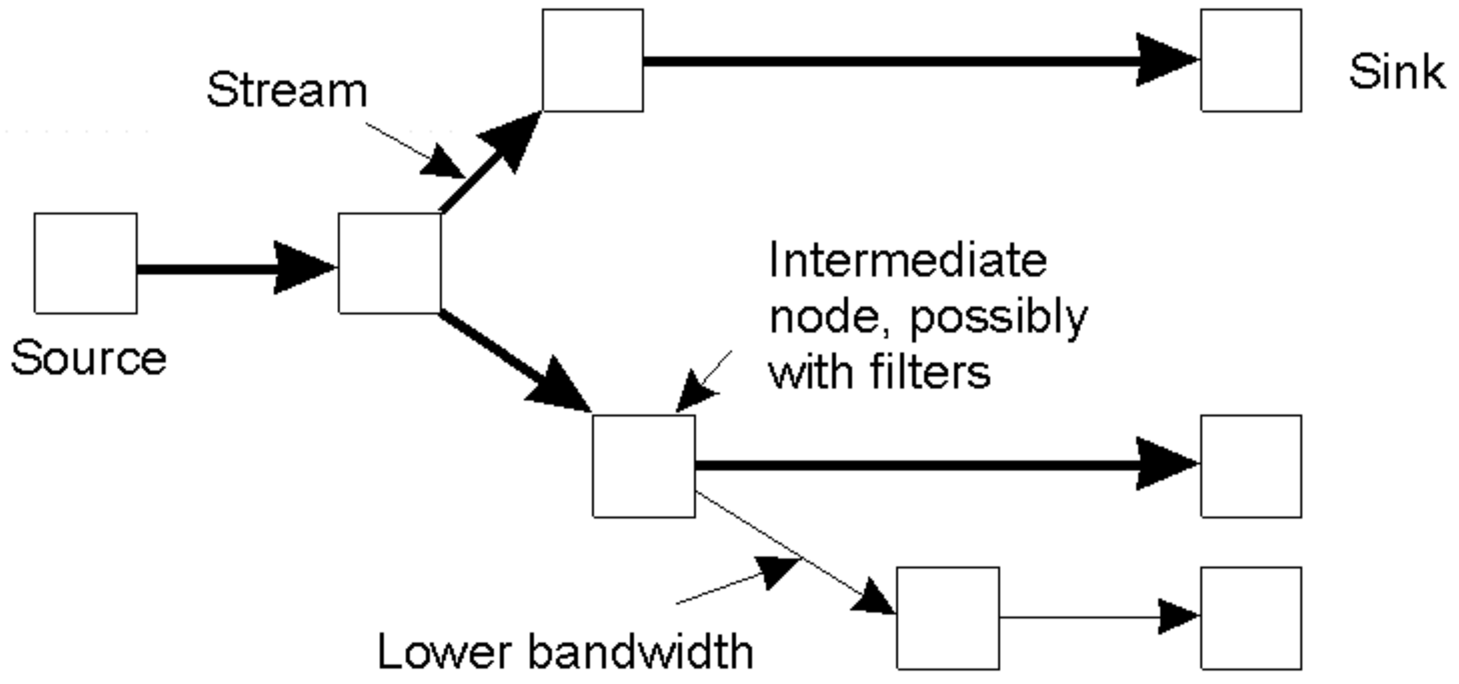
Setting up a stream between two processes across a network.

Data Stream (2)



Setting up a stream directly between two devices.

Data Stream (3)



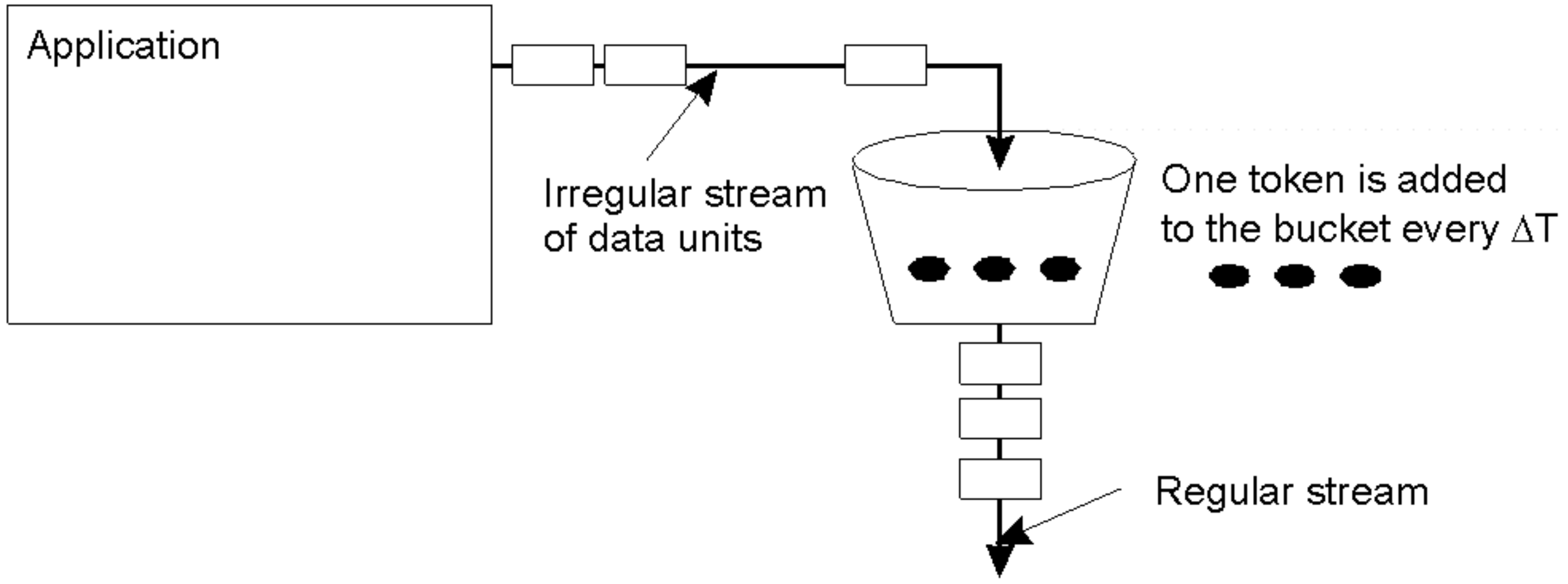
An example of multicasting a stream to several receivers.

Specifying QoS (1)

Characteristics of the Input	Service Required
•maximum data unit size (bytes) •Token bucket rate (bytes/sec) •Toke bucket size (bytes) •Maximum transmission rate (bytes/sec)	•Loss sensitivity (bytes) •Loss interval (μsec) •Burst loss sensitivity (data units) •Minimum delay noticed (μsec) •Maximum delay variation (μsec) •Quality of guarantee

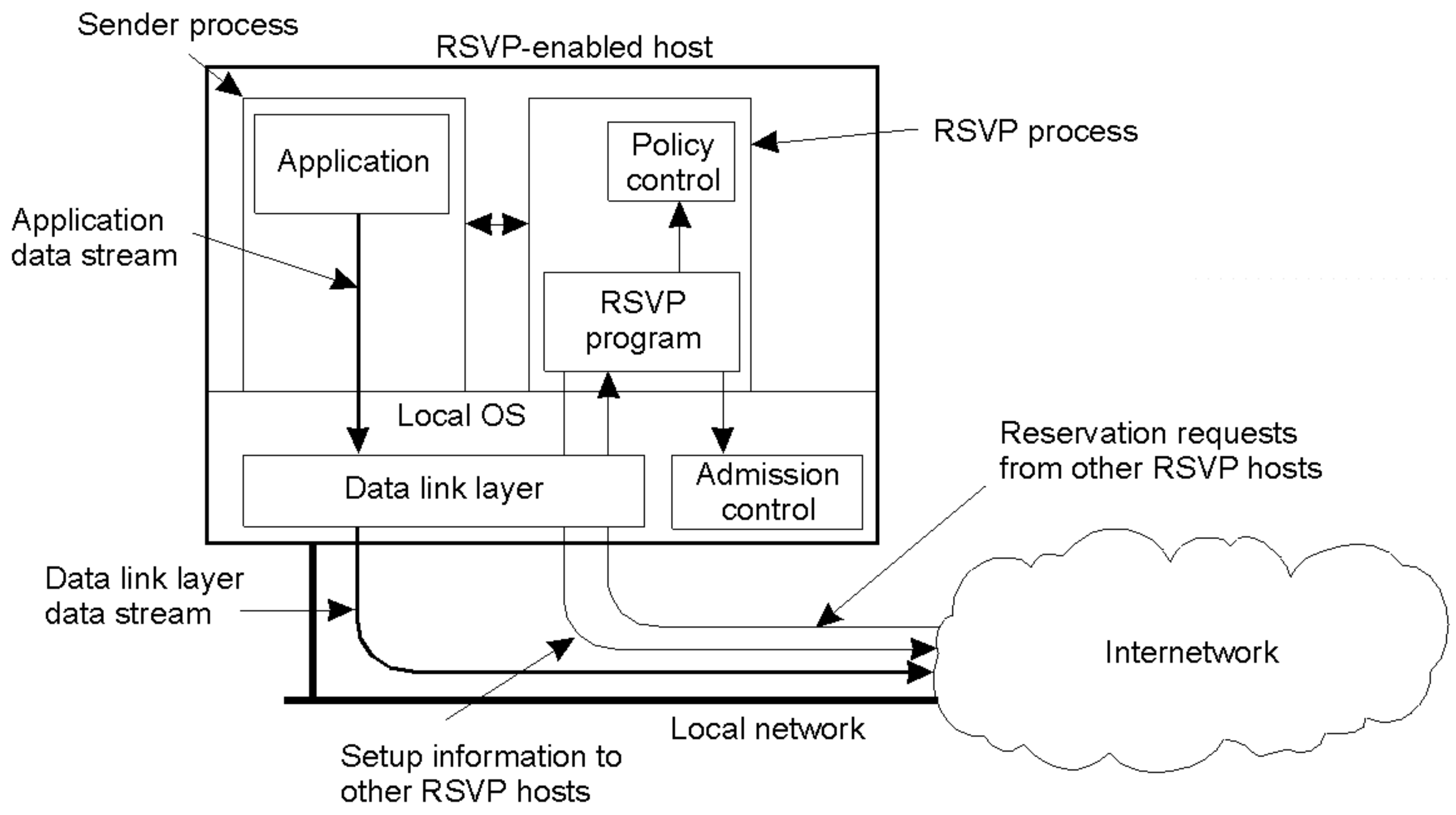
A flow specification.

Specifying QoS (2)


Application
Irregular stream of data units
One token is added to the bucket every ΔT
Regular stream

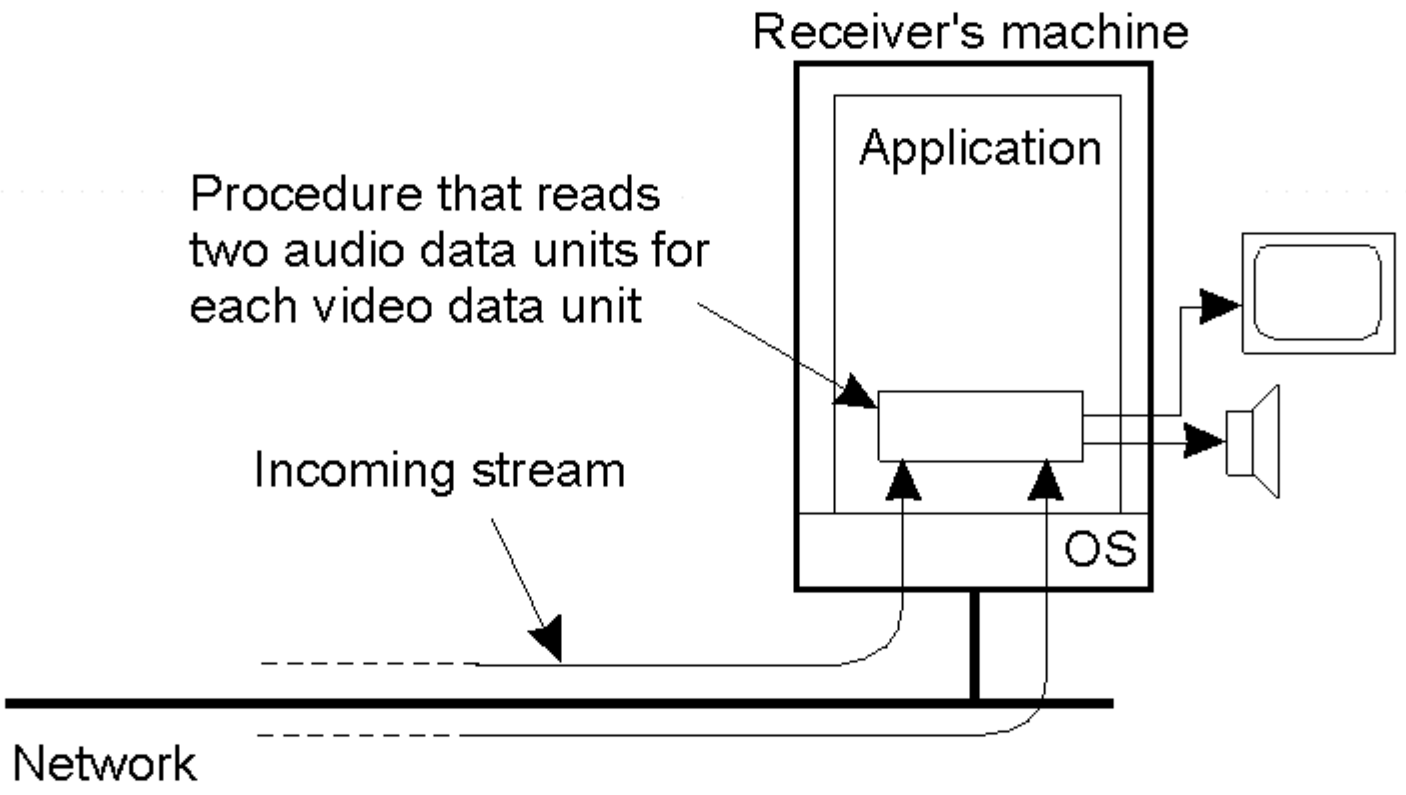
The principle of a token bucket algorithm.

Setting Up a Stream



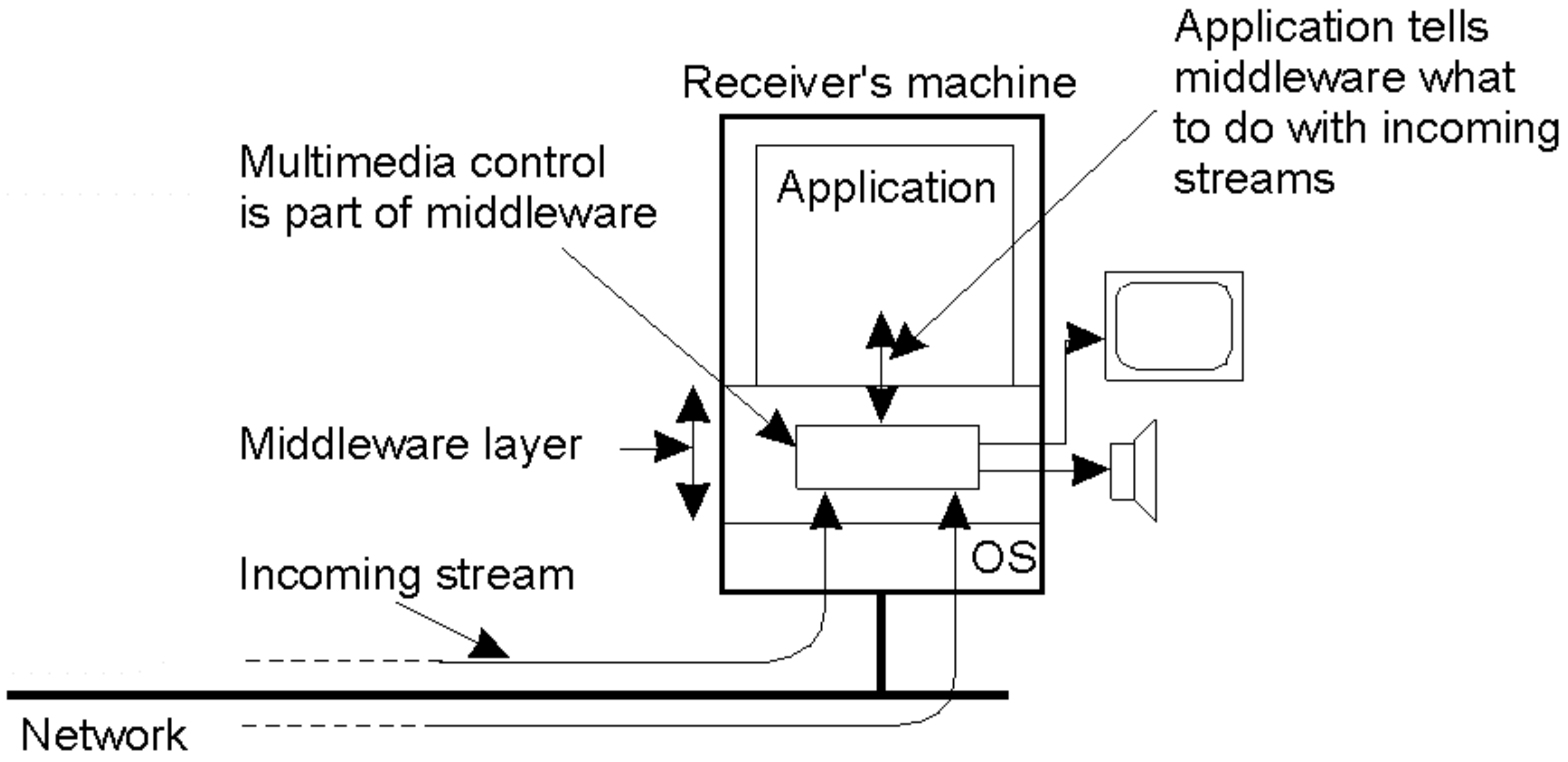
The basic organization of RSVP for resource reservation in a distributed system.

Synchronization Mechanisms (1)


Receiver's machine
Application
Procedure that reads two audio data units for each video data unit
Incoming stream
OS
Network

The principle of explicit synchronization on the level data units.

Synchronization Mechanisms (2)


Receiver's machine
Application tells middleware what to do with incoming streams
Multimedia control is part of middleware
Application
Middleware layer
OS
Incoming stream
Network

The principle of synchronization as supported by high-level interfaces.