Assignment 01: Chat Application - Topology

Sam Gilb
Chloe Bates
Colton Spector

Approach 1: The Ring

- Logical View: In this model, each node will be connected to two other nodes, one on each side forming a ring of connections. All the data is transmitted through the nodes in one direction within the ring.
- Transactional View**:**

  Joining a chat

  For the first node N==1

  1. Will be returned “You are talking to yourself; no one in network”

  For a node joining a network N>=1

  1. Will join the tail of the network every time, connecting back to the head, the longest existing node in the network

  Leaving a chat

  For a node leaving a network != the head

  1. The leaving nodes connections will establish the new connection, disconnecting the leaving node from the ring.

  For node leaving the network == head

  1. The head node passes its tail link to its next node

  Sending and Receiving messages

  (Sending)

  1. The Node sending the message will send the message clockwise to every node until the message has been received to the sending node, where it is printed for the first time to the screen.

  (Receiving)

  1. The messages will come clockwise from the sending node until the sending node has received and then prints their own message

Approach 2: Distributed Star (Linked List)

- Logical View: Nodes connected together in a linear fashion with multiple central connection points. Each node will link to the next node and each node will know what node is before it (Doubly linked)
- Transactional View**:**

  Joining a chat

  1. Start at the head node and go until next link = null
     a. If there is no head node become the head node

  Leaving a chat

  1. Start at head node and go until next link = the node that wants to leave

     Three Exit Cases

     a. Head Node

        -Temp store the next node

-Delete the head node
-Set the next node to head node

b. Last Node
-Remove node
-Set reference of previous node to null

c. Middle Node
-Temp store address of node to be deleted
-Change the references for the next node and the previous node to point to each other
-Delete node

Sending and Receiving messages

(Sending)

For the generating node

1. Send the message to the previous and next node if applicable

For each node upon receiving a message

1. Send message to node which hasn't sent it the message
2. Center distributes to all nodes connected

(Receiving)

For regular node

1. Receive message from other node
2. Determine if the message needs to be propagated and do so if required

Compare & Contrast:

When comparing and contrasting the two network topologies, we considered the metrics: overall structure, failure, data transmission, and cost.

<u>Ring Topology</u>

Overall Structure

- Managing the connection between nodes is simple, since it is without a central node.
- Easily visualized, assisting in an easier implementation.
- Being able to change the head of the ring when the head leaves the chat.

Failure

- Every node is a failure point due to sending data from node to node.
- The P2P structure allows faults to be easily identified and isolated.

Data Transmission

- Able to span farther because the signal is regenerated each time a data token moves past a node.

Cost

- Low cost due to not having a hub, unlike the distributed star topology.

Distributed Star (Linked List) Topology

Overall Structure

- Simple data structure allows for easy implementation.
- Easy to connect new nodes without affecting the rest of the network, unlike the Ring Network.
- Easy to maintain since each node has their own connection to one another.

Failure

- The star's hub is the only failure point, so the failure of one node does not impact the entire network, like the Ring Network.

Data Transmission

- Generally performs at high speed with transferring data.
- Works well with heavy traffic and a large network.
- The more nodes in the list the longer it takes for the message to propagate.
- No centralized manager of messages.

Cost

- High. Using a hub increases the cost of the network.

Conclusion:

The network topology that best suits our Distributed Chat Application is the Distributed Star (Linked List) Topology. We have outlined why we chose this network architecture with respect to each of the above metrics:

- This network topology's *overall structure* is better suited for our needs because it is easily adaptable when connecting/disconnecting new nodes into the network.
- Having a single *failure* point, as opposed to every node, is more optimal with a Chat system that could possibly have many nodes.
- *Data transmissions* are done at a good speed, even at high volumes.
- Although, the *cost* is higher than the Ring's, we found that for our Chat Application, the other benefits of the Distributed Star outweigh the high cost.

However, we reserve the right to pivot if we think it will provide a better end product.