IIT-Hyderabad

Project Document

BroadCast and Multicast Routing

Computer Network 2
(CS3543)

Project Name	Implementation of Broadcast and Multicast Routing	
Date	*2019-03-31*	
Author	*Harshit Patel - cs16btech11017*	*Shreyash Anarase- es16btech11002*

Contents

1 SW Development Plan

1.1 Project Overview

1.1.1 Objective and Project Scope

Objective:

To implement broadcast and multicast routing and compare their performances.

Scope of Project:

- *Demonstrating multimedia streaming to a dynamic group of hosts.*
- *Highlighting the inefficiencies of broadcast routing and emphasizing the importance of multicasting.*
- *Implementing IGMP protocol and Multicast routing algorithms.*
- *Simulating the working of routers and routing algorithm.*
- *Implementation of broadcast routing algorithms.*
- *Building and testing using circleCI*

Major review items
Simulating clients which can be part of several multicast groups.
Simulating routers for packet forwarding and group management.
Centre based approach for spanning tree creation.
Dynamically modifying group of hosts.

1.2 Assumptions, Dependencies and Constraints

<<Identify and List down the Assumptions, Dependencies and Constraints of the SW System>>

Item	Assumptions, Dependencies and Constraints	Remarks
1.	Identifying receivers of a multicast packet.	
2.	Addressing a packet sent to these receivers.	
3.	Designing the router functionalities.	
4.	Simulating multiple multicast groups with dynamically changing hosts.	
5.	Buffering of packets for persistent delivery.	
6.	Minimizing the lag while playing the multimedia at receivers end.	

7.		

1.3 Roles and Responsibilities

Student Name	Roles and Responsibilities
Harshit Patel - cs16btech11017	**Developer** **Software Requirements Analysis** Verifying requirements and performing analysis on requirements; Multicast routing
Shreyash Anarase- es16btech11002	**Developer** **Software Architecture** -Mapping the requirements into Architecture Broadcast routing
Software Development	**Developer -** *Harshit Patel* • *multimedia streaming* • *performance measure between multicast and broadcast routing*
	Developer - *Shreyash Anarase* • *testing using circleCI*

1.4 Development Plan

1.4.1 Development Schedule

Estimated Project Period	25/03/2019 - 30/04/2019
Project Team Size	2
Estimated Man Months	2

Milestone	1st Review	Final Review
Planned Schedule	2-April-2019	During final exam week.

1.4.2 Development Environment

Item	Development Environment	Remarks
Program Languages	*C++*	*Follow the OOP design rule*
Compiler, Build	*G++ 7.3.0*	*A new version is expected if a chip is changed* *Specify compiler version.*

Target Kernel	*LINUX 4.1.0 and above*	*The version is expected to be changed according to new chipset.*
Word Processor for Document Creation	*MS Word, MS Excel*	
Configuration Management	*Github, circleCI*	

2 SW Requirements Specification

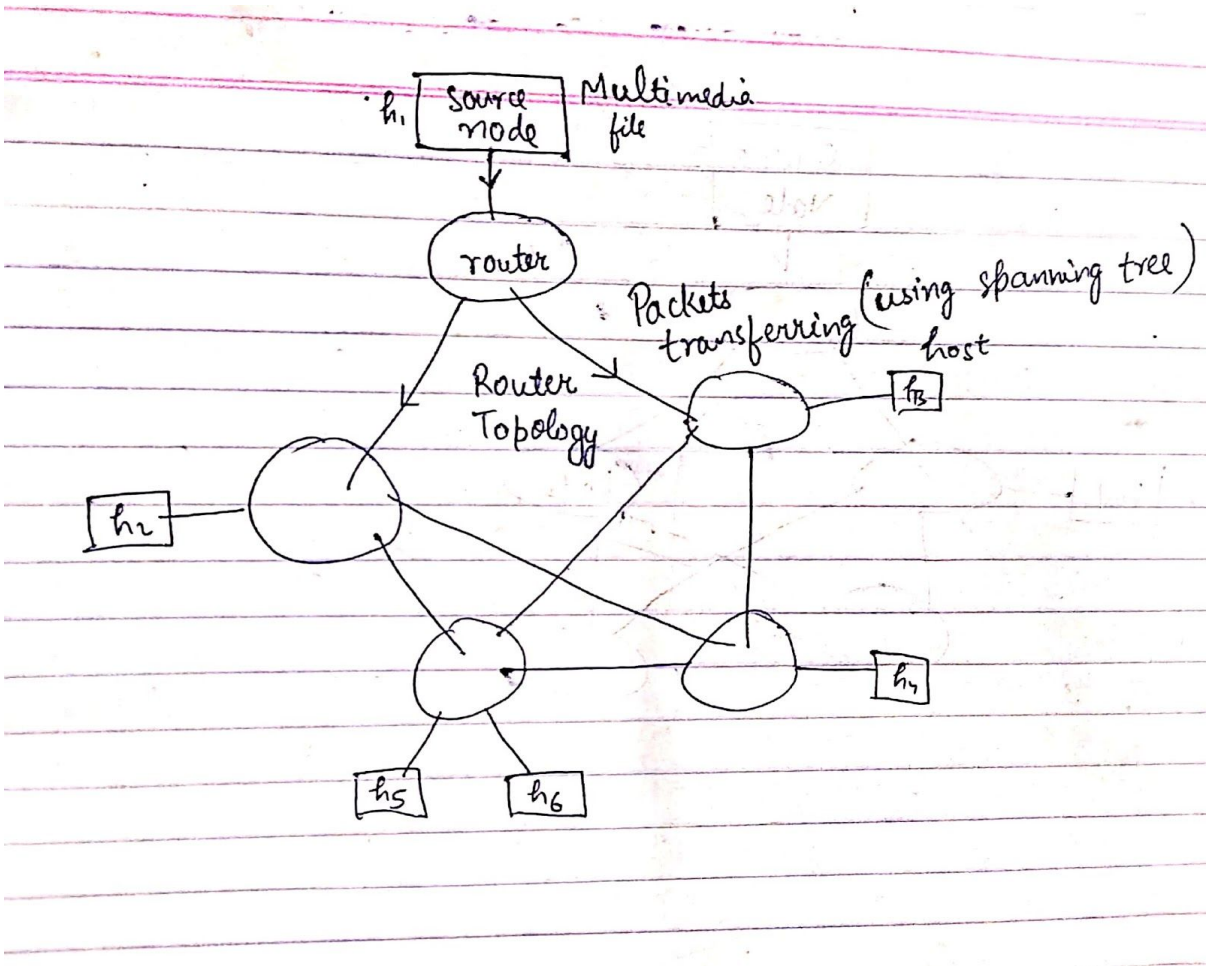
2.1 Major Functional Requirements

No	Requirement Id	Function Requirement Name	Description
1	*performance_id*	*performance*	Ability to stream multimedia without loss and lag
2	algo_id	algorithms	Correct implementation of multicast and broadcast algorithms.
3	graph_id	graph	A connection graph between hosts, router, server for streaming multimedia.

3 SW High & Detailed Level Design

3.1 Overall Architecture

Multicasting application:



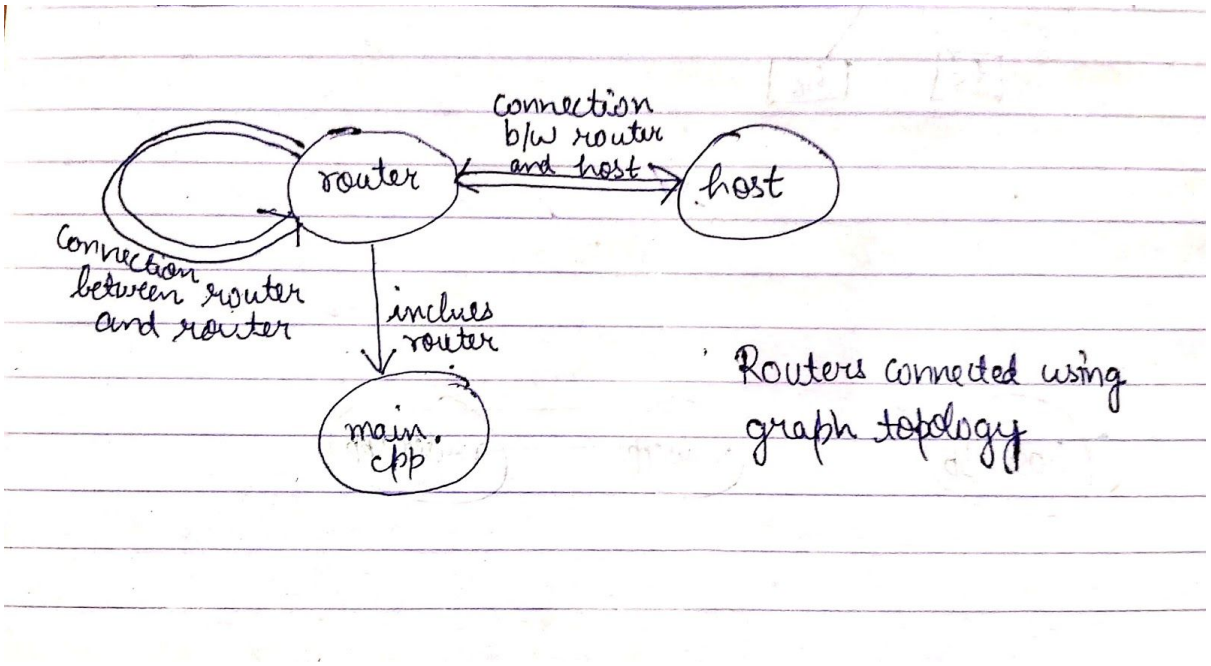
3.2 SW System Operation Design

We have 2 classes:

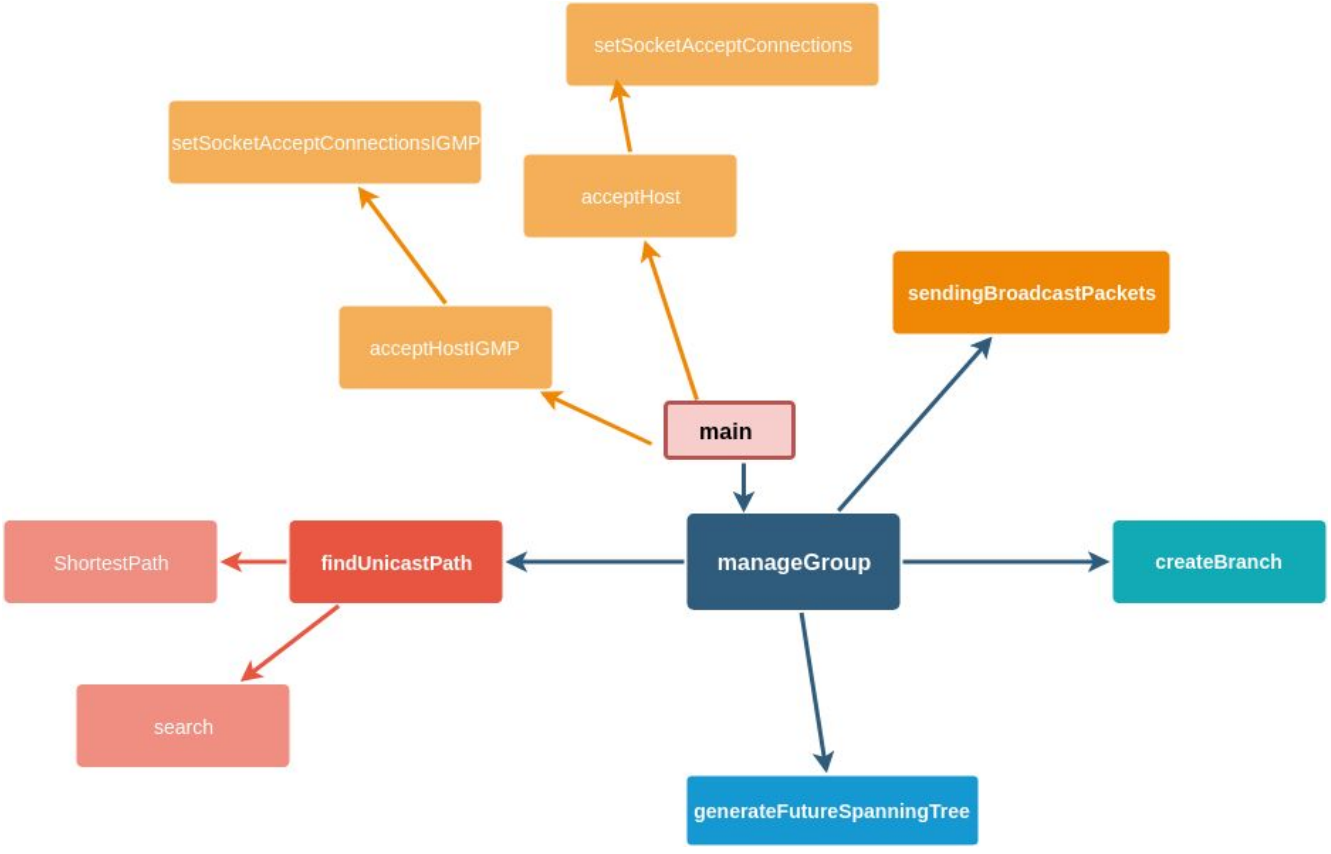
Host

Router

Main used to create instances of routers and connect them among themselves using the graph topology and to the hosts.



3.2.1 {DesignID} Structure Diagram



Major functions :
these functions are run in threads.
acceptHostIGMP: setSocketAcceptConnectionsIGMP
acceptHost : setSocketAcceptConnections

connectServerRouter
connectServerRouter

no. of threads = no. of content senders
manageGroup : findUnicastPath , createBranch, generateFutureSpanningTree, sendingBroadcastPackets, sendingMulticastPackets

findUnicastPath : ShortestPath, search

n threads
createBranch
to create the spanning tree

generateFutureSpanningTree

sendingBroadcastPackets

sendingMulticastPackets

3.2.1.1 {Module 'n'} Component Design

3.2.1.1.1 Module Description

Describe the functions of the corresponding block.

Module	Description
acceptHost, acceptHostIGMP	to accept hosts and connect them to routers
setSocket setSocketIGMP	creating the sockets for communication
Managegroup	manage the sending of one media file
sendDataToRouter() recvDataFromRouter()	send and recieve data to and from routers
indunicastpath, search, shortestpath	finding the shortes path from from each router to centre router
createBranch	creating the spanning tree for the graph topology
nerateFutureSpanningTre e	dynamically modifying the spanning tree

3.2.1.1.2 Sequence Diagram

- *Initially we run over main.cpp file which acts as a master program.*
- *It takes the input of number of routers.*
- *Then we create separate hosts and connect them to their corresponding routers.*
- *Until this connection is set up, all the hosts and routers wait.*

- *After this is done, we input the graph topology which demonstrates the connection between the routers*
- *Connection between routers is established using this topology.*
- *Each node informs its corresponding router about the groups to which it is subscribed using IGMP protocol. This is done in parallel.*
- *Next, we create a spanning tree, using center based approach.*
- *The source nodes starts sending packets along this spanning tree generated.*
- *Each hosts reads from its separate files, to which channels should it listen at any given point. This information is communicated to its routers.*
- *Based on this the structure of spanning tree will keep on changing. A thread runs in the background for this purpose.*
- *A separate spanning tree is created for each source.*
- *A router on receiving a packet forwards it to its hosts(if host is present in the particular group) or just forwards the packet for other hosts.*
- *Each host on receiving the packet, plays the live packets and also stores the packet in a file designated for this for future access and to demonstrate the dynamically changing groups*

3.3 SW Code Structure

- **Mapping list of modules and files (or folders)**

Modules **Files/Folder**

- **Implementing host class and its functions : host.cpp**
- **Implementing router class and its functions : router.cpp**
- **A master program to establish router network: main.cpp**
- **Hosts sending memReport(IGMP) : MemReport**
- **Hosts storing the content received by them : 0/1/2/3/4/5/6**
- **For each group for a host : subfolder using group name inside 0/1/2/3..**

3.4 Requirement vs. Module Mapping

Requirement ID	SW Design Elements		Notes
	Module/Function	Class	
Connecting routers and clients.	*acceptHost()* *acceptHostIGMP()*	*main.cpp*	
Creating sockets for communication	*setSocket()* *setSocketIGMP()*	*host.cpp* *router.cpp*	
Storing unicast path towards center	*findUnicastPath()*	*main.cpp*	
calculating shortest distance for communication	*shortestDistance()* *search()*		
Managing further host and router IGMP communication	*manageHost()* *manageRouter()*	*router.cpp* *main.cpp*	

Communicating data across routers	*sendDataToRouter()* *recvDataFromRouter()*	*router.cpp*	
Managing a particular group for its streaming	*manageGroup()*	*main.cpp*	
Sending membership report and receivng membership query	*routerCommunication()*	*host.cpp*	
Creating dynamic spanning tree	*generateFutureSpanningTree()*	*main.cpp*	
Appending a branch to spanning tree	*createBranch()*	*main.cpp*	
Playing the audio file as it is received	*play0()* *play6()*	*host.cpp*	

4 SW Unit Test Report

4.1 Bugs known at submission date

DATE: 30/4/2019
Sometimes the host playing the packet is not terminated after all packets are played.
This is a minor bug.

5 SW Development Completion Report

5.1 Project Result Analysis

5.1.1 Development Work Promotion Results

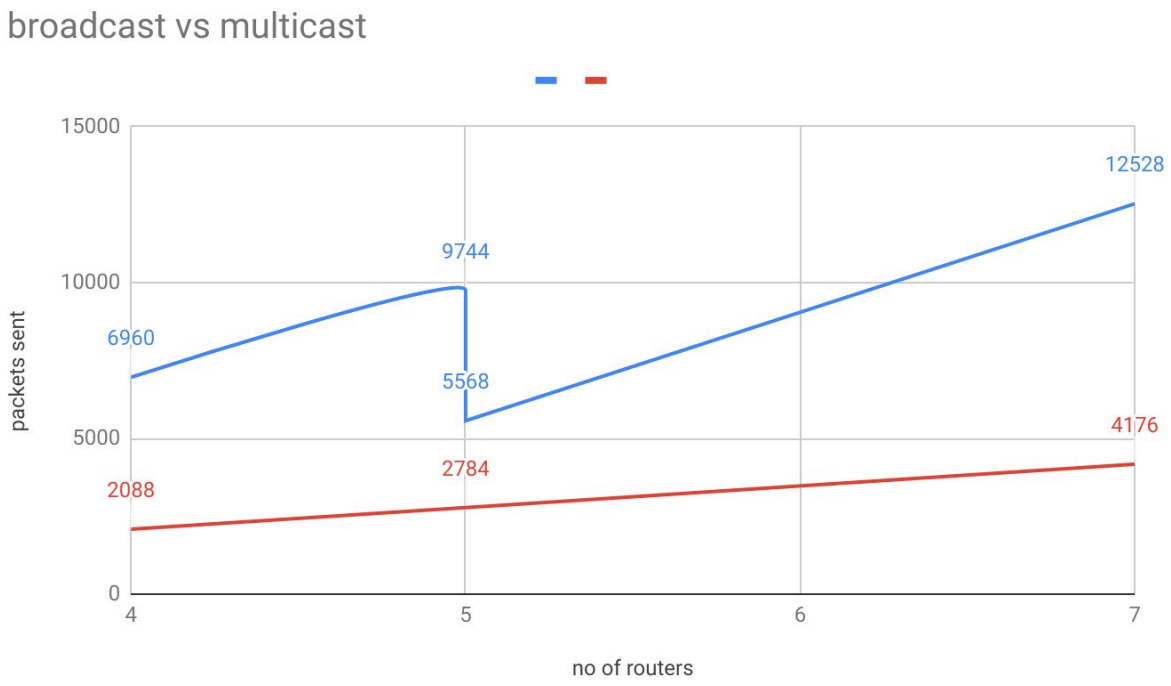
Implemented Multicasting and simulating routers, hosts and their communication
Implemented IGMP protocol and center based spanning tree creation
Multimedia transfer by splitting it into smaller files and playing individual files continuously at the host end.

Items	Result
Broadcasting and multicasting :	**Difference in the number of packets send for a file streaming.**

5.1.2 Development Results and Utilization

RESULTS

For this graph we plotted packets sent in broadcasting and multicasting vs the number of routers for different graph densities when the router number is constant.
It was observed that changing graph density effects broadcasting more than multicasting for which the packets sent didn't differ much with respect to density.



This application can be further developed to improve the streaming quality at the receivers end and minimizing the buffering. It can be used for multimedia streaming, text streaming.

5.1.3 Deliverables List

Executable Name	Description
main	To implement router and host communication and Multicast implementation
host	To run a host node.

Libraries : c++ thread, mutex.

■ References

Computer Networking : A top down approach , kurose ross 6th edition