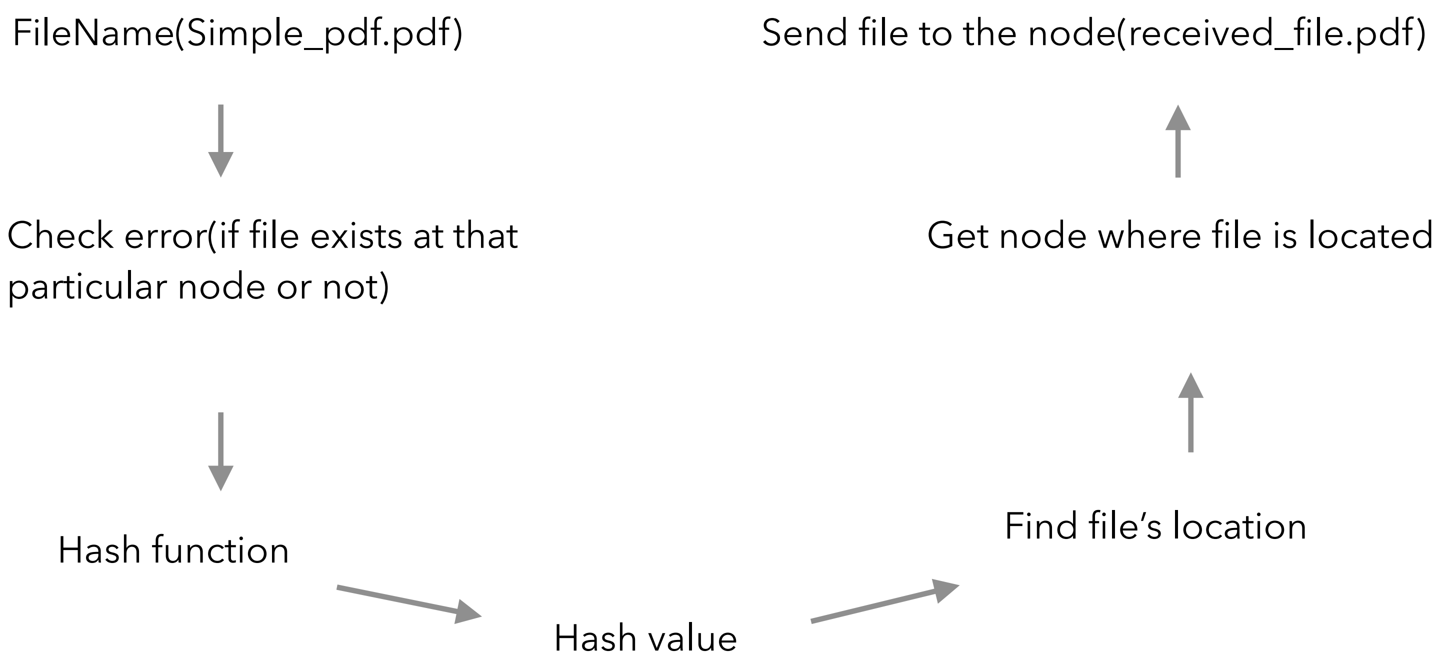
Assignment Report

To implement the program I decided on using Java and the main class is **cdht** with a bunch of functions fetching details from the command line argument provided.A socket was created to send and receive packets from/ to the nodes created.I decided on using the format with Actions so that it's easier for me to perform the next task using them as arguments.For example: for file transfer I decided to use event actions like EVENT_SND etc used as SND(Action);2(Node);1(seq no.);3(ack no.) and then parsed them to get the values.I used the same format to store the information about the Successor1, Successor2, Predecessor1, Predecessor2 etc. and used them where needed.The ping interval was chosen to be 5 seconds.
Some classes were extending thread as we were asked to run the program with more than one server and threading is a facility to allow multiple activities to coexist within a single process. The message formats used in the implementation are like"Please provide sufficient arguments" , Please wait, Looking for the successor and predecessor... and please try request the file again" and "You have already got the file, no need to send" and "Invaild File Name, Please try again …." .The process for requesting and sending file was as follows,


FileName(Simple_pdf.pdf)
Check error(if file exists at that particular node or not)
Hash function
Hash value
Find file's location
Get node where file is located
Send file to the node(received_file.pdf)

There were a couple of packets that were dropped during the transfer of file which is logged in the log file.There were also some ping request error(very rare) while pinging a particular peer. For killing the peer I decided to keep a counter rather than including sequence number and timeout interval.A peer was considered to be not alive if it did not respond after 4 requests(wait time 20 seconds) and then the predecessors and successors changes accordingly. In order to send the file, the current file used is 2012.pdf, if any other file need to be sent, the file name in the code need to be changed from 2012.pdf to the file to send.To run the code, compile with javac cdht.java

Work Breakdown

Date	Implementation
16/03/2019	Implemented the initialisation stage, initialised the sockets and the process of sending and receiving messages. Problem: The program was producing the exception error. Reason and Solution: repeated initialisation of the socket which was later resolved.The program was producing the output but the synchronisation with the nodes was not proper e.g. Ping request was received from Peer 1 and 4 was also Peer 3 and 4 where it should be Peer 4 and Peer 5.
21/03/2019	Completed the initialisation and the Ping Successor stages.Started implementing the Requesting file stage.Completed the filename restrictions, hash function and file location. Problem : The xterm console was not taking the input from keyboard while trying "request X". Reason and Solution : Didn't create a server and client TCP connection.Solved the problem by creating the sockets for each.
23/03/2019	Implemented the file location code and located the file and printing appropriate messages when and where the file was found. Problem : The receiving node was not able to receive the request sent by the sending node. Reason and Solution : The nodes were not connected as a ring and hence the successors and predecessors were not set which was later on fixed.

Date	Implementation
28/03/2019	Debugging.
4/04/2019	Started implementing sending the file. Started with breaking the file in different chunks(according to MSS).Introduced the concept of ACK and Sequence number and ACK was increased when the file chunk was received by the receiving peer, if not the file was recreated(pdf to bytes array). Needs to be implemented: Receiving the file chunks.
7/04/2019	Completed sending the file from the senders side. The receiving side was able to receive the chunks. The only thing left to implement is to transform the chunks to pdf file again. Problem:The conversion of bytes->pdf was bit of a hustle. Needs to be implemented: Collecting the chunks and forming a pdf and then moving on to making the log file, peer departure and killing the peer.
11/04/2019	Completed sending and receiving file.Created a duplicate pdf with same size & created the log file. Started working on the Peer departure and Peer kill. Problem and Solution: The duplicate file size was coming out to be more than the original one due to which the duplicate file was not accessible.Debugged, found error in loop and fixed it. All done!!!

I tried to make the code as efficient as possible, but a particular improvement could've been having more printed messages/comments for a more detailed and easily understandable code.

Demo Link

https://www.youtube.com/watch?v=MtKoVlo8x_E