CISC 434: Communication - III

- Communication
  - Layered Protocols
  - Types of Communications
  - Remote Procedure Call (RPC)
  - Message Oriented Communication
  - Stream-Oriented Communication
  - Multicast Communication
Message Oriented Persistent Communication

- Message Queuing Model / Message-Oriented Middleware (MOM)
  - Intermediate storage capacity for messages – do not require either the sender or receiver to be active during message transmission
  - Intended to support message transfers that are allowed to take minutes instead of sec/ms (contrary to Berkley Sockets and MPI)
The sender is usually given only the guarantees that its message will eventually be inserted in the receiver’s queue.

No guarantees are made about when, or even whether the message will actually be read.

Four combinations:
**Message-Queuing Model – contd.**

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

<table>
<thead>
<tr>
<th>Primitive</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Put</td>
<td>Append a message to a specified queue</td>
</tr>
<tr>
<td>Get</td>
<td>Block until the specified queue is nonempty, and remove the first message</td>
</tr>
<tr>
<td>Poll</td>
<td>Check a specified queue for messages, and remove the first (Non blocking)</td>
</tr>
<tr>
<td>Notify</td>
<td>Install a handler to be called when a message is put into the specified queue</td>
</tr>
</tbody>
</table>
Message-Queuing System

- Transfers messages from a source to a destination

- Main components
  - Source queue – A message can be put only into this local queue, and the message has a destination queue address
  - Destination queue – A message can be read only from here
  - Queue names – A database of queue names – maintained by queuing system
  - Queue managers – interact directly with the application that is sending or receiving a message
  - Special queue managers – Relays/Routers – forward incoming messages to other queue manager
Message Brokers

- A special node in a message-queuing system that handles the conversation
- Main task: convert incoming messages to a format that can be understood by the receiver application
A Message-Queuing System with Routers
An Example Message Queuing System

General Organization of IBM’s WebSphere Message Queuing System – WebSphere MQ

![Diagram of message queuing system]

- Sending client
- Routing table
- Send queue
- Client’s receive queue
- Receiving client

- Program
- MQ Interface
- Stub

- Queue manager
- Server stub
- MCA

- Enterprise network
- Message passing (asynchronous)

- Local network
- RPC (synchronous)

- To other remote queue managers
WebSphere MQ

- **Queue managers**
  - Removes messages from its send queues and forwards those to other queue managers
  - Picks up incoming messages from the network and stores each message in the appropriate input queue

- **Message Channels**
  - A unidirectional connection between a sending and receiving manager

- **Message Channel Agent (MCA)**
  - Each end of a message channel is controlled by an MCA
  - A sending MCA checks send queues for a message, wraps it into a transport level packet, and sends it along the connection to its corresponding receiving MCA
  - A receiving MCA listens for an incoming packet, unwraps and stores it in the appropriate queue
Data Stream

- Data Stream – a sequence of data units
  - Simple stream – consists of only a sequence of data
  - Complex stream – consists of several related simple streams called sub-streams – e.g., transmitting a movie

- Types of Media
  - Continuous media – the temporal relationships between different data items are fundamental to correctly interpreting the data – motion pictures require that successive images display at a uniform spacing \( T \) in time
  - Discrete media – temporal relationships of between data items are not fundamental to correctly interpreting the data – text, still images, executable files
Data Stream – Transmission Modes

- **Asynchronous**
  - Data items are in a stream are transmitted one after the other, but there are no further timing constraints on when transmission of items should take place.
  - Specifically true for discrete data streams – e.g., file transfer.

- **Synchronous**
  - Maximum end-to-end delay defined for each unit in a data stream.
  - A data unit may be transferred much faster than the maximum tolerated delay.
  - E.g., a sensor sampling temperature at a certain rate and send it via network.

- **Isochronous**
  - Data units are constrained by maximum and minimum end-to-end delay – distributed multimedia system (audio and video).
Data Stream

A general architecture for streaming stored multimedia data over a network.
Data Stream Related Issues

Compression
- To reduce required storage and network capacity

Quality of Service Control
- The required bit rate at which data should be transported.
- The maximum delay until a session has been set up.
- The maximum end-to-end delay.
- The maximum delay variance, or jitter.
- The maximum round-trip delay.
Streams and Quality of Service

- Stream Synchronization – takes place at the level of data units of which a stream is up
  - Synchronization between a discrete and a continuous data stream – slide show with audio
  - Synchronization between continuous data streams – playing movie
Multicast Communication

- Information from one sender to multiple receivers
- Application-Level Multicasting
  - Nodes organize in a overlay network
  - Achieved by setting up a tree from the sender to the receivers
  - The trees can be generated dynamically
- Gossip-Based Data Dissemination
  - Based on epidemic protocols – how diseases spread among people
  - No central dissemination coordination – uses only local information
Summary

- **Message Oriented Persistent Communication**
  - Message Queuing Model
  - Message Queuing System Architecture
  - Message Brokers

- **Stream-Oriented Communication**
  - Quality of Service
  - Synchronization

- **Multicast Communication**