Report

The algorithm allows a set of networked peers to establish their availability by exchanging heartbeat messages.

Algorithm Description

1. Initialization

a. The program starts by reading a list of peer hostnames from a configuration file.
b. It excludes its own hostname from this list, determining the number of peers in the cluster.

2. Socket Setup

a. An inbound UDP socket is created and bound to port 8080 for receiving heartbeats.
b. n-1 outbound UDP sockets are created, one for each peer, for sending heartbeats.

3. Main Loop

The main loop continues until heartbeats have been received from all peers. The process is as follows:

1. Use select() to efficiently monitor all sockets for read/write readiness.
2. For each socket that is ready:
    - a. If it's the inbound socket:
        - i. Receive the packet using *recvfrom()*.
        - ii. Identify the sender using the packet's source address.
        - iii. If the message is a valid heartbeat and it's the first one from this peer:
            1. Increment the count of received heartbeats.
            2. Mark this peer as having sent a heartbeat.
    - b. If it's an outbound socket:
        - i. Send a heartbeat message to the corresponding peer using *sendto()*.

4. Completion

Once heartbeats have been received from all peers, exit the main loop.

a. Clean up resources (free memory, close sockets).
b. Print "ready" to indicate successful completion.