Computer Networks (CNT 5106C)

Spring 2019

Peer-to -Peer Network Application

Submitted by:

Kunwardeep Singh 2421-3955

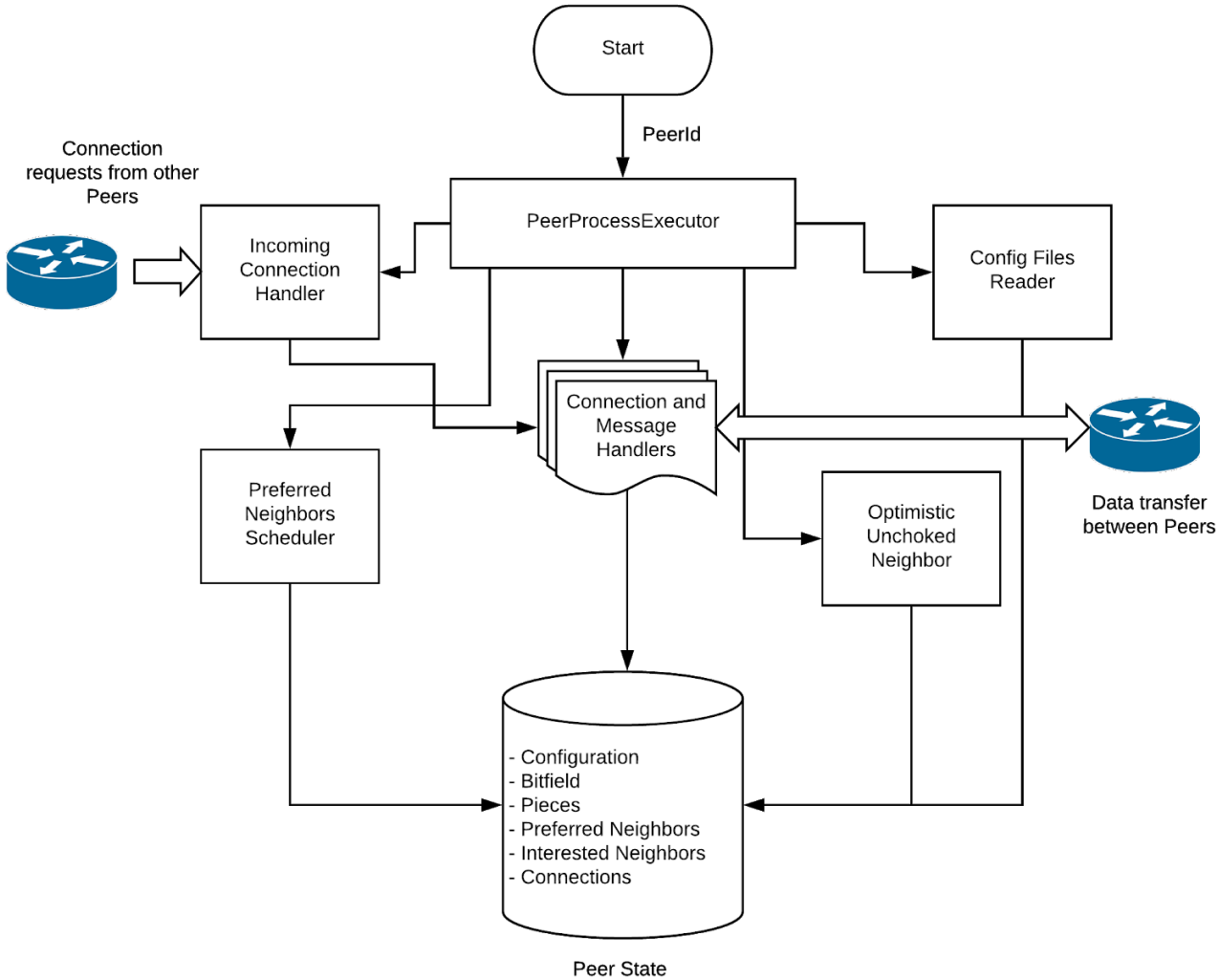
Nimish Kochhar 6139-4423

Abhijeet Kumar 1404-4371

Project Objective:
Create a P2P file sharing software similar to BitTorrent.

Working Environment:
Unix
To compile:
make
Start Peers remotely:
bash start_remote.sh or java StartRemotePeers
Start Peers locally:
bash start.sh
Stop Peers remotely:
bash stop_remote.sh

Application Flow for a Peer:



Program Structure

This application program has been divided into 6 modules with 24 java classes, in total. The details for these modules and classes are given below

1. Java Data Transfer objects (DTOs) - POJO classes for transferring information encapsulated in an object. The details are as under:

BitTorrentState.java

Variables	**Type**	**Description**
numberOfPieces	Integer	Number of pieces (obtained from Common.cfg)
numberOfPreferredNeighbors	Integer	Number of preferred neighbors
unchokingInterval	Integer	Unchoking Interval (Sec)
optimisticUnchokingInterval	Integer	Optimistic Unchoking Interval (Sec)
fileName	String	Name of the file to be shared
fileSize	Long	Size of the file to be shared
pieceSize	Integer	Size of each piece
peers	ConcurrentHashMap<String, PeerState>	Hash map between PeerID and Peer state
Method	**Return Type**	**Description**
getPeerState(String id)	PeerState	Returns the peer state
getPeers()	Map<String, PeerState>	Returns the peer state and peerID map
numberOfPeers()	Integer	Returns the Number of peers
getNumberOfPieces()	Integer	Returns the number of pieces
setNumberOfPieces(int numberOfPieces)	Void	Sets the number of pieces
getNumberOfPreferredNeighbors()	Integer	Returns the number of preferred neighbors
setNumberOfPreferredNeighbors(int numberOfPreferredNeighbors)	Void	Sets the number of preferred neighbors
getUnchokingInterval()	Integer	Returns the unhooking Interval
setUnchokingInterval(int unchokingInterval)	Void	Sets the unhooking interval

getOptimisticUnchokingInterval()	Integer	Returns the optimistic unhooking Interval
setOptimisticUnchokingInterval(int optimisticUnchokingInterval)	Void	Sets the optimistic unchoking interval
getFileName()	String	Returns the filename
setFileName(String fileName)	Void	Sets the file name
getFileSize()	Long	Returns the file size
setFileSize(long fileSize)	Void	Sets the file size
getPieceSize()	Integer	Returns the piece size
setPieceSize(int pieceSize)	Void	Sets the piece size
calculateAndSetNumberOfPieces()	Void	Calculates and sets the number of pieces

PeerState.java

Class variables	**Type**	**Description**
sequenceId	Integer	Sequence ID No.
peerId	String	Peer ID as string
hostName	String	IP address of the host
port	Integer	Port No.
hasSharedFile	Boolean	Has shared file or not(1/0)
fileReceived	Boolean	File received (False/True)
bitField	Bitset	Bit sequence from Bitset Class
fileSplitMap	ConcurrentHashMap<Integer, byte[]>	Concurrent Hashmap for individual files
preferredNeighbours	ConcurrentHashMap<String, String>	Concurrent Hashmap for preferred neighbors
chokedNeighbours	ConcurrentHashMap<String, String>	Concurrent Hashmap for choked neighbors
interestedNeighbours	Map<String, String>	Hashmap for interested neighbors
connections	ConcurrentHashMap<String, PeerConnectionHandler>	Concurrent Hashmap for connections
queue	BlockingQueue<Message>	Blocking queue for message
dataRate	double	Data rate initialized from 0

serverSocket	ServerSocket	Server Socket
downloadComplete	AtomicBoolean	Atomic Boolean initialized from False

Method	Return Type	Description
getDownloadComplete()	Boolean	Returns the boolean value for complete download
setDownloadComplete(boolean value)	Void	Sets the boolean value for complete download
getConnections()	ConcurrentHashMap<String, PeerConnectionHandler>	Returns the concurrent hash map
getServerSocket()	ServerSocket	Returns the server socket
setServerSocket(ServerSocket serverSocket)	void	Sets the server socket
setTimer1(Timer timer1)	void	Setting the timer
stopScheduledTasks()	void	Stopping the scheduled tasks
setInterestedNeighbours(Map<String, String> interestedNeighbours)	void	Assigning the interested neighbors
getDataRate()	double	Returns the data download rate
setDataRate(double dataRate)	void	Sets the data download rate
getQueue()	BlockingQueue<Message>	Returns the blocking queue
putFileSplitMap(int index, byte[] piece)	synchronized void	Sets the properties of filesplitmap
getInterestedNeighbours()	Map<String, String>	Returns the map for interested neighbors
preferredNeighboursCount()	int	Returns the size of pref. Neighbors
removeInterestedNeighbours(String peerId)	void	Removes from the list of interested neighbors
putInterestedNeighbours(String peerId)	void	Inserts in the list of interested neighbors
putPreferredNeighbours(String peerId)	void	Inserts in the list of Pref. Neighbors
setBitField(BitSet bitField)	void	Sets the values for the bitfield
getFileSplitMap()	ConcurrentHashMap<Integer, byte[]>	Returns the file split map

setFileSplitMap(ConcurrentHashMap<Integer, byte[]> fileSplitMap)	void	Assigns the values in the file splitmap
setHasSharedFile(boolean hasSharedFile)	void	Sets the boolean for Has shared file
isFileReceived()	Boolean	Returns the Boolean for file received or not
setFileReceived(boolean fileReceived)	void	Sets the boolean for file received or not
getSequenceId()	Int	Returns the sequence ID
setSequenceId(int sequenceId)	void	Assigns the sequenceID
toString()	String	Concatenates the variables and converts into string

Class Variables	**Type**
run()	Void
peerSocket	Socket
peerState	PeerState
is	ObjectInputStream
os	ObjectOutputStream
logger	Logger
remotePeerId	String
startTime	long
stopTime	long
running	boolean
asyncMessageSender	Thread
Method	**Return type**
PeerConnectionHandler(Socket peerSocket, PeerState peerState)	Constructor
setRemotePeerId(String remotePeerId)	Void

run()	Void
processChoke()	Void
processUnchoke()	Void
processHave(Message receivedMsg)	Void
processPiece(Message receivedMsg)	Void
setDataRate(int size)	Void
processRequest(Message receivedMsg)	Void
processInterested()	Void
processNotInterested()	Void
processBitField(Message message)	Void
getNextInterestingPieceIndex(BitSet remote, BitSet current)	Integer
processHandshake(Message response)	Void
receiveMessage()	Message
sendMessage(Message message)	Synchronized Void
broadcastMessage(Message message)	Void
stopAllConnections()	Void
stop()	Void
run()	Void

2. Handlers - This classes handles the connections between Peers. Two handlers are created, namely Incoming Connection Handler and Peer Connection handler.

3. Scheduler:

Task	**Description**
OptimisticUnchokingScheduler(PeerState currentPeerState)	Task scheduler to optimistically unchoke peers as per the protocol

Task	Description
PreferredNeighborsScheduler(PeerState currentPeerState)	Task scheduler to re-determine preferred neighbors as per the protocol

5. Messages

ActualMessage.java

Variable and Methods	Type/Return Type	Description
length	Int	Length of the message
payload	Object	Payload for the message
getLength()	Int	Returns the length of the message
setLength(int length)	Void	Sets the length of the message
getPayload()	Object	Returns the payload
setPayload(Object payload)	Void	Sets the payload
toString()	String	Returns the entire message in string format

Message Classes

Variables/Method	Type/Return Type	Description
BitfieldMessage	Constructor	Creates the bitfield message
getPayload()	BitSet	Returns the payload
ChokeMessage()	Void	Sets the Choke message
Header	String	"P2PFILESHARINGPROJ"
ZERO_BITS	String	"0000000000"
HandshakeMessage(String peerId)	Handshake message	Sets message type and peerID for Handshake message
toString()	String	Returns the combined message
getPeerId()	String	Returns the PeerID
validate(String peerId)	Boolean	Validates the message and peerID

HaveMessage(Integer index)	Constructor	Sets have message characteristics
InterestedMessage()	Constructor	Sets Interested Message Characteristics
getMessageType()	Message Type	Returns the message type
setMessageType(MessageType messageType)	Void	Sets the message type
MessageType	Constructor	Sets and fetches the classified message “type”
NotInterestedMessage()	Constructor	Creates the Not Interested Message
PieceMessage(byte[] payload, int index)	Constructor	Sets the properties of the piece message
RequestMessage(Integer index)	Constructor	Sets the properties of the request message
UnchokeMessage()	Constructor	Sets the properties of the Unchoke message

6. Utils: This module contains the utilities for handling all the utilities providing support to the main functions and processes in the above mentioned modules. The classes in this module are dedicated to the file utilities, common constants for the entire package grouped together in the PropertiesEnum class and logger variables and methods clubbed in logger class. An important point to note here is the fact that the logger utility class has been implemented as singleton in this design so as to avoid concurrency issues.

FileUtils.java

Method	**Return Type**	**Description**
splitFile()	ConcurrentHashMap<Integer, byte[]>	Returns fileSplitmap
joinPiecesAndWriteFile(PeerState peerState)	Void	
makeFilesAndDirectories(String peerId)	Void	

Constant	**Description**
NUMBER_OF_PREFERRED_NEIGHBORS	Number Of Preferred Neighbors

UNCHOKING_INTERVAL	Value of the unchoking Interval (sec)
OPTIMISTIC_UNCHOKING_INTERVAL	Value of the optimistically unchoking Interval (sec)
FILENAME	Name for the file
FILESIZE	Size of the file
PIECESIZE	Size of the individual pieces
COMMON_PROPERTIES_CONFIG_PATH	Path for the common properties
PROPERTIES_FILE_PATH	File path for the general properties
PROPERTIES_CREATED_FILE_PATH	Project/Peer Properties file path
PEER_PROPERTIES_CONFIG_PATH	File path for the properties in PeerInfo
PEER_LOG_FILE_EXTENSION	.log file extension
PEER_LOG_FILE_PATH	Logger files filepath

Method	**Return Type**	**Description**
PropertiesEnum(String value)	Enum	Assigns the input values to the constants in the Enum
getValue()	String	Fetches the value for constants in the Enum at a given time.

PropertiesEnum.java

Logger.java

Class Variable	**Type**	**Description**
Map	Map<String, Logger> map	Hash map for logger and PeerID
printWriter	PrintWriter	Initializing Printwriter
peerId	String	PeerID for each peer
Method	**Return Type**	**Description**
getLogger(String peerId)	Logger class	Returns the logger for each peerID
Logger(String peerId)	Constructor	Creates directories for logging and initializes Printwriter
makeLogDirectoryForPeer(String peerId)	Peer Log file	Logger directories for each peer

initPrintWriter(File file)	Void	Initializes Printwriter
getTimeStamp()	String	Timestamp for logger
writeFile(String message)	Void	Prints the input message
logReceivedHaveMessage(String fromId, int pieceIndex)	Void	Print log for Have message received
logTcpConnectionTo(String toId)	Void	Print log for making a Tcp Connection to peer
logTcpConnectionFrom(String fromId)	Void	Print log for making a Tcp Connection from peer
logChangePreferredNeighbors(Map<String, String> preferredNeighbors)	Void	Print log for changing preferred neighbors
logNewOptimisticallyUnchokedNeighbor(String unchokedNeighbor)	Void	Print log for a new optimistically uncooked neighbor
logUnchokingEvent(String peerId1)	Void	Print log for an unhooking event
logChokingEvent(String peerId1)	Void	Print log for the choking event
logInterestedMessageReceived(String from)	Void	Print log for receiving Interested message from a peer
logNotInterestedMessageReceived(String from)	Void	Print log for receiving Not Interested message from a peer
logPieceDownloadComplete(String from, int pieceIndex, int numberOfPieces)	Void	Print log for a completed piece download
logDownloadComplete()	Void	Print log for a completed download