Message-Queuing Systems

- **Message-queuing (MQ) systems** (also referred to as **Message–Oriented Middleware**) support asynchronous persistent communication
  - Asynchronous: sender continues execution immediately after submitting its message for transmission
  - Persistent: messages are stored by the MQ system until receiver can accept it
- MQ systems are mainly used for **Enterprise Application Integration (EAI)**, i.e., integration between applications within a given enterprise
- Example middleware: IBM's WebSphere MQ, Microsoft's MSMQ, Oracle's Streams Advanced Queuing

1

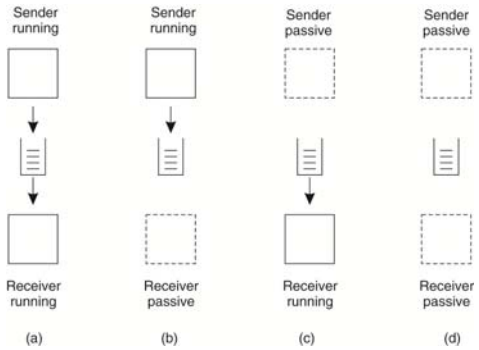
Message-Queuing Model (1)

- Applications communicate by inserting messages in queues, which are provided by the MQ system
- Each application has its own queue to which other applications can send messages
- Each queue has a system-wide unique name; Every message contains the name of the destination queue
- A sender is given the guarantee that its message will be inserted in the recipient's queue
  - No guarantees are given about when or if the message will be read, which is determined by the behavior of the recipient

2

Message-Queuing Model (2)

- Communication is loosely coupled in time: Sender and receiver can execute independently of each other



Four combinations for loosely-coupled communications using queues.

3

Basic Interface Offered to Applications

Primitive	Meaning
Put	Append a message to a specified queue (nonblocking)
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

4

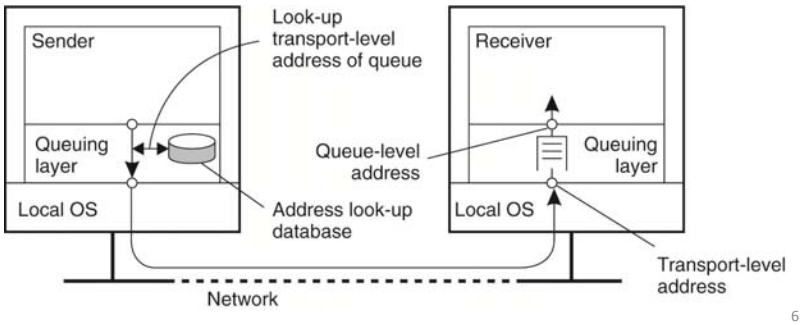
General Architecture of a MQ System (1)

- Applications can only put messages into and read messages from **local queues**
  - A **local queue** is a queue on the same machine or on a machine nearby (e.g., on the same Ethernet) that can be effectively reached through an RPC
- Sender's local queue is called the **source queue**; the receiver's local queue is called the **destination queue**
- The MQ system is responsible for transferring messages from their source queue to their destination queue

5

General Architecture of a MQ System (2)

- Every message contains the name of the destination queue
- A MQ system maintains a mapping of queue names to network locations in order to transfer messages to their destination queue
  - The mapping is maintained in a (possibly distributed) database



6

General Architecture of a MQ System (3)

- Queues are managed by queue managers
  - A queue manager interacts directly with the application that is sending or receiving a message
- One way to map queue names to network locations is to have each queue manager maintain a copy of the queue-to-location mapping
  - A message can be directly sent to the destination queue manager
  - This is unscalable: every queue manager must be updated when queues are added or removed

7

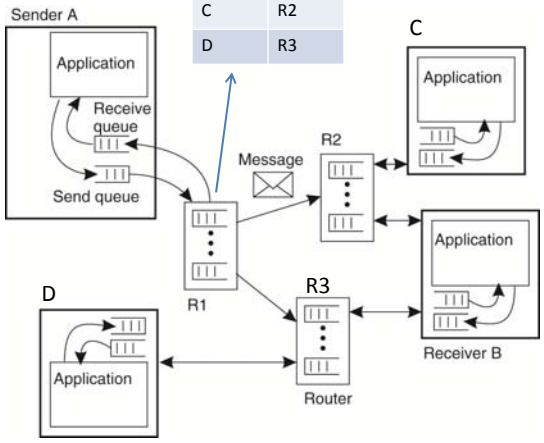
General Architecture of a MQ System (4)

- A scalable solution is to employ a few special queue managers that operate as **routers**, or **relays**
- A message is first transferred from the source queue to the nearest router; the router then forwards the message toward its destination queue
- How does a router know where to forward a message?
  - Each router maintains a routing table that maps destination queue names to next-hop routers
  - Forwarding decision is made by consulting the routing table
- This is scalable
  - Queue managers only need to know where the nearest router is
  - When queues are added or removed, only routing tables stored in routers need to be updated

8

R1's routing table

Queue Name	Next-Hop
B	R2
C	R2
D	R3

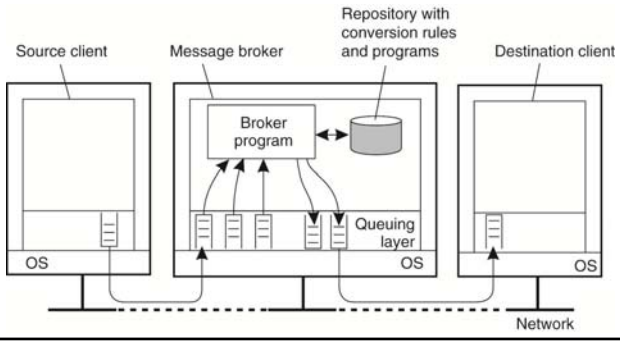


The general organization of a message-queuing system with routers.

9

Message Brokers

- A MQ system is often used to integrate different applications into a single coherent distributed information system
- Different applications may have different message formats; message brokers can be used to handle conversion between different message formats (e.g., converting from one byte order to another, converting from one external data representation to another)
  - A message broker maintains a repository of rules and programs that can transform a message of one type to a message of another type



10

MQ Systems vs Email Systems

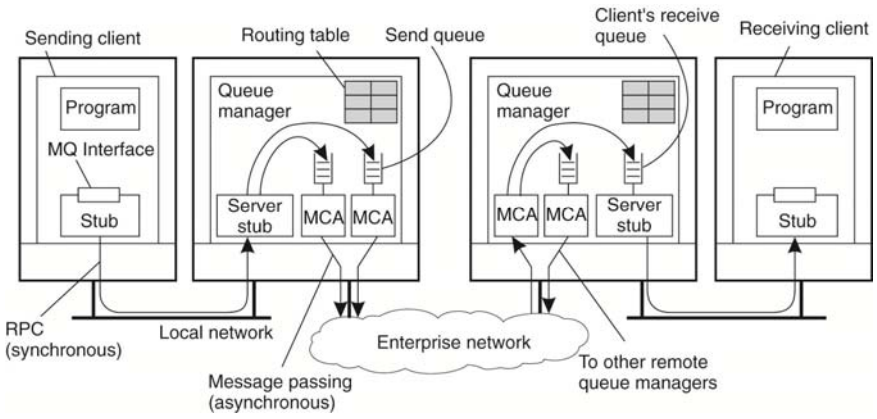
- Both MQ systems and email systems support asynchronous persistent communication
- There are a few differences
  - No routing in email systems: messages are sent directly from the source mail server to the destination mail server
  - MQ systems are more general purpose than email: MQ systems enable communication between processes; email systems support only end users

11

Example: IBM's Websphere MQ

- All queues are managed by queue managers, which handle outgoing and incoming messages
- Queue managers are linked together to form a certain topology and route messages toward their destination queue
- Queue managers and applications can run on the same machine or on separate machines
  - If on separate machines, the application communicates with the queue manager using RPC

12



General organization of IBM's message-queuing system.

13

Message Channels

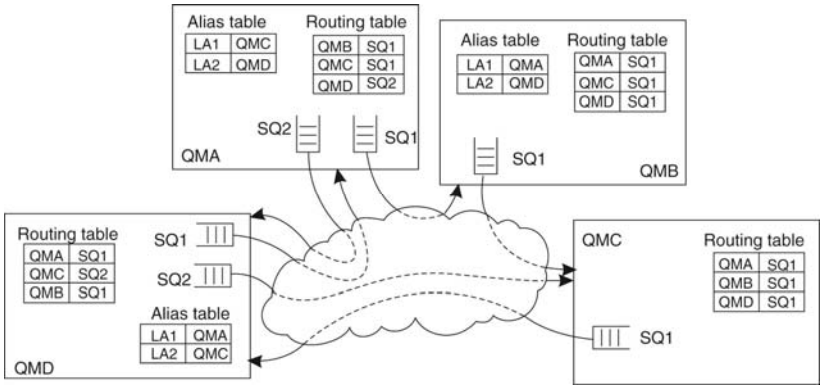
- Queue managers are pairwise connected through **message channels**
  - A message channel is a unidirectional, reliable connection (e.g., a TCP connection) between a sending and a receiving queue manager, through which queued messages are transported
  - Each message channel has one associated send queue that contains the messages to be sent on the channel
- At each endpoint of a channel is a **message channel agent (MCA)**
  - The sending MCA is responsible for wrapping messages in the send queue into transport-level packets and sending them
  - The receiving MCA is responsible for receiving packets, unwrapping them, and storing unwrapped messages into the appropriate queue

14

Message Transfer

- Each message carries a destination address consisting of the name of the destination queue manager and the name of the destination queue
- Each queue manager has a routing table (maintained manually)
  - A table entry is a pair (destQM, sendQ), where destQM is the name of a destination queue manager and sendQ is the name of a local send queue
  - The entry means that a message for destQM should be appended to sendQ
  - Note: Each message channel has exactly one send queue, so sendQ effectively specifies to which queue manager the message is to be forwarded
- **Aliases** can be used to allow applications to use the same logical name for a queue even if the queue manager of that queue changes
  - An alias defined within a queue manager M1 is another name for a queue manager M2, which is only available to applications interfacing to M1
  - Changing the name of a queue manager requires that we change its alias table entry and routing table entry in all queue managers. However, applications are not affected.

15



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

16