**Modifications Done in One Simulator**

**In default settings.txt**

* Scenario Name changed as:

Scenario.name = Apocalypse

* Router Name changed as:

Group.router = ProphetRouter

* Message TTL changed as:

Group.msgTtl = 720

* Created three groups of nodes

1. ALPHA [0 1]
2. BETA [2 3]
3. GAMMA [4 5]

Scenario.nrofHostGroups = 3

Group.nrofHosts = 2

Group1.groupID = ALPHA

Group1.speed = 1.5, 2.5

Group2.groupID = BETA

Group2.speed = 1.5, 2.5

Group3.groupID = GAMMA

Group3.speed 1.5,2.5

Bluetooth Interface Transmit Range

btInterface.transmitRange = 250

**Simulating the Scenario in Local Machine**

**In DTNHost.java**

A Folder name Game is created within the src folder. In this folder folders having names of the nodes are created as soon as the simulation starts.

* Modifications done in **DTNHost constructor** of coreDTNHost.java

File theDir = **new** File("C:\\Mini\_Project\\One Simulator Media\\src\\Game\\"+**this**.name); //Path of your device

// if the directory does not exist, create it

**if** (!theDir.exists()) {

System.*out*.println("creating directory: " + theDir.getName());

**boolean** result = **false**;

**try**{

theDir.mkdir();

result = **true**;

}

**catch**(SecurityException se){

//handle it

}

**if**(result) {

System.*out*.println("DIR created");

}

}

//RANIT

**Working with Multimedia in ONE- Linking the OpenCV Library**

* Keep the **OpenCV\_2413.jar** file in src/lib/.
* Modifications done in **one.bat** to link OpenCV Library at Runtime.

java -Xmx512M -cp .;lib/ECLA.jar;lib/opencv-2413.jar;lib/DTNConsoleConnection.jar core.DTNSim %\*

**Converting Image Data to .txt files**

A folder named Content is created in the src/input folder where the image file is stored.

A class extract.java is created within the Package input to convert image data to string and write it in Text file.

**extract.java**

package input;

import java.io.BufferedWriter;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.OutputStreamWriter;

import org.opencv.core.Core;

import org.opencv.core.Mat;

import org.opencv.core.Size;

import org.opencv.highgui.Highgui;

public class extract {

public static void grayscale()throws IOException

{

System.loadLibrary(Core.NATIVE\_LIBRARY\_NAME);

Mat frame=new Mat();int i=0,j=0;String strr,str,strrr;double row,col,gray;String k;

strr="C:\\Mini\_Project\\One Simulator Media\\src\\input\\content\\image.jpg";

frame = Highgui.imread(strr);

Size size =frame.size();

row=size.height;col=size.width;

FileOutputStream fos = new FileOutputStream("C:\\Mini\_Project\\One Simulator Media\\src\\input\\content\\image\_data.txt");

OutputStreamWriter osw=new OutputStreamWriter(fos);

BufferedWriter bw=new BufferedWriter(osw);

for (i = 0; i <row; i++)

for (j = 0; j <col; j++) {

double[] data = frame.get(i, j); //Editing image

gray=(data[0]+data[1]+data[2])/3;

k=Double.toString(gray);

bw.write(k+" ");

}

bw.close();

osw.close();

fos.close();

}

}

When Message is created, change is made in MessageCreateEvent class.

**In MessageCreateEvent.java**

* Modifications done in **MessageCreateEvent()** of input.MessageCreateEvent.java

//RANIT

String k;

**if**(id.startsWith("M"))

{

**try**

{

extract.grayscale(); //Converts Image to Text File

s = **new** Scanner(**new** File("input/content/image\_data.txt"));

}

**catch**(Exception e)

{

System.out.println("ERROR");

}

content = s.nextLine();

**try**

{

**if**((from>=0)&&(from<=1))

{

k="ALPHA"+Integer.toString(from);

}

**else** **if**((from>=2)&&(from<=3))

{

k="BETA"+Integer.toString(from);

}

**else**

{

k="GAMMA"+Integer.toString(from);

}

FileOutputStream fos = **new** FileOutputStream("C:\\Mini\_Project\\One Simulator Media\\src\\Game\\"+k+"\\image\_data.txt");

OutputStreamWriter osw=**new** OutputStreamWriter(fos);

BufferedWriter bw=**new** BufferedWriter(osw);

bw.write(content);

bw.close();

}

**catch**(Exception e)

{

//System.out.println("RELAX");

}

}

//RANIT

A class extract.java is created within the Package routing to reconstruct the image file from the text file.

**reconstruction.java**

package routing;

import java.io.BufferedReader;

import java.io.FileInputStream;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.Scanner;

import org.opencv.core.Core;

import org.opencv.core.Mat;

import org.opencv.core.Size;

import org.opencv.highgui.Highgui;

public class reconstruction {

public static void grayscale(String des)throws IOException

{

System.loadLibrary(Core.NATIVE\_LIBRARY\_NAME);

Mat frame=new Mat();int i=0,j=0;String strr,str,strrr;double row,col,gray;String readcon="";

Mat frame\_1;

strr="C:\\Mini\_Project\\One Simulator Media\\src\\input\\content\\white.jpg";

frame = Highgui.imread(strr);

frame\_1=frame.clone();

Size size =frame\_1.size();

row=size.height;col=size.width;

FileInputStream fis = new FileInputStream("C:\\Mini\_Project\\One Simulator Media\\src\\Game\\"+des+"\\image\_data.txt");

InputStreamReader isr=new InputStreamReader(fis);

BufferedReader br=new BufferedReader(isr);

readcon=br.readLine();

Scanner s= new Scanner(readcon);

for (i = 0; i <row; i++)

{

for (j = 0; j <col; j++) {

String word=s.next();

System.out.println(word);

gray=Double.parseDouble(word);

double[] data = frame.get(i,j);

//Editing image

data[0]=data[1]=data[2]=gray;

frame\_1.put(i,j,data);

}

}

Highgui.imwrite("C:\\Mini\_Project\\One Simulator Media\\src\\Game\\"+des+"\\image\_reconstructed.jpg",frame\_1); //Your Path

br.close();

isr.close();

fis.close();

}

}

When Message is transferred from one node to another, in order to simulate it in the folders change is made in ActiveRouter class.

**In ActiveRouter.java**

* Modifications done in **messageTransferred()** method of routing.ActiveRouter.java

//RANIT

String data=m.content;

String to=getHost().toString();

System.*out*.println(to);

**try**

{

FileOutputStream fos = **new** FileOutputStream("C:\\Mini\_Project\\One Simulator Media\\src\\Game\\"+to+"\\image\_data.txt"); //Your Path

OutputStreamWriter osw=**new** OutputStreamWriter(fos);

BufferedWriter bw=**new** BufferedWriter(osw);

bw.write(data);

bw.close();

**if**(getHost()== m.getTo())

{

System.*out*.println("Reconstruction Started");

reconstruction.*grayscale*(to);

}

}

**catch**(Exception e)

{

//System.out.println("RELAX");

}

//RANIT

**Passing String Content in Messages**

Keep the image file in Content Folder in package input. The grayscale function of the extract class writes a text file corresponding to the image file in that folder.

**In Message.java**

* Modifications done in **Message class** of core.Message.java

*//RANIT*

private String content;

*//RANIT*

* Modifications done in **Message() class constructor** of core.Message.java

*//RANIT*

public Message(DTNHost from, DTNHost to, String id, int size,

String content) {

*//RANIT*

*//RANIT*

this.content = content;

*//RANIT*

* Added a method called getContent() in core.Message.java

*//RANIT*

public String getContent()

{

return this.content;

}

*//RANIT*

**In MessageCreateEvent.java**

* Modifications done in **MessageCreateEvent** class of input.MessageCreateEvent.java

//RANIT

**private** Scanner s = **null**;

**private** String content;

//RANIT

* Modifications done in **MessageCreateEvent()** of input.MessageCreateEvent.java

//RANIT

String k;

**if**(id.startsWith("M"))

{

**try**

{

extract.grayscale();

s = **new** Scanner(**new** File("input/content/image\_data.txt"));

}

**catch**(Exception e)

{

System.out.println("ERROR");

}

content = s.nextLine();

**try**

{

**if**((from>=0)&&(from<=1))

{

k="ALPHA"+Integer.toString(from);

}

**else** **if**((from>=2)&&(from<=3))

{

k="BETA"+Integer.toString(from);

}

**else**

{

k="GAMMA"+Integer.toString(from);

}

FileOutputStream fos = **new** FileOutputStream("C:\\Mini\_Project\\One Simulator Media\\src\\Game\\"+k+"\\image\_data.txt"); //YourPath

OutputStreamWriter osw=**new** OutputStreamWriter(fos);

BufferedWriter bw=**new** BufferedWriter(osw);

bw.write(content);

bw.close();

}

**catch**(Exception e)

{

//System.out.println("RELAX");

}

}

//RANIT

* Modifications done in **processEvent()** method of input.MessageCreateEvent.java

//RANIT

Message m = **new** Message(from, to, **this**.id, **this**.size, content);

//RANIT

**In ActiveRouter.java**

* Modifications done in **messageTransferred()** method of routing.ActiveRouter.java

//RANIT

Message res = **new** Message(**this**.getHost(),m.getFrom(),

*RESPONSE\_PREFIX*+m.getId(), m.getResponseSize(), m.content);

//RANIT

**In DeliveredMessagesReport.java**

* Modifications done in getPathString() method of report.DeliveredMessagesReport.java

//RANIT

String str = m.getFrom().toString();

**for** (i=1; i<hops.size()-1; i++) {

str += "->" + hops.get(i);

}

str += "->"+ hops.get(i)/\*+"("+ m.getContent()+")"\*/;

**return** str;

//RANIT