

**Network Communication Protocols**

**(NCP)**

 **24CSCN03I**

Presented to TA. Seif

|  |  |
| --- | --- |
| Seif Ahmed Amin | 230598 |
| Seif Allah Mohamed | 229879 |
| Abdelrahman Sherif | 237318 |

This Java messaging system provides a publish-subscribe model over TCP sockets. There is a Broker server in the middle accepting incoming connections and supporting two types of clients: Publishers and Subscribers. Publishers connect to the broker to send messages using direct routing (to a named queue) or fanout routing (broadcast to all queues). Subscribers bind themselves to the broker, provide the name of the queue for which they want to subscribe, and receive all messages present on the queue. All message queues are controlled in a centralized manner by the Broker through an Exchange object, which holds all active queues and their messages. Each client is run in its individual thread to facilitate concurrent communication.

1. **Broker**

Role:

* Master server that hosts all message queues and client connections.

1. Primary Responsibilities:

* Handles socket connections on port 3000.
* Services every client (publisher or subscriber) in a separate thread through Runnable.
* Shares an Exchange object to handle queues.

1. For Publishers:

* Reads routing type: direct or fanout.

1. For Subscribers

* Subscriber sends a queue (routing key).
* If the queue is available, sends all queued messages and closes connection.

1. Direct:

* Publisher specifies a queue name.
* If it finds the queue, it accepts and stores messages until "exit".

1. Fanout:

* Every message from the publisher is forwarded to every queue.

1. **Publisher**

Role:

* Publishes messages to the broker.

1. Key Responsibilities:

* Connects to the broker on port 3000.
* Identifies itself as a "Publisher".

1. Chooses routing type:

* Direct:
* Chooses a queue to send messages to.
* Sends messages in a loop until "exit".

1. Fanout:

* Sends messages to all queues via the broker.
* Sends messages in a loop until "exit".
* Graciously handles queue not found and user exits.

1. **Subscriber**

Role:

* Continuously listens to a specific queue for messages.

1. Primary Responsibilities:

* Logs in on port 3000 to the broker.
* Identifies itself as a "Subscriber".
* Continuously asks user for a routing key (queue name) until it receives one.
* Upon subscribing, reads and prints all available messages it can receive from that queue.
* Terminates after receiving all messages (design is pull-once, not continuous).