COMP3331
CDHT Report

Charlie Bradford z5114682

May 18, 2018

1 Program Design

The program consists of four concurrent threads.

1.1 Input Handler

This thread handles input from the user and can be used to request files, manual pings etc.

1.2 Successor Checker

This thread sends three pings spaced by one second to the peers successor every 30 seconds. If the successor doesn't respond to any of the three pings the peer assummes the successor is dead and intiates the process of fixing all nearby peers' successors and predecessors.

1.3 TCP Server

Handles and automatically responds to UDP messages. Responsibilities are:

- Forwarding or replying to file request messages when appropriate
- Alerting the user when a file has been found
- Fixing the successors of its predecessor's two predecessors in the case of ungraceful peer death
- Fixing its own successors when alerted of peer death

1.4 UDP Server

Handles and automatically responds to UDP messages. Responsibilities are:

- Responding to ping requests
- Establishing predecessors when ping or establish messages are recieved
- Letting Successor Checker know when a ping is responded to

1.5 Design Tradeoffs

Ideally a peer would check its successor more often, to reduce lapses in communication. However this could overload the computer and significantly interfere with user input. Peer checking could also be done without output to prevent interference with user input, however it is important to demonstrate successful ping requests and responses.

1.6 Possible Extensions

It could be interesting to distribute actual files between the peers and implement a file transfer system. A peer could request a file and if the request went around the CDHT without response then the peer would assume that the file is unavailable, otherwise the the owner of the file could transfer it the requesting peer, who would then have it available for other peers to request.

Another idea could be to have the main CDHT but also have several 'sub-peergroups' within. All members could request files but perhaps peers have confidentials files, only accessible to peers that share in a subgroup. This would be good in an office where file sharing is often required but certain files may be priviledged to a group of executives, or a group that has signed contracts with another company.

2 TCP Messages

2.1 File Request

Used to request a file.
Format: “rcq <filename><origin_of_request>0”

Announced by peers when forwarding.

2.2 File Response

A response to a file request.
Format: “rrp <filename><origin_of_request><location_of_file>”

Sent directly to the origin of the request.

2.3 Peer Departure

A message sent by a departing peer or by a dead peers successor and predeccsor.
Format: “dep <origin><new_successor><new_second_successor>”

A peer sends this message to it’s two predecessors when it quits. Alternatively a peer sends this message to its second predecessor when its second predecessor informs it that its first predecessor is dead via an Ungraceful Peer Departure message.

2.4 Ungraceful Peer Departure

Sent by a peer to its second successor when its first successor is determined to be dead.
Format: “ded <origin><new_second_predecessor>0000”

3 UDP Messages

3.1 Ping Request

Generic ping. Sent to successor three times every 30 seconds automatically or to any peer in the CDHT manually with user input.
Format: “png <origin_of_request><relationship>”

3.2 Ping Response

Response to ping, sent whenever a ping request is received.
Format: “prp <origin_of_response><relationship>”

3.3 Establish Relationship

Used to establish a predecessor relationship.
Format: “est <origin_of_message><relationship>”

4 Demonstration

Youtube Link[1]

Unfortunately I scrolled past the udp server really quickly but if you pause at 2:58 you can see the code that receives ping messages and sends the ping responses from a udp socket.

5 Other Information

This project was written in Python3.

There were some issues with the supplied set up code, including some strange quotation marks, so this was the setup script I used.

[1]https://youtube.com/ORXNtc0MVlM

```
xterm -hold -title "Peer 1" -e "python3 cdht.py 1 3 4" &
xterm -hold -title "Peer 3" -e "python3 cdht.py  3 4 5" &
xterm -hold -title "Peer 4" -e "python3 cdht.py  4 5 8" &
xterm -hold -title "Peer 5" -e "python3 cdht.py  5 8 10" &
xterm -hold -title "Peer 8" -e "python3 cdht.py  8 10 12" &
xterm -hold -title "Peer 10" -e "python3 cdht.py  10 12 15" &
xterm -hold -title "Peer 12" -e "python3 cdht.py  12 15 1" &
xterm -hold -title "Peer 15" -e "python3 cdht.py  15 1 3" &
```