Universität Stuttgart
Fakultät Informatik, Elektrotechnik
und Informationstechnik

Master Thesis Nr. 3407

Device Selection Algorithm for Mobile Traffic Offloading

Sumeet Mahajan

Studiengang:	INFOTECH
Prüfer:	Prof. Dr. Kurt Rothermel
Betreuer:	M.Sc. Patrick Baier
begonnen am:	15. October 2012
beendet am:	16. April 2013
CR-Nummer:	C.2.1, C.2.2, C.2.3, C.2.4, E.1

IPVS

Institut für Parallele
und Verteilte Systeme
Abteilung Verteilte Systeme
Universitätsstraße 38
D-70569 Stuttgart

NEXUS

Abstract

Aim of thesis is to reduce the data traffic from mobile network by exploiting Ad-hoc modes of communication. This can be achieved by sending message to few mobile nodes which will forward data to rest of the mobile nodes via Ad-hoc communication.

This thesis focuses on selection of those few mobile nodes (Target Set) which will receive message directly from mobile network and can effectively reduce the data traffic on mobile network by forwarding that message to many other mobile nodes via Ad-hoc communication. Thesis consists of two tasks. First task of thesis is to implement the concept of Mobile Traffic Offloading on android based smart phones using desktop as server and two smart phones as its clients. Second task of the thesis is to implement the same concept on a large scale with more scope for Target Set selection. Second task consists of five different approaches for the selection of Target Set of mobile. First approach is Random approach, in this approach server sends message to a randomly selected Target Set of mobile nodes from all available mobile nodes. Second and third approaches are Distance and Distance Extended. These approaches considers the euclidean distance between each pair of mobile nodes. Distance approach considers the current euclidean distance and checks the possibility of Ad-hoc communication. Distance Extended approach uses the current distance between the mobile nodes and also calculates the probabilistic future distance between mobile nodes. Fourth approach is Same Street, this approach considers the position of mobile nodes on the street map and uses that position to find Target Set of mobile nodes. Final approach of thesis is Path Length approach. This approach also is a map based approach and finds shortest distance between each pair of mobile nodes. This approach also considers the probability of Ad-hoc communication in future.

Finally this thesis presents the evaluation for both tasks. Evaluation of first task shows that how message delay changes with increase in message size. Evaluation of second task compares all Target Set selection approaches on the basis of number of Ad-hoc messages, Message Delay and Energy Consumption .Thesis also compares the performance of advance approaches(Distance Extended and Path Length) on increasing the Ad-hoc range of communication.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

According to the statistics provided by International Telecommunication Unit (ITU) number of mobile phone users are increasing rapidly and also we know that now a days everyone is switching to smart phones[2]. All most every smart phone user has access to multimedia data services such as live streaming, video feeds, multimedia messages etc. This increase in number of smart phone users has significantly loaded mobile network with huge amount of data traffic. Recent predictions have shown that these trends are going to increase continuously, all service providers are worried about it and are reacting in different ways to avoid this issue because these continuous trends are showing that soon network will reach its critical limit. Many research works have provided the concept of mobile traffic offloading.

Mobile Traffic Offloading This concept of mobile traffic offloading means to reduce the Data traffic from the mobile network. One such way is by shifting some amount of data traffic from the mobile network to Local Area Network using Ad-hoc mode of communication.

Consider a scenario with five mobile nodes. All mobile nodes are subscribed to same news channel, there is a high quality video feed of size 10 MB from that news channel and that video feed can tolerate delay upto 3 hours i.e. expiration time for that video feed is 3 hours. Following two cases will explain two different approaches that a mobile network can follow to send message that video feed to all subscribed mobile nodes:

- **Case 1:** As shown in figure 1.1 mobile network will send same high quality video separately to all five mobile nodes. It loads mobile network with total traffic of 50 MB.

- **Case 2:** As all mobile nodes are waiting for same message and it can tolerate delay, so mobile network will forward this message using an optimized approach shown in figure 1.3. Now mobile network will forward message to only two mobile nodes and these two mobile nodes will forward message to remaining three mobile nodes via Ad-hoc interface. This reduces the traffic on mobile network as mobile network will send message to only two mobile nodes instead of sending message to all five mobile nodes, with this optimized approach traffic load on mobile network is reduced to 20 MB.

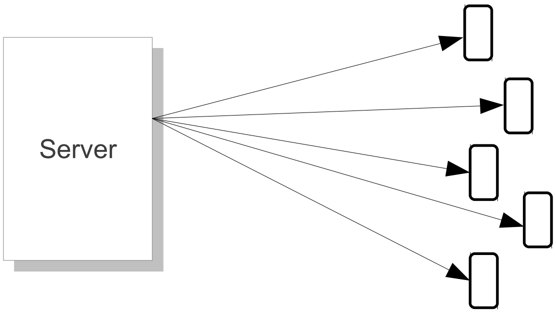


Figure 1.1: Case 1: All mobile nodes are receiving message via cellular interface.

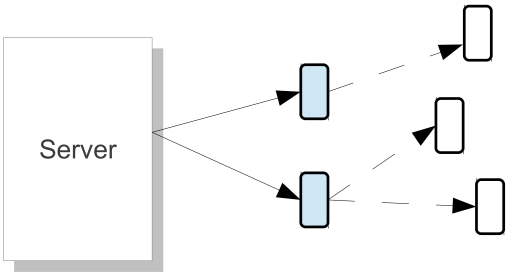


Figure 1.2: Case 2: Two mobile nodes are receiving message via cellular interface and message is forwarded to rest to mobile nodes via Ad-hoc interface.

1.2 Problem and Solution

This section explains the problem statement involved in thesis and its possible solution.

1.2.1 Problem Statement

Our goal is same as explained by Case 2 in previous section. We have to reduce the traffic in mobile network and to make it possible we can make use of Ad-hoc modes of communication such as Bluetooth and Wifi-Direct. According to our approach mobile network will not send message to all mobile nodes; only few mobile nodes will receive message via mobile network interface and these mobile nodes will then forward message to rest of the mobile nodes via Ad-hoc interface. We will call those few mobile nodes as Target Set of mobile nodes.

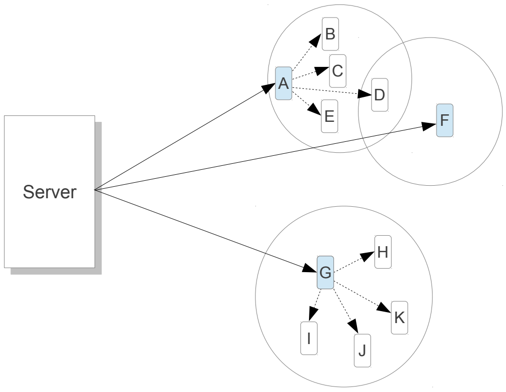


Figure 1.3: Bad Target Set

Figures 1.3 and 1.4 show the the difference between a good target set and a bad target set. According to figure 1.3 mobile nodes A,F and G were added into target set and according to figure 1.4 mobile nodes D and G were added into target set. Our aim is to reduce the traffic on the mobile network. Hence for this case, first figure is an example of bad target set as server is sending message to three mobile nodes but in second figure server is sending message to only two mobile nodes.

Main objective of our work is to find an efficient Target Set of mobile nodes from all available mobile nodes. Efficiency of our approach significantly depends on the selection of Target Set of mobile nodes. Making decision about selection of Target Set is a complex task. Mobile nodes added in Target Set should have a very good Ad-hoc coverage i.e.

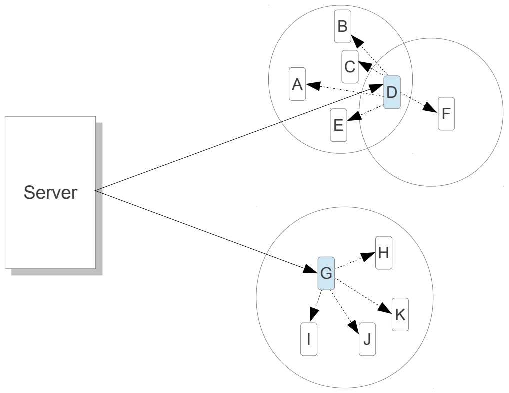


Figure 1.4: Good Target Set

mobile nodes should be able to make Ad-hoc connection with many other mobile nodes in order to increase forwarding via Ad-hoc interface. It is difficult to calculate constant Ad-hoc coverage for mobile nodes as mobile nodes will not stay at same place for long time.

1.2.2 Solution

As we discussed that selection of target set is a tedious task due to mobility of the mobile nodes. Hence, there can not be any exact information that mobile nodes can exchange message via Ad-hoc communication or not. In order to deal with this problem we will consider the probability that one mobile node can send message to other mobile nodes via Ad-hoc communication. This probability will be based on current position of mobile nodes, distance between two and prediction about their future positions.

We have proposed and implemented various algorithms for target set selection. Our all device selection algorithms consider the probability that one mobile node can communicate with other mobile nodes via Ad-hoc communication. All these algorithms are described in chapter 4.

1.3 Research Challenges

Designing and implementing Device Selection Algorithms for Mobile Traffic Offloading was a challenging task, this section explains the few challenges involved in this task

Mobility Prediction: In real life mobile node are always moving so it is difficult to predict that one mobile node can forward Ad-hoc message to how may other mobile nodes. If mobile nodes are moving very fast then there will be very small time intervals for Ad-hoc message forwarding.

Communication Interface: In our approach we are using two communication interfaces one is mobile network interface and other is Ad-hoc interface. Ad-hoc interface have limited resources such as communication range and data transmission speed.

Reliability: Reliability is another important factor for any of the Device Selection Algorithms, here reliability means that message should be delivered to every mobile node before message expires. But many times Ad-hoc networks are not reliable.

Delay Tolerance: Our approach does not assures immediate delivery of message, few mobile node will receive message immediately via cellular interface but Ad-hoc messages will be forwarded opportunistically from those few mobile nodes to rest of the mobile nodes. Delay tolerance is a challenge for our approaches as our approaches tolerates delay but it should be avoided.

1.4 Thesis Tasks

This thesis consists of two main tasks this section gives brief overview of the tasks, in next chapters we will discuss these tasks in details.

- **Task 1** This task was a small scale implementation using android platform. We developed an android application using two mobile nodes and a server. For this task two android based phones were used as mobile nodes and server application was running on desktop. We used this application to measure the message delay on sending messages of different size.
- **Task 2** This task was large scale implementation using The ONE (Opportunistic Network Environment) simulator, we will do detailed discussion about this simulator in Chapter 5. In this task we implemented different algorithms on The ONE simulator for target set selection and measured the resulting change in number of Ad-hoc transmissions. Efficiency of all algorithms is directly proportional to the number of Ad-hoc transmissions, as more Ad-hoc transmissions implies that there is less traffic on mobile network.

1.5 Thesis Structure

This thesis is consists of seven chapters:

- **Chapter 1** This chapter contains introduction to the topic and also describes the problem statement with its possible solution.
- **Chapter 2** This chapter contains information about the other research works related to this thesis topic.
- **Chapter 3** This chapter describes system design for Android setup and Simulator setup.
- **Chapter 4** This chapter contains detailed explanation about the conceptual design of Android setup and all device selection algorithms which are implemented on The ONE network simulator.
- **Chapter 5** This chapter contains explanation about the implementation details and information about the simulator.
- **Chapter 6** This chapter contains results obtained from different algorithms and comparison of results.
- **Chapter 7** This chapter summarises the work and gives a brief view about the possible future works.

Chapter 2

Basic Concepts and Related Works

The Introduction chapter presented introduction about thesis topic and our approach. This chapter describes the other works related to this topic, but there are few concepts which must be understood in order to understand those works so this chapter also explain about those concepts.

2.1 Overview

In previous chapter we discussed about the issue of increasing data traffic on mobile network. There are many approaches available now a days which can reduce the traffic from mobile network, so that large number of users can get access to data services with very less data traffic on mobile network. But all these approaches are being used for small scale. Large scale deployment of such approaches requires many tedious tasks such as mobility prediction. Further in this chapter we will discuss about many approaches for mobile traffic offloading and the challenges involved with these approaches.

2.2 Traffic Offloading using Femtocell

Femtocell acts as a small cellular base station, purpose of Femtocell is to provide better services of cellular network for indoor environments. It connects service provider's network via Broadband. Femtocell can be used to reduce the cellular traffic congestion and to improve the coverage as well as capacity, specially for indoor environments. Frequency spectrum used by femtocell is same as macro cell of the wireless network so it does not require user to add any kind of hardware in cell phones[5].

Figure 2.1 shows that how Femtocell can be used for indoor environment for mobile traffic offloading. As shown in figure femtocell connects user and service provider via internet connection and one internet connection is being used for many other mobile devices.

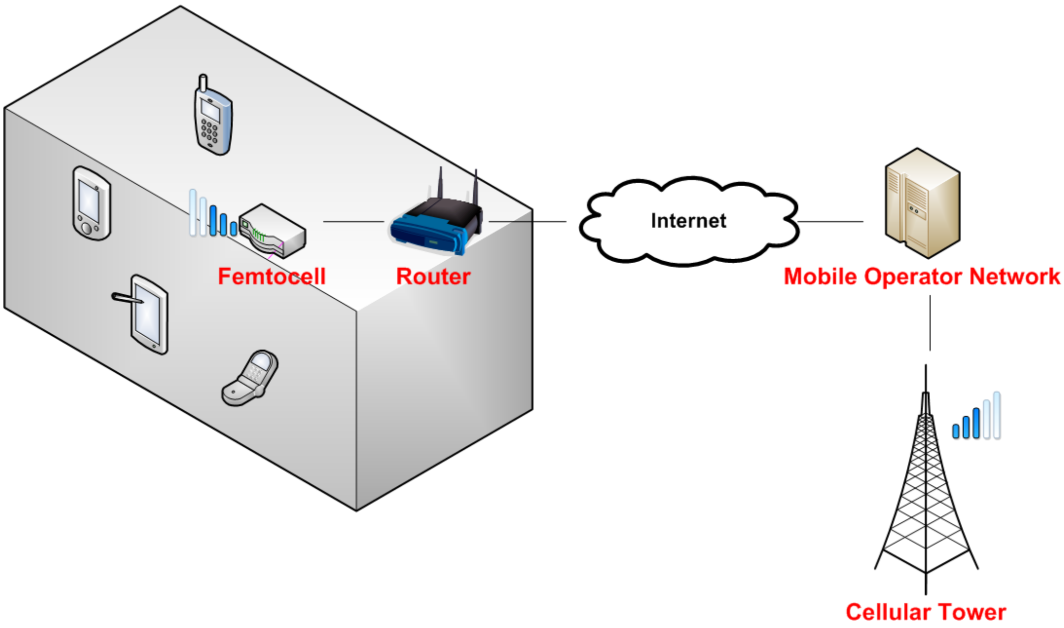


Figure 2.1: Femtocell for Indoor environment[17]

2.2.1 Issues related with Femtocells

Few issues which can arrive on using femtocells for traffic offloading are as follows[11]:

1. **Interference:** Interference can affect Femtocell in following ways:
    - There can be Interference issues between Femtocell and Metrocells[10] if they are using same frequency channel[3].
    - If many femtocells are installed near to each other then also interference between them will affect the quality of signal.
2. **Cost:** Using femtocells require customer to install some hardware in their premise, so cost is one more issue related to Femtocell.

2.3 Opportunistic Communication

It is a phenomenon of communication which requires both cellular and Ad-hoc mode of communication. Opportunistic communication is evolved from MANETs(Mobile Ad-hoc Networks) and DTNs(Delay Tolerant Networks)[7].

Opportunistic Communication considers the mobility of mobile nodes. So all devices in MANETs are free to move in any direction. It considers that mobile nodes can opportunistically exchange data in future even if their current position does not assure any direct path between them. Opportunistic communication does not assure that message will be transferred immediately, So it requires network that can tolerate some delay[11] such networks are called Delay Tolerant Networks.

2.3.1 Delay Tolerant Networks(DTNs)

Network can be define as delay tolerant on the basis of its insensitivity to delays. Such networks do not assume any type of end to end connectivity between mobile nodes. Data is transferred from one node to other node opportunistically i.e. transmission of data will be done whenever there is any kind of connectivity between two mobile nodes, these kind of transmission can only be possible in DTNs as they can bear delay upto several hours and even days. Some examples of such networks are Terrestrial Mobile Networks, Military Ad-hoc Networks etc. There can be few issues with DTN based mobile networks in realistic environment :

1. As we discussed before that DTN can tolerate some amount of delay while forwarding data. But in real time environment data will not be of same type and different types of data has different delay tolerance, data can be an audio file which can tolerate delay upto several hours and it can be an urgent message which should be delivered as soon as possible. So in realistic environment DTN has to deal with heterogeneous data and with such heterogeneous data it is difficult for a service provider to decide that that how such data can be offloaded.

2. In real time environment service provide has to deal with different types of users with different requirements. Users' demands, their interests will be different. So these things must be considered while offloading the data using DTN based mobile networks.

3. Delay Tolerant networks have limited resources such as storage, battery capacity etc.

Multiple Data Offloading[18] is approach which considers all such problems of the DTN based real time mobile network. Two different types of data has been used to simulate real environment. This approach has used two different types of users; first type is called helpers and the second is normal users. These helpers are linked directly with the service provides and will be used to forward data to rest of the users. These helpers acts as bridge between users and service providers. Data will be forwarded to helpers via cellular interface and then helpers will forward data to rest of the users. As we discussed before that this approach also considers the fact that data which has to be forwarded is of different types, So two types of data has been used and different types of data will be forwarded to different helpers according to the requirements of users linked with them.

Another approach based on Delay Tolerant Network is, Relieving the Wireless Infrastructure[29]. It considers a special time interval called as Panic zone. Panic zone starts few minutes before the message deadline expires.

According to this approach first server sends copies of data to few selected mobile nodes and these mobile nodes will forward data to rest of the nodes via Ad-hoc communication. Approach compares following two scenarios:

First Scenario: Server sends message to large number of mobile nodes at one time and wait until Panic Zone is reached . These mobile nodes will forward data to rest of the mobile nodes via Ad-hoc communication and all mobile nodes will send acknowledgement to server after message is received. If there are mobile nodes which did not send acknowledgement to server before Panic Zone, so server will send message directly to them without waiting for Ad-hoc communication.

Second scenario: Server sends data initially to few mobile nodes and the reinject the data after some time and this process continues till Panic Zone and at Panic zone server sends message directly to all mobile nodes who did not send acknowledgement without waiting for Ad-hoc communication.

2.4 Mobility Prediction Based Approach

Mobility prediction means to predict movement of mobile nodes. Getting this information in advance can be helpful for many cellular traffic offloading approaches. This information can be very useful to know that in future how many mobile nodes will be in Ad-hoc proximity of this mobile node. Wireless networks provides more flexibility for mobility prediction since they do not follow any fixed wired infrastructure[15].

Probability Based Mobility Prediction

In real time mobile network mobility prediction is always a challenging task, another solution for this challenge is by considering the probability that one mobile node can forward message to how many other mobile nodes via Ad-hoc interface. Next traffic offloading approach is based on Target set selection. First we will discuss briefly about targets set selection and how it can be helpful for Mobile Traffic offloading and then we will discuss that approach.

Offloading using Target Set Mobile traffic offloading can be done by selecting an efficient target set of mobile nodes. Target set of mobile nodes are the few mobile nodes that receive message directly from mobile network and then they forward this message opportunistically to rest of the mobile nodes as shown in figure 2.2.

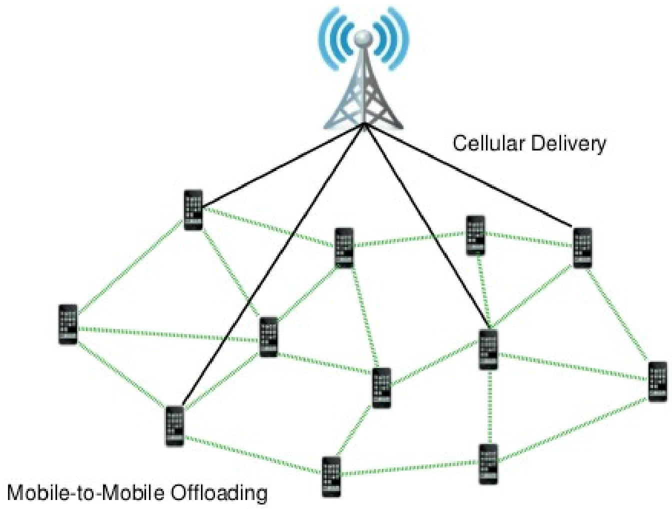


Figure 2.2: Traffic offloading using Target set[16]

The only factor that can impact the traffic offloading using Target Set is selection of initial target set of mobile nodes which will receive message via cellular communication, Different strategies can be used to to make an efficient target set selection[14]. Now we will discuss few approaches which are based on Target Set selection.

TOMP[1] presents many strategies based on graphs. It uses a factor called coverage. Coverage of a mobile node means that how many other mobile nodes are covered by it, i.e. how many other mobile nodes are in Ad-hoc proximity of this mobile node. A mobile node is said to have good coverage if it covers many other mobile nodes. TOMP uses current position of mobile nodes to calculate the current coverage and future coverage is calculated on the basis of meeting probability of mobile nodes in future. This coverage is then given as input to Greedy algorithm in form of a matrix and target set is received as output.

Another approach[16] for target set selection is based on an intensive 6-months long survey. Survey was done to find information regarding the daily activities of the mobile phone users. Aim of the survey was to find some non-random behaviour from the random motion of the mobile phone users. Results obtained from this survey were then used for target set selection using Random, Greedy and Heuristic algorithms. The figure above shows that how target set selection can help in Offloading cellular traffic offloading.

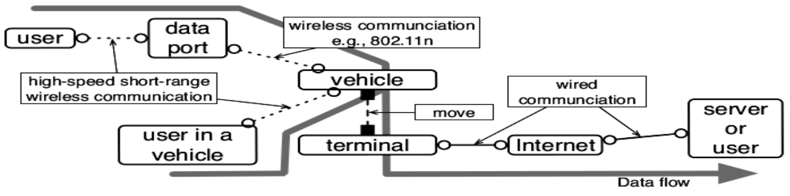
Mobile nodes in the target set will transfer data to rest of the mobiles nodes using Ad-hoc communication. Different modes of Ad-hoc communication can be used for this purpose.

Mobility Prediction using Random movements[12]:

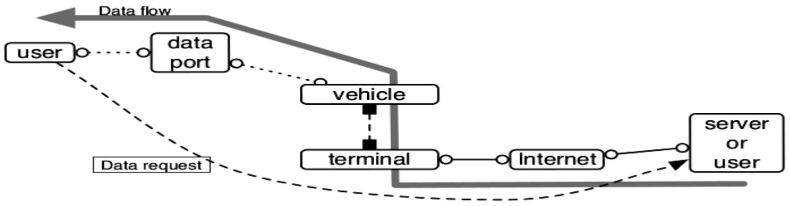
A normal mobile user will have a random path and can provides challenges while predicting mobility. OptiPath can be a great help for predicting non random motions. This is an android application which periodically sends location updates with a time stamp to server, this data is then used to predict mobility of the nodes from possible routes in google maps and can be compared with other available mobile nodes in those routes to predict the possibility of Ad-hoc communication.

Mobility Prediction using Non-Random movements[18]:

If we consider special cases like a tank moving in a battle field and a car following path suggested by navigator etc, then we can make use of Non-Random paths which will result in reliable mobility prediction.



(a) Uploaded data flow



(b) Downloaded data flow

Figure 2.3: Using non random motion of public transport

One more non random motion is movement of public transport vehicles according to this research work the public transport such as buses and trams etc, they all use a fixed path and fixed time so this fact can also be used for mobility prediction of mobile nodes. This approach has been used for both data uploading and downloading process.

As shown in figure 2.3 some data ports will be installed at bus stops and tram stations data can be downloaded from these ports and it can be uploaded to these ports. Then these ports will transfer data to some electronic storage system (installed in public transport vehicles) using wireless communication such as 802.11n and finally data can be forwarded to mobile nodes via reliable wired communication. Since Public transports does not have a random motion hence it is easy to calculate the delay, so before data transmission (Downloading or Uploading) the predicted delay is checked against the size of data to make sure that whether it is possible to transfer data before its expiration or not.

2.5 Traffic Offloading Using Social Behaviour

Every approach which has been discussed till now somehow can be influenced with the social behaviour of mobile nodes. Social behaviour of a mobile node means daily movement of mobile node users, their interaction with other mobile nodes and their participation in social networking websites. In this section we will discuss cellular traffic offloading approaches that exploits this kind of social behaviour.

Using daily interaction One such such approach is by using mobility traces of mobile nodes[16]. This approach exploited social interactions between mobile nodes. Mobility traces of students and researchers were used for simulation, users were divided in two categories; active and inactive users. User is said to be an active user if it can make contact with at least 1 another user before message delivery time ends. All active users are identified as target set using greedy algorithm and this social interaction is then used for message forwarding. Another approach suggested in same research work is using Reality Mining Trace, according to this approach around 100 mobile nodes were traced for around 11 months to find some regularity in mobility of the mobile nodes and this trace information was then used for offloading mobile traffic.

Another social behaviour based approach is Data Replication[20], this approach also exploits social behaviour and daily routine of mobile node user. Online Social networking can also be used for short range communication but the there are few privacy issues which will not be acceptable by users. No user would like to receive message from some strangers and use of online social networking can even create some kind of security threats.

2.6 Related Works Conclusion

2.6.1 Summary

In this chapter we discussed about the works related to the topic of Mobile Traffic Offloading. We concluded that in order to offload the traffic in mobile network we need to have Delay Tolerant Network. We have seen that the mobility prediction is very important for traffic offloading. We have seen that mobility prediction is tedious task as mobile node will be moving continuously. We discussed different mobility prediction approaches based on probability and based on motion of the mobile nodes.

2.6.2 Comparison to Our Approach

Our approach is based on Target Set selection and Opportunistic Communication. As we discussed before that efficient target set selection is very significant if message transfer is being done opportunistically. All of our algorithms for target set selection use cellular interface and then forward data message using Ad-hoc communication, so for our algorithms Ad-hoc range is an important parameter. We calculate the coverage of all available mobile nodes and treat them in form of matrix which is similar to *TOMP* explained in section 2.4.

This matrix is given as input to greedy algorithm and greedy algorithm calculates the target set. A mobile node is added into target set if it is in contact with atleast one other mobile node. This approach is similar to *Using Daily Interaction* approach explained in section 2.5.

Target set selection is a very important factor that affects the efficiency of our approach, we have done target set selection on the basis of coverage of a mobile node. To decide coverage of mobile nodes we have used current location of mobile nodes and the current coverage. In order to calculate Future coverage we have used the probability that mobile nodes will meet each other and this probability is also decided on the basis of current position of the nodes.

Calculation of coverage has been done in the same way as presented in *TOMP*. Target set selection problem is Np-hard. *TOMP* has used an very efficient way to avoid issues related to target set selection we faced similar issues and then we implemented the similar approach as mentioned in *TOMP*.

In section 2.3 we discussed about Opportunistic Communication and Delay Tolerant Networks. Our approach is also delay tolerant and nodes in target set forward message to remaining nodes opportunistically but for our approach we would like to reduce this delay in message transmission.

Reliability is another important factor of our approach. Reliability means that all mobile nodes should receive message before message deadline expires. So according to our approach, in the end server checks if it has received acknowledgement from all mobile nodes or not. If there are mobile nodes who did not receive message before message deadline expires then server will send message directly to them without waiting for Ad-hoc communication this scenario is similar to First scenario used by approach Relieving Wireless Infrastructure[29] explained in section 2.3.1.

Chapter 3

System Model

3.1 Overview

This chapter explains the system model for small scale implementation using Android Platform and large scale implementation on the network simulator. Our main purpose is to design a system which can reduce the data traffic on the mobile network by shifting some amount of traffic to Local Area Network. There are different Device Selection Algorithms which are implemented on simulator system model. These Algorithms will be discussed in next chapter.

3.2 System Model Android Implementation

Android implementation is small scale implementation of mobile Traffic offloading concept. System set-up for android task consists of two android phones and a server. For this task our server application was running on desktop and clients application on android phones. As per requirement of traffic offloading concept, system requires minimum of two clients. First android phone will be connected to desktop server via Wi-Fi interface and two android phones will be connected to each other via Ad-hoc interface. For Ad-hoc communication Bluetooth was used but system can be easily upgraded using advance mode of Ad-hoc communication called Wi-Fi Direct which exhibits transmission range upto 150 meters.

3.2.1 Server

System model consists of a server. As described before server application was running on Desktop. Server is leading component of the system model as it initiates the message transmission process. Message of different sizes were sent to observe the time delay in message transmission process. In our system model it is assumed that server already knows the location of mobile nodes. It will send message to one of the mobile node and then waits for acknowledgement. Server calculates the message delay with the help of inbuilt timer. This timer starts as soon as server sends message to client and stops after receiving acknowledgement form client. Server is enable to send different types to data to clients it can be data file, image etc. In our case we used images of size upto 10 mb.

3.2.2 Clients

System model consists of two mobile nodes with enabled bluetooth feature, these two mobile nodes will act as clients. One mobile node will receive message directly from server and after successful reception of message it will send acknowledgement back to server. After message is successfully received by first mobile node, it will be sent to another mobile node via bluetooth and finally acknowledgement will be sent from receiver mobile node to sender mobile node. In real life mobile nodes will not stay at same position for long time, to actualize this factor distance between two mobile nodes was changed for different measurements.

Figure 3.1: System Model Android Implementation

Figure 3.1 shows that server is sending message to first android phone via Wi-Fi interface, first android phone is sending message to second android phone via Ad-hoc interface and both android phones are sending Acknowledgement(ACK) back to server.

3.3 System Model Simulator Implementation

Simulator implementation is done to simulate real time behaviour of a mobile network communication with large number of mobile nodes. The system model consists of one server and many mobile nodes. For the measurement of different algorithms 100-900 clients are used. Connectivity between server and mobile nodes and mobile nodes between each other is an important factor for all measurements. In order to reduce the complexity of the system we do not consider any type of hardware failure. We consider that the hardware components such as server and mobile nodes are reliable and are tightly connected to each other. Different algorithms has been designed to select an efficient target set to increase the number of Ad-hoc transmissions. Efficiency of system is measured depending on the number of Ad-hoc messages sent.

Our aim is to reduce the number of data messages sent via cellular interface by selecting an efficient target set of mobile nodes from all available mobile nodes. Server will send message to few mobile nodes via cellular interface and these mobile nodes will forward message to rest of the mobile nodes via Ad-hoc interface. Different algorithms has been designed and implemented for this purpose. All these algorithms will select few mobile nodes that will receive message form server via cellular interface and then forward message to remaining mobile nodes via Ad-hoc interface.

3.3.1 Server

Server is central component of our system model. Our server in android implementation was dealing with just two mobile nodes but this server has to deal with many mobile nodes. Server can communicate with clients via cellular interface. We assume that server is able to handle following types of messages:

1. Messages sent from clients to server in order to inform the server about the existence of clients.
2. Acknowledgement sent by clients after successful reception of message.
3. Data message transferred from server to few mobile nodes via Cellular interface and then to rest of the nodes via Ad-hoc communication.

3.3.2 Clients

The mobile nodes in the simulator are called as clients. All clients have enable Ad-hoc interface with transmission range of 10 meters, which is equivalent to Bluetooth but transmission range. It can be changed by upgrading Ad-hoc interface from Bluetooth to Wifi-Direct with transmission range upto 150 meters. Clients are able to receive message from server via cellular interface and client can forward message to other clients via Ad-hoc interface.

Clients in simulator will not accept same message twice. All clients will send an acknowledgement back to server after successfully receiving the message. As we said before that this simulator implementation simulates the real time behaviour so clients are considered to be mobile. Speed of the mobile node depend on the speed of user who is holding it. In our case the speed varies form 0.5 m/s to 1.5 m/s.

We assume that clients are able to handle following types of messages:

1. Clients sends location update message as the initial step of communication to update the server about its existence.
2. Clients receives data message form server via cellular interface and will send it to other clients via Ad-hoc interface.
3. Clients sends acknowledgement to server after successfully receiving message.

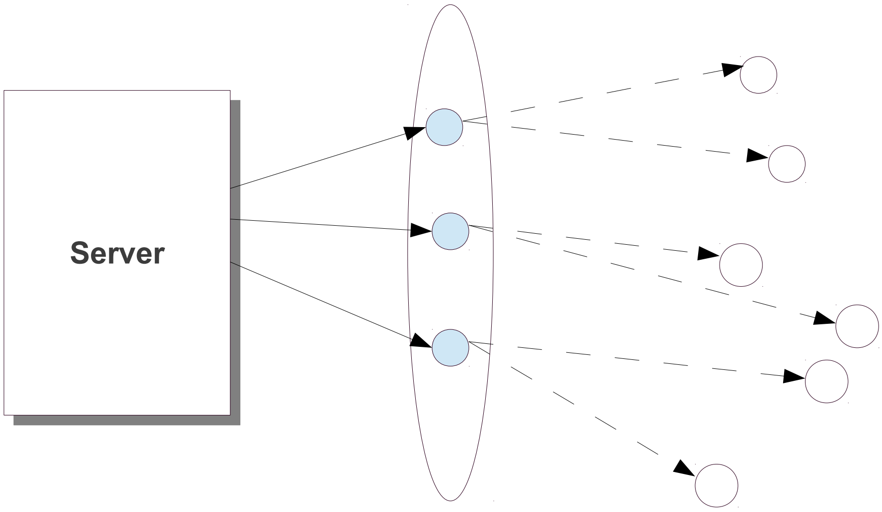


Figure 3.2: System Model Simulator Implementation

Figure 3.2 shows that server is sending message to few selected mobile nodes represented by blue circles and these mobile node are forwarding message to remaining mobile nodes represented by white circles. In above figure continuous line represents cellular interface and dotted line represents Ad-hoc interface.

Our system is delay tolerant, it means that system does not grantee that message will be transferred immediately to all clients. Sever will send message to all mobile nodes immediately after target is decided but the communication between mobile nodes is not immediate. Clients will forward message to other clients when they will come in contact with them as clients in our system are considered to be mobile. So delay in message delivery is not a problem for our system. System can handle maximum delay of 24 minutes and client who did not receive message even after 24 minutes will receive message directly after from server via cellular interface.

3.3.3 Communication Interface

Our system model has two modes communication. It can communicate through mobile network as well as through Ad-hoc network. UMTS interface is used for mobile communication and for Ad-hoc communication we are using bluetooth but system can be upgraded to advance mode called Wi-Fi Direct.

Modes of Ad-hoc communication:

1. Bluetooth[8]: It is wireless communication technology which uses ISM frequency band 2.4 to 2.485 GHz. It used for short distances data transmissions. The specific range of Bluetooth technology is 10 meters with transmission speed upto 24 Mbit/s.

2. Wi-Fi Direct[9]: It is new mode of communication which is used to connect two devices without using any wireless access point. It has data transmission range of nearly upto 182 meters with transmission speed of upto 250 Mbit/s.

Chapter 4

Concepts

4.1 Overview

The hardware components and system model was described in previous chapter. This chapter will describe that what concepts are implemented on the system model to achieve objective of the work, which is to reduce cellular traffic. Our approach for this will be to select an efficient target set of the mobile nodes which can help to reduce the traffic by exploiting Ad-hoc communication. In previous chapter we discussed system model for both small and large scale implementation. In this chapter we will discuss about conceptual design for small and large scale implementation of the Mobile Traffic Offloading .We will do detailed explanation about the different approaches that we used for target set selection. Initial parts of this chapter contains discussion about the small scale implementation done on Android platform, later this chapter will describe the different target set selection algorithms.

4.2 Small Scale Implementation using Android Platform

For this small scale implementation we used a desktop server and two android phones. An android application is developed, this application is basically a server-client model that sends data from Desktop server to one android phone via Wi-Fi and then from one android phone to another android phone via Bluetooth, application was first tested with Wi-Fi network and then with UMTS network. Server sends message to mobile node and wait for Acknowledgement, after acknowledgement is received by server data will be sent to another android phone via Ad-hoc network using Bluetooth.

4.2.1 Objective

This task is small scale implementation of concept of mobile traffic offloading using two mobile nodes. Motive behind this task was to observe the message delay in Wi-Fi and Ad-hoc network on changing number of bytes transferred and distance between mobile nodes. Message delay is calculated as subtraction result of start timer from end timer. Start timer gives the value of time when message sending process started and end timer is timer value after acknowledgement is received. This task also actualises the concept of mobile traffic offloading by shifting some amount of traffic to Ad-hoc network.

Figure 4.1 shows flowchart representation of the Small Scale Implementation. In next part of this section we will discuss all steps of this flowchart.

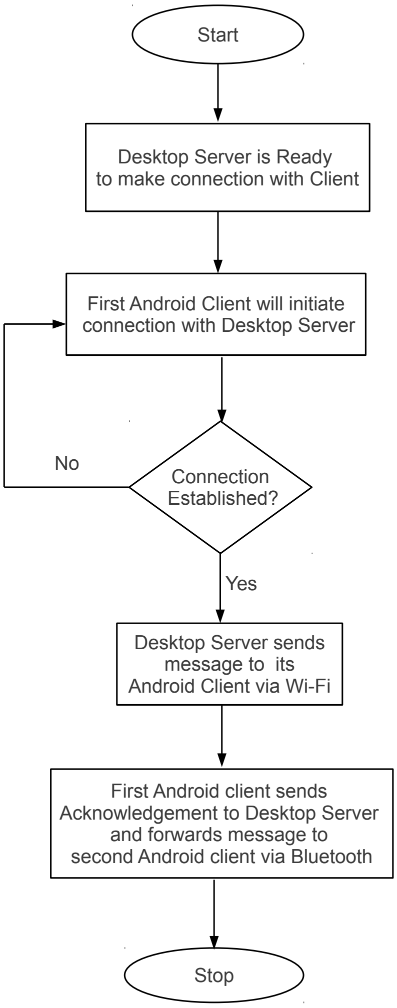


Figure 4.1: Flowchart Representation of Small Scale Implementation

Message Forwarding

This process of message forwarding is done in following steps:

1. Server will send message (Cellular MSG) to first android phone which is already registered to server via Cellular interface and will wait for acknowledgement.

2. This first android phone will send acknowledgement (ACK) to server after successfully receiving message from server,this android phone will act as server for second android phone.

3 Second android phone will be continuously listening to its server(first android phone), then message (Ad-hoc MSG) will be forwarded from first android phone to second android phone via Ad-hoc communication.

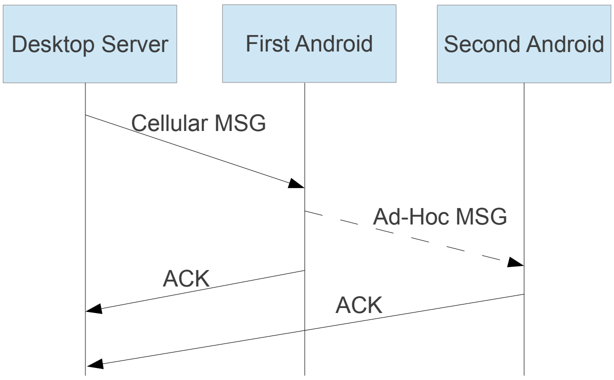


Figure 4.2: Message Delivery and acknowledgement

4.2.2 Types of messages

There were two different types of messages that were exchanged:

1. **Type of data sent:** Images with different sizes were sent from Desktop Server to android phone and then from one android phone to another android phone.
2. **Acknowledgement:** Acknowledgement was sent from two android phones to server after receiving the message successfully.

4.3 Large Scale Implementation

In previous section device selection for traffic offloading was done on a small scale with two mobile nodes. In this section we will discuss the real behaviour of mobile network using large number of mobile nodes.

4.3.1 Objective

Consider a case with some 100 mobile nodes, if there is a common message (news update, Video feed etc) for all mobile nodes, then mobile network has to send same message to 100 different mobile nodes. This will result in useless burden on the cellular network. Now consider another case that some of these mobile nodes are located near to each other and can easily exchange message with each other via AD-hoc communication. So server will send message to few mobile nodes and then these mobile nodes will further forward the message to remaining mobile nodes, which will significantly reduce the useless burden of the cellular network. Efficiency of this approach solely depends on the selection of mobile nodes (target set) which will receive message directly form the server.

Later in this chapter we will discuss some algorithms for selection of mobile nodes. Objective of all these algorithms is to find and efficient set of mobile nodes which can offload the traffic on the cellular network by increasing the number of Ad-hoc forwarding and reducing the number of forwarding via cellular interface.

4.3.2 Device selection and Message Forwarding

The process of selection of target set of mobile nodes and message forwarding is done in different steps. Now we will discuss every step of this process:

1. All mobile nodes will send a location update message to server.
2. Target set of mobile nodes will be calculated.
3. Server will send message to target set.
4. Remaining mobile nodes will receive message from the nodes in target set and will send acknowledgement to server.
5. In last few minutes server will send message directly to mobile nodes from whom no acknowledgement was received.

Step 1 This is the initial step of the process. Server does not have any information about the clients present in the network. So all client will send a message to server. This message will inform the server about the existence of the mobile nodes as well as the current location of the mobile nodes.

Step 2 This step is the most important step of the process. Among all mobile nodes best mobile will be added into the target set. This decision that mobile node should be added into target set or not will depend on the coverage of a mobile nodes, i.e. if a mobile node can send message to many other mobile nodes via Ad-Hoc communication then it has a good coverage and it can be added into target set.

Step 3 and 4 After target set of mobile nodes is decided, Process will flow in following steps:

1. Server will send message to these mobile nodes via cellular interface, after successful reception of message these mobile nodes will send an acknowledgement back to server.
2. In parallel message will be forwarded from mobile nodes in the target set to the rest of the mobile nodes via Ad-hoc communication and all mobile nodes who received message via Ad-hoc communication will also send acknowledgement to server after message has received.

Step 5 Last few minutes of the process is called as Panic Zone. In this zone server will look for all clients who did not send acknowledgement to server, then without any delay server will forward message directly via cellular interface.

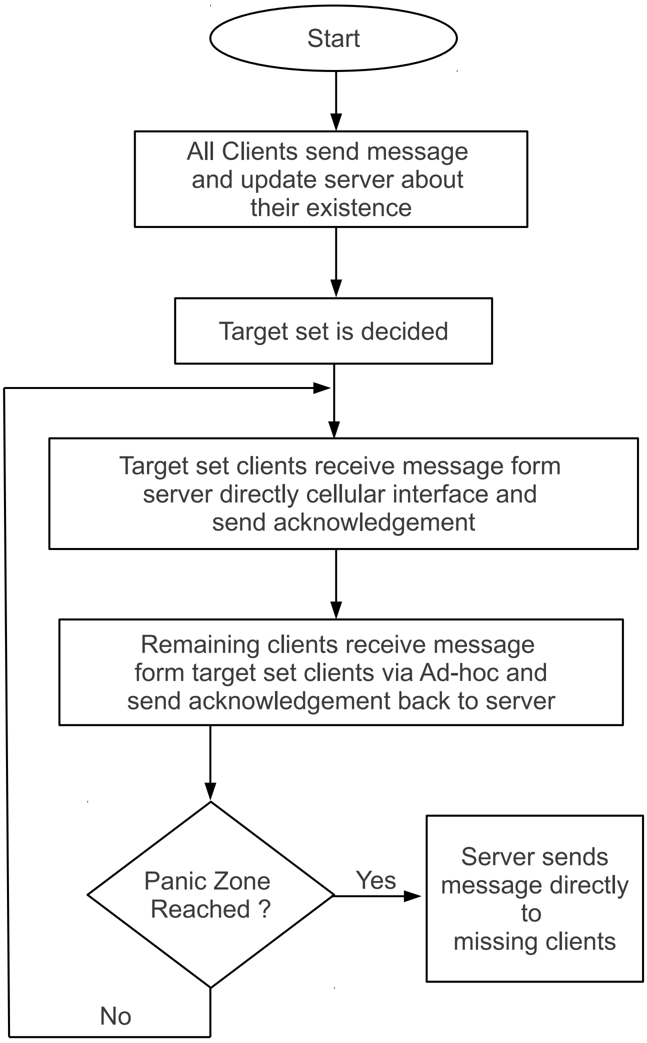


Figure 4.3: Flow chart representation of device selection and message forwarding process

4.3.3 Types of Messages

During the process of message forwarding we have used different types of messages such as location updates, acknowledgements and data messages. All messages are unique, every message has a source address, destination address and unique message ID. Messages were forwarded from Server to Clients and vice versa. Now we will discuss about all these messages.

Location Update Inquiry: This type of message will be sent by server and will be broadcast to all of the clients. In this message server will request all of the clients to send the information about their current location.

Location Update: All mobile nodes will send one location update to server as response to Location Update Inquiry sent by server. Every location update contains current position, last visited position and estimated future position of the client. First location update will inform server about its existence and then this location update will be used to estimate the future movement of the client.

Data Message(Request): The data message which is transmitted from server to selected target set and it will be transmitted to rest of mobile nodes via Ad-hoc communication. Like other messages this message also contains receiver's address. Size of this message is not fixed. It will be according to the data sent. To make the system more realistic this message comes with limited life period.

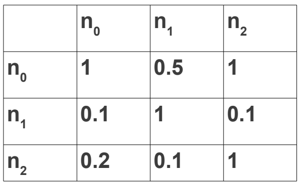
Acknowledgement(ACK): It is acknowledgement which will be sent by all mobile nodes to server after reception of "Request". Mobile nodes who will receive message from server will send acknowledgement to server and the mobile nodes who will receive message via Ad-hoc communication will also send message to server after successfully receiving the "Request".

4.4 Target Set Selection

In this section we will discuss the target set selection process. Target set is set of mobile nodes which will receive message directly from server via cellular interface. Selection of mobile node to be added in target set depends on its coverage. Before going further we will define coverage of a mobile node.

Coverage: Mobile node A is said to cover mobile node B, if A can successfully forward message to B via Ad-hoc communication before message deadline expires. Coverage of a mobile node can take any value form the range [0,1]. Higher value of coverage assures that there is high possibility that mobile node A can forward message to mobile node B either now or in near future. Any mobile node is said to have good coverage if it covers many other mobile nodes.

Later in this chapter we will discuss different algorithms to calculate coverage between each pair of mobile node. To make explanation easier, now we consider that coverage between any two nodes is any value from range [0,1] these coverage values will be treated as a |N|x|N| matrix. Where N is total number of mobile nodes. This coverage matrix will be given as input to target set selection algorithm which is discussed next in this chapter.



	n_0	n_1	n_2
n_0	1	0.5	1
n_1	0.1	1	0.1
n_2	0.2	0.1	1

Figure 4.4: 3x3 Coverage matrix

Figure 4.4 shows a simple 3x3 coverage matrix for three clients n_0 , n_1 and n_2. As explained before each element in this matrix is probability of coverage of between clients in corresponding row and column. Coverage(n_0, n_1) = 0.5 it means that probability that node n_0 covers n_1 and vice versa is 0.5.

4.4.1 Target Set Selection Process

After calculating coverage matrix. This matrix will be used for target set selection. Every approach will give a coverage matrix as output. Now we will explain that how that coverage matrix will be treated to obtain best target set for each algorithm. As per requirement of thesis target set selection will be done in order to maximise the Ad-hoc forwarding and to minimise the traffic in mobile network. As discussed in [1] this problem can be seen as well known set cover problem.

Set Cover Problem

Definition: Consider a universal set U of m elements .i.e $|U| = m$. Let X be a set of subsets of U such that $X = \{S1, S2....Sn\}$ such that $Si \subseteq U$ where $1 \leqslant i \leqslant n$. The set cover problem associates each of these subsets with a cost 'C' and aims to find collection of subsets which entirely covers universal set U while keeping cost minimal.

For example,assume that we are given a universal set with following elements $U = \{1, 2, 3, 4, 5\}$ and $X = \{\{1, 3, 5\}, \{1, 2, 4\}, \{5, 2\}, \{1, 3, 2\}\}$ with cost $Cj = 1$, $1 \leqslant j \leqslant 4$.

So it is visible that there is no subset of X which entirely covers universal set but we can cover the entire set with this combination of subsets $\{\{1, 2, 3\}, \{4, 5\}\}$ and union of different subsets can be used after considering cost factor but there is no unique solution.

Complexity of Set Cover Problem

Set covering problem is NP-hard. So it is unlikely to have a unique solution for target set selection. Resulting coverage matrices from Distance Extended Algorithm and Path Length Algorithm (will be discussed in section 4.5) can contain any polynomial value from range [0,1] which means that whatever set target set is obtained after computation is just an approximation and not a best and unique result. To fulfil the requirement of the target set selection task greedy algorithm[1] will be used. This algorithms deals with NP-hard problem and will result in efficient way for target set selection.

Greedy Algorithm

Consider U a universal set and $X = \{S_1, S_2, ..., S_n\}$ is a set of n subsets of universal set U. Algorithm will return Output O which covers universal set U entirely. Greedy algorithm picks one subset form X with largest number of elements till it entirely covers universal set U. Algorithm 1 describes the flow of greedy Algorithm.

Greedy Algorithm can be understood in a better way after considering following example.

$U = \{3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$, $X = \{S_1, S_2, S_3, S_4, S_5\}$ with subsets $S_1 = \{3, 4, 5, 8, 9, 11\}$ $S_2 = \{3, 4, 5, 6\}$ $S_3 = \{6, 7, 10\}$ $S_4 = \{3, 6\}$ $S_5 = \{4, 5\}$ Greedy algorithm will add S_1 to Output set O as it is the set with largest number of elements and then Subset S_1 will be subtracted from all of the sets. After first iteration there will be no S_1,S_4,S_5, $S_2 = \{6\}$ and $S_3 = \{6, 7, 10\}$. This will continue till universal U is totally covered and for this case Output set $O = \{S_1, S_2, S_3\}$.

Basic Greedy Algorithm:

Algorithmus 1 The Greedy Algorithm

```
1: $O \leftarrow \emptyset$
2: while $U \neq 0$ do
3:     for $i = 1 \rightarrow n$ do
4:         Select $S_i$ with max $|S \cap U|$
5:         $O \leftarrow O \cup max(S_i)$
6:         for $j = 1 \rightarrow n$ do
7:             $S_j = S_j - S_i$
8:         end for
9:     end for
10: end while
11: return O
```

Stepwise explanation of Greedy Algorithm:

1. Algorithm starts with an empty set(step 1).
2. When universal set is empty algorithm will check for intersection of universal set with all subsets of X (step 2-4).
3. Subset with maximum coverage is added to the empty set (step 5).
4. Subset added into empty set will be subtracted (step 6-7).
5. Output set is returned.

Target set Selection using Greedy Algorithm

Greedy algorithm deals with NP-hard problem of set covering problem. In this section we will see that how greedy algorithm has been implemented for target set selection. Consider N x N coverage matrix which will be obtained by using algorithms with N as total number of clients and n_{ij} represents element of that matrix with row i and column j. Algorithm will take coverage matrix as input of algorithm and will return target set T.

Algorithm 2 describes that how greedy algorithm is used to find an efficient target set. Algorithm start with an empty target set T. Algorithm considers sum of resultant coverage of each node and coverage is calculated as:

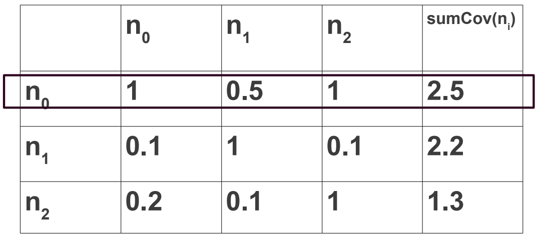
$$sumCov(n_i) = \sum_{1<i<N} n_{ij} \tag{4.1}$$

Algorithmus 2 Greedy Algorithm for Target set selection

```
 1: $T \leftarrow \emptyset$
 2: $TOTAL\_COV[1....N] \leftarrow ADD(n_i j)$
 3: $MAX\_COV = GET\_MAX[TOTAL\_COV[1....N]]$
 4: while $MAX - COV \geq 0$ do
 5:    $n_{max} \leftarrow GET\_ROW(MAX\_COV)$
 6:    $T \leftarrow T \bigcup n_{max}$
 7:    for $i = 1 \rightarrow n$ do
 8:       for $j = 1 \rightarrow n$ do
 9:          if $n_{max_j} > 0$ then
10:             $new_{n_{ij}} = new_{n_{ij}} - n_{maxj}$
11:          end if
12:       end for
13:    end for
14:    $MAX\_COV \leftarrow 0$
15: end while
16: return T
```

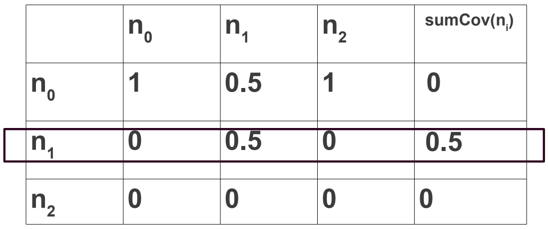
Stepwise Explanation of the algorithm:

1. Algorithm starts with an empty Target set (step 1).
2. An extra column will be added to input matrix. Each cell in that column represents sum of value of all cells of corresponding row (step 2).
3. From last column algorithm will look for cell with maximum resultant sum (step 3).
4. Algorithm checks if maximum coverage is greater than zero(step 4).
5. When coverage is greater than zero the corresponding node will be added to target set(step 6) .
6. All other rows are subtracted from value of corresponding cell and sum is also updated (step 7-12).
7. Maximum value cell obtained in step 3 is changes to zero(step 14).
8. Algorithm continues till all values of columns added in step 2 are zero.



	n_0	n_1	n_2	sumCov(n_i)
n_0	1	0.5	1	2.5
n_1	0.1	1	0.1	2.2
n_2	0.2	0.1	1	1.3

(b) First Step



	n_0	n_1	n_2	sumCov(n_i)
n_0	1	0.5	1	0
n_1	0	0.5	0	0.5
n_2	0	0	0	0

(d) Second Step

Figure 4.5: Coverage Matrix Calculation.

Above example shows that how exactly greedy algorithm will be executed to generate target set. First Algorithm will look for row with maximum sum so it will start from first i.e row n_0 as it has maximum value for sum. So n_0 will be added to target set as shown in figure 4.5(b) and Elements of that row will be subtracted form corresponding element of rest of the rows and sum of and finally sum of that row will be changed to 0. Then in next step algorithm will run again and then it will go to row n_1 as shown in figure 4.5(d) and then n_1 will be added to target set and output targets et will be [n_0, n_1]

4.5 Device selection Algorithms for mobile traffic offloading

This section gives detailed description of Random algorithm and different approaches which were used to get the Coverage matrix to calculate target set.

4.5.1 Random Algorithm

Random Algorithm Concept

Random algorithm is most basic of all algorithms which are implemented. It is basically used to compare the performance of all other algorithms. Random algorithm randomly decides target set from number of available clients and sends request message to all mobile nodes in that target set via cellular interface. Random algorithm gives random results but surprisingly results obtained form random algorithm are quiet good and comparable to other advance algorithms.

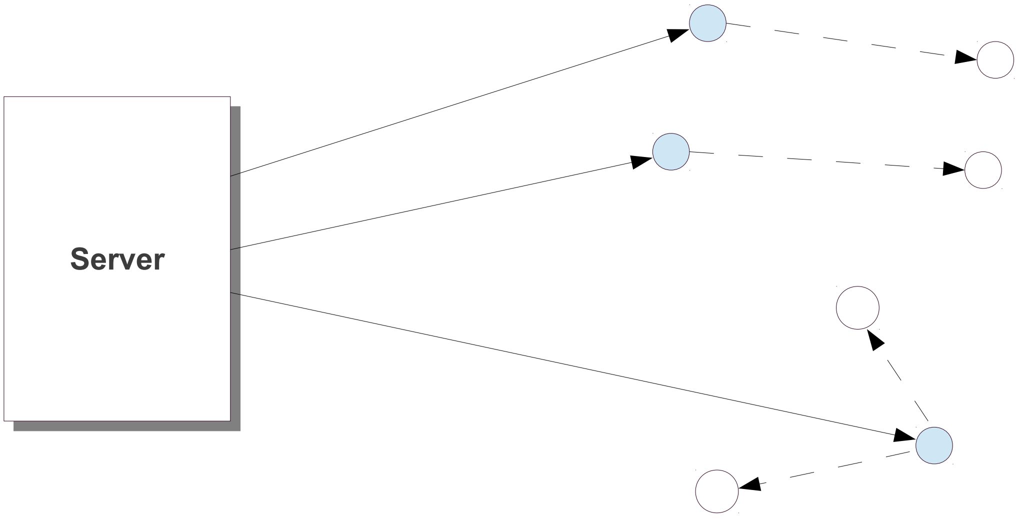


Figure 4.6: Random Algorithm for target set selection

This algorithm follows a unicast approach and treats each mobile node as an individual entity i.e this algorithm does not consider any relation between two mobile nodes. As shown in figure 4.6 the circle represents mobile nodes. Circle in blue colour are the mobile nodes who receive message directly form server via cellular interface and then these blue nodes will forward message to other mobile nodes via Ad-hoc interface, in above figure continuous line represents cellular interface and dotted line represents Ad-hoc interface.

4.5.2 Distance Algorithm

This approach is same as static approach used in TOMP[1].

Distance Algorithm Concept

The coverage in this case is based on Euclidean distance. Coverage is obtained after comparing euclidean distance between the mobile nodes with Ad-hoc range. For each pair of nodes algorithm calculates euclidean distance and compare that distance with Ad-hoc range of the nodes. So if the distance is less than Ad-hoc range then it means that clients can forward message via Ad-hoc communication so probability of coverage is 1, if distance is more than Ad-hoc range then clients can not exchange messages so it is 0.

$$P(coverage) = \begin{cases} 1 & \text{if } dist(A,B) \leq AdHocRange, \\ 0 & \text{else.} \end{cases} \tag{4.2}$$

This approach gives the static coverage of mobile nodes and does not require any information related to speed and future movement, so this algorithm can be used if there is no information available about speed and direction of mobile nodes

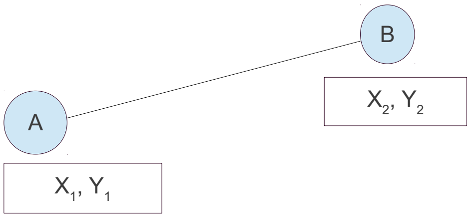


Figure 4.7: Euclidean Distance between two mobile nodes

Figure 4.7 shows two mobile nodes(A and B). Euclidean distance between any two mobile nodes will be calculated as:

$$dist(A,B) = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2}$$

4.5.3 Distance Extended Algorithm

Distance approach considers the probability of coverage as either 0 or 1. It does not consider the case that if there is any chance that clients can cover each other in near future or not.

Distance Extended Concept

Distance Extended Algorithm is basically an advanced version of Distance algorithm. Coverage of mobile nodes is calculated in same way as in distance algorithm. According to Distance Extended Algorithm if clients do not cover each other then probability that client will cover each other in future is considered. Probability that clients will cover each other in future will be inversely proportional to euclidean distance between them.

Covering Probability in future is given as:

$$DE\_Pcoverage = \frac{1}{dist(A, B)} \tag{4.3}$$

where denominator is euclidean distance between any two clients A and B.

$$P(coverage) = \begin{cases} 1 & \text{if } dist(A, B) \leq AdHocRange, \\ DE\_Pcoverage & \text{else.} \end{cases} \tag{4.4}$$

This algorithm is more realistic, as in most of the cases mobile nodes will not be static and an efficient algorithm should always consider the mobility factor of mobile nodes. There is no factor which can assure that Ad-hoc forwarding will be possible between two clients or not. So there will always be an approximation about the Ad-hoc forwarding between mobile nodes. One such approximation is given by this algorithm. Algorithm considers that future covering probability of clients is inversely proportional to the distance between them. So if clients are not in Ad hoc range but are near to each other then it is more likely that Ad-hoc forwarding between them is possible but if mobile nodes are not in Ad-hoc range and are far form each other the it is less likely that there will Ad-hoc forwarding between them.

Scenario In Distance Extended algorithm we are considering the future position of mobile nodes by considering the probability that mobile nodes can exchange message in future according to their current position. Ad-hoc transmission need some time for transmission after the connection is established, so if mobile nodes are moving with very high speed then there is possibility that mobile nodes will move away from each other before transmission is finished. There can be one scenario that limits the scope of this algorithm.

In following figure blue circles represents the mobile nodes added into target set according to distance extended algorithm, white circles are the remaining mobile nodes and big circle around all mobile nodes represents the Ad-hoc proximity of the mobile nodes in blue colour.

Figure 4.8: Position of nodes while calculating Coverage

Consider above figure as situation when coverage was being calculated according to distance extended algorithm, but now consider the case when mobile nodes which are added in target set are moving very fast and they change position before server sent message to that mobile node and new position of the mobile nodes is :

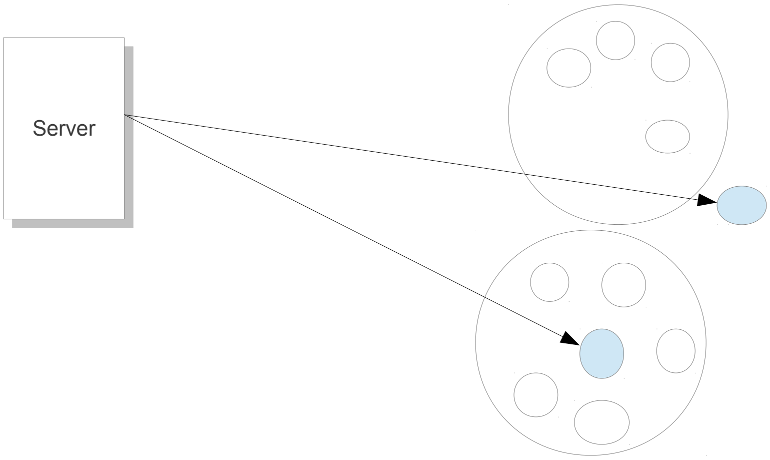


Figure 4.9: New Position of Nodes

As it can be seen that new position of mobile nodes does not satisfy the coverage which was calculated before. Although in the end all nodes will receive message, some mobile nodes will receive message before Panic Zone and some during after Panic Zone. As we discussed before that, this scenario may limit the scope of this algorithmic as in this scenario algorithm will end up in making wrong calculations for coverage of a mobile node.

4.5.4 Same Street Algorithm

Same street algorithm considers the street map and position of mobile nodes on the street map.

Same Street Algorithm Concept

Same Street algorithm first decides that whether clients are on the street or not. Firstly, streets are defined by considering the nodes of the map. If a node on map has more than two neighbouring nodes then it is a junction unless it is a street. Following figure shows the difference between street and junction.

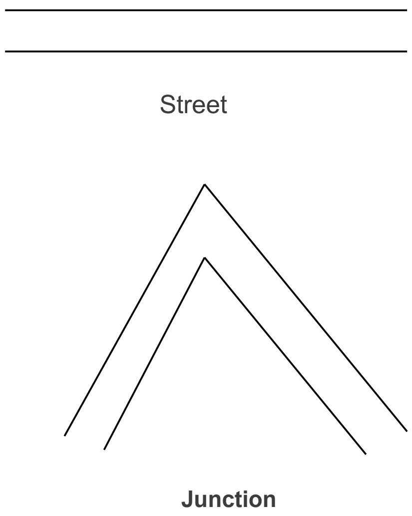


Figure 4.10: Street and Junction

So each client is checked with every other client that whether they are on the same street or not. If clients are on same street then it is considered that they will exchange message. This algorithm is suitable when it is sure that mobile nodes on street will either move towards each other or one mobile node will follow other mobile node.

$$P(coverage) = \begin{cases} 1 & \text{if } SameStreet, \\ 0 & \text{else.} \end{cases} \tag{4.5}$$

Scenario Unlike all other algorithms in this case we are not considering the Ad-hoc range. This will somehow increase the scope of this algorithm. So we consider that when client are on one street and then they can exchange messages.

When mobile nodes are lying on the same street then there can be following three cases:

Case 1. Clients will move towards each other and message will be forwarded as soon as they are in Ad-hoc proximity with each other.

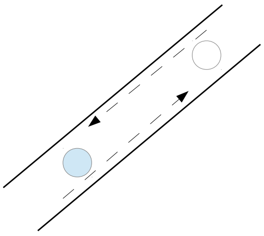


Figure 4.11: Clients moving towards each other

Case 2. One client will follow other client, i.e one client will walk behind another client and vice versa then same as in case 1 message will be forwarded as soon as they are in Ad-hoc proximity with each other.

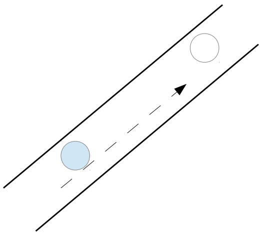


Figure 4.12: One Client is following another Client

Case 3. But there can be one case when clients are moving away form each other, in that case clients will not be able to exchange message even if they are in the same street.

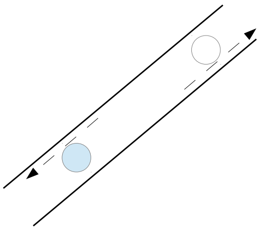


Figure 4.13: Clients moving away from each other

4.5.5 Path Length Algorithm

Path Length algorithm considers clients if they are in one street unless considers the shortest path between them.

Path Length Concept

Path Length Algorithm is enhanced version of Same Street Algorithm. Consider the case when length of street is very small but clients are not in the same street in that case same street algorithm will not give an efficient result, but Path Length algorithm is not limited to one street it is extended to shortest path between two clients. Firstly path between two clients is calculated using Dijkstra's Algorithm[25].

Figure 4.14: Shortest Path between two clients

As shown in figure that there are two paths between the mobile nodes longer path is represented by brown colour and shorter in black. Algorithm will consider the black path as it is the shortest path between two clients. Elements of coverage matrix are obtained after comparing length of shortest path with Ad-hoc range. If clients are not covered then probability of coverage in future is considered. Probability of coverage in future is defined as:

$$PL\_Pcoverage = \frac{1}{PathLength(A, B)} \tag{4.6}$$

Where denominator is the length of shortest path between any two client. So when clients do not cover each other then this algorithm considers the probability that they will cover each other in future. Probability of covering each other in future is inversely proportional to length of shortest path between them.

$$P(coverage) = \begin{cases} 1 & \text{if } PathLength(A, B) <= AdHocRange, \\ PL\_Pcoverage & \text{else.} \end{cases} \tag{4.7}$$

Chapter 5

Implementation

We discussed the system model in Chapter 3 and all algorithms in Chapter 4. In this chapter we will discuss that how we implemented our algorithms on the System Model. We have discussed that we have done Small Scale Implementation using android platform and Large Scale Implementation using The ONE simulator. In this chapter we will discuss about android implementation which also includes a brief description about advantages of android operating system. This chapter also contains description about simulation environment for the Large scale implementation which was done using The ONE simulator and some important information about that simulator.

5.1 Testing Environment For Android Implementation

This task was done to actualize the concept of Mobile traffic offloading as explained in previous chapter. Task was implemented using Desktop and two android based smart phones. In this section we will first discuss about the android platform and then we will discuss the implementation architecture.

5.1.1 Android

Android is a Linux based operating system for smart device like mobile phone and tablets with Java programming interface. It allows user to develop and implement different applications. Android SDK which is android's development kit, it provides all necessary tool required to develop an android application.

There are many other factors that makes Android a developer friendly operating system[27]:

- Android is open source and has liberal licensing, so developers can make full use of an android based handset.

- There is a large group of enthusiastic developers who are continuously developing innovative application, so there is a good scope of exchanging ideas and assistance.
- Android is truly an open operating system. It allows developer to access many internal application such as; developer can develop an application to send message, make calls, access camera, access bluetooth etc.

5.1.2 Implementation Architecture

As discussed before our Android implementation has three main components server and two clients. FTDesktopServer and two MainActivity classes are main classes of the implementation. Our server class is FTDesktopServer. Server class will always be listening to its android client to establish Wi-Fi connection with first android phone (AndroidSenderReceiver) and send messages to first android phone via Wi-Fi interface. This class also contains Java timers these timer calculate the time taken for message transmission form server to client. This first android phone will also act as server for second android phone (AndroidReceiver). First android phone will initiate bluetooth connection and send same message to second android phone via bluetooth interface. As you can see in following diagram that both androids have a class named as MainActivity. Every Android application has an activity class which contains different steps of lifecycle of an android application and connects all parts of an application together so for our android implementation both MainActivity classes are serving this purpose.

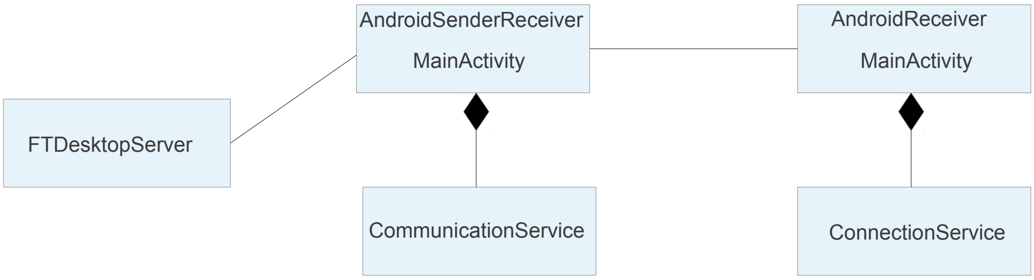


Figure 5.1: Android Implementation Class Diagram

CommunicationService This class is connected to MainActivity class of first android phone(AndroidSenderReceiver). This class contains the IP address of the server. It will initiate Wi-fi connection with server which is already listening. This class has one more characteristic, as we mentioned before that first android phone acts as sever for second android phone. So this class also listens as server to the client which is second android phone

(AndroidReceiver) and forward message received from FTDesktopServer via bluetooth. Two android phones identify each other using MAC address. Another important task of CommunicationService class is to calculate the time taken for message transmission from server to client via bluetooth interface.

ConnectionService This class is connected to MainActivity class of second android phone(AndroidReceiver). This class contains the MAC address of the server i.e first android phone. It will initiate bluetooth connection with server which is already listening.

5.2 Simulation Environment For Large Scale Implementation

The algorithms which were discussed in last chapter were required to be implemented on real time cellular network. Designing and analysing a real time environment can be very expensive and time consuming. So it is always advisable to model real time environment using simulation tools, simulation tools allows you to test real time components in real time environment in a less expensive way. Many network simulators are available such as OPNET, Ns-2 etc.

Our requirement was a delay tolerant network simulator with enabled cellular and Ad-hoc modes communication. To fulfil our requirements we have used The ONE (Opportunistic Network Environment) simulator[24]. The ONE is Java based network simulator that simulates delay tolerant networks[23]. It is discrete event driven simulator. The ONE can simulate moving nodes, inter node communication and routing messages between nodes. It allows user to simulate many real time movement scenarios which are similar to real world. Performance of The ONE simulator depends on many factors like density of mobile nodes, distance between sender and receiver mobile node. Results obtained by different implementations can be visualize form Graphical User Interface(GUI).

5.2.1 The ONE Architecture

Internal architecture of The ONE simulator makes it an agent based simulation engine. The wireless mobile nodes act as agents. There can be any number of agents, all agents share some common parameters such as communication range, message size etc.

The One has a complex internal architecture, in next parts of this section we will try to understand the architecture of simulator, in order to reduce complexity we will only discuss about the parts of its architecture which were important for our implementation.

Basic architecture of The ONE Simulator[22].

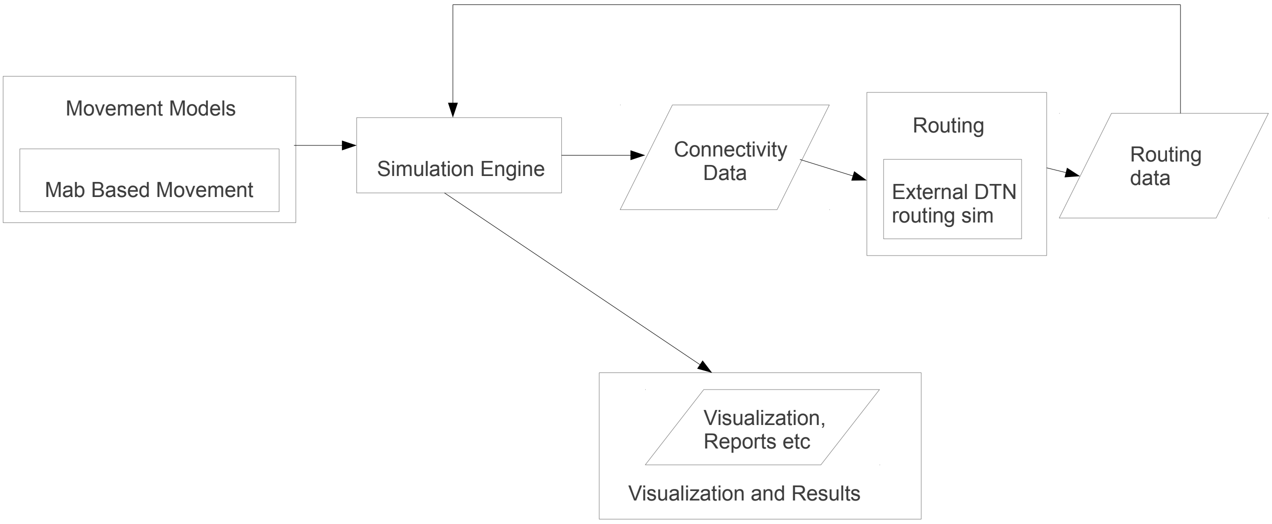


Figure 5.2: The ONE internal architecture

Movement Models

Movement models mean that how nodes will move during simulation. The ONE has many movement models such as Random Movement and Map Based Movement. For our approach we have used map based movement models.

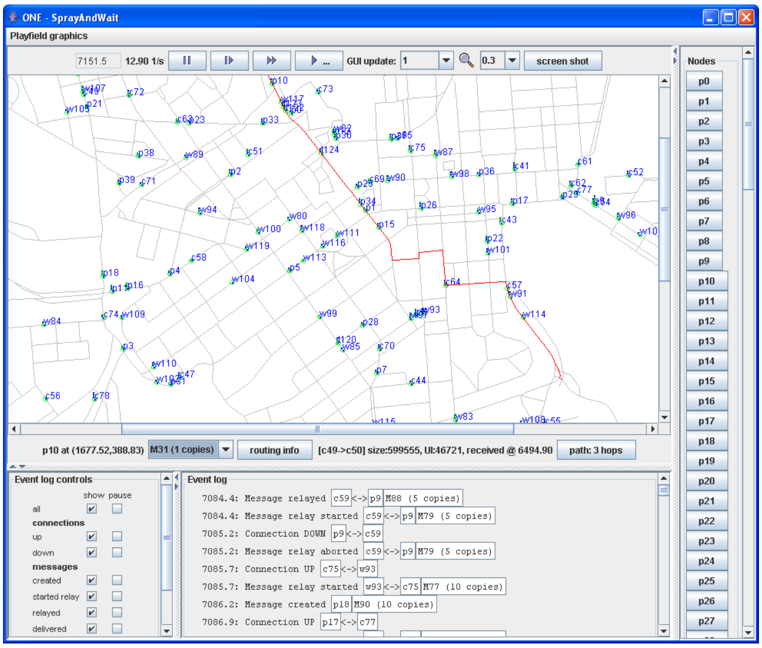
Map Based Movement Models Map based movement models first fetch map data and use that data to decide the movement of the nodes. In our approach we use shortest path based map models so first simulator will place nodes on the random position on the map then destination node will be decided by the algorithms discussed in previous chapter and path between source node and destination node will be the shortest path between them according to Dijkstra's Algorithm[25].

Routing

Routing means that how simulator will route messages during simulation. As we discussed before that simulator is delay tolerant so there are special routing schemes for DTN (Delay Tolerant Network). Simulator does not assures that message will be delivered immediately, When a node A meets another node B and there is chance to exchange messages then first node A checks that whether the message is meant for node B or not if yes then message is forwarded to node B. There are many routing modules available in simulator, for our approach we are using Epidemic Routing[26].

Visualization and Results

The ONE simulator allows user to get result in Batch mode as well as in user interactive GUI. Following figure shows that how GUI of simulator looks like.



GUI update: 1

0.3

p10 at (1677.52,388.83)

[c49->c50] size:599555, UI:46721, received @ 6494.90

	show	pause
all	☑	☐
connections		
up	☑	☐
down	☑	☐
messages		
created	☑	☐
started relay	☑	☐
relayed	☑	☐
delivered	☑	☐

```
7084.4: Message relayed c59<->p9 M88 (5 copies)
7084.4: Message relay started c59<->p9 M79 (5 copies)
7085.2: Connection DOWN p9<->c59
7085.2: Message relay aborted c59<->p9 M79 (5 copies)
7085.7: Connection UP c75<->w93
7085.7: Message relay started w93<->c75 M77 (10 copies)
7086.2: Message created p18 M90 (10 copies)
7086.9: Connection UP p17<->c77
```

Figure 5.3: Graphical user interface of The ONE simulator

Checking results in GUI can be useful testing and debugging purposes and exact values of different parameters can be obtained in batch mode.

5.3 Implementation Architecture

In previous section we discussed about the Java based The ONE simulator. In this section we will discuss the architecture of different *Device Selection Algorithms* which were implemented on the simulator. We will have a brief look at the all Java classes which were define for implementation. The main two classes are ServerApplication and ClientApplication. For every algorithm there will be one server and one or more than one clients. For our algorithms we have used clients from 100 to 900. We will first discuss the basic architecture of the implementation. Then we will discuss about the architecture of the Random Algorithm and other device selection algorithms (Distance Algorithm, Distance Extended

Algorithm, Path Length Algorithm and Same street Algorithm) as all other algorithms have same architecture.

5.3.1 Basic Architecture

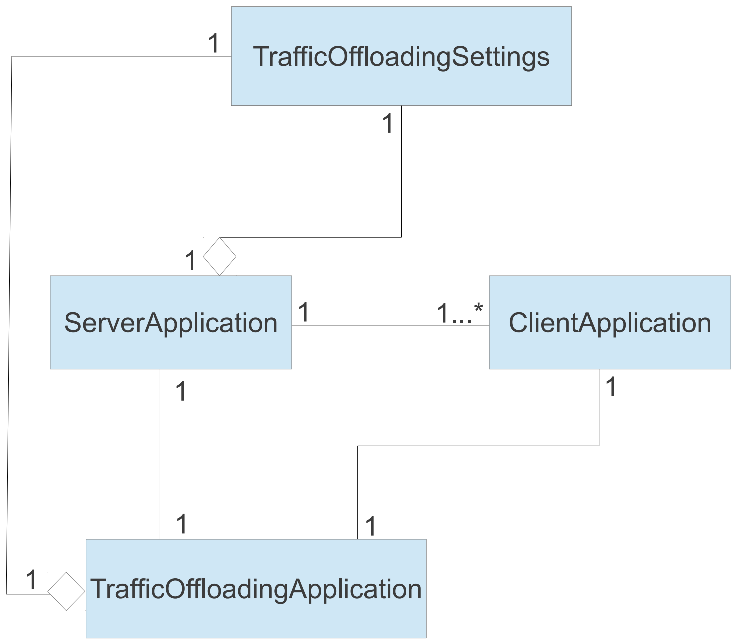


Figure 5.4: Basic Architecture

Above figure shows the basic architecture of our algorithms in the form class diagram. These four classes are the main classes of the implementation architecture.

SeverApplication This is the server application class. As described before that every algorithm has one server. Function of SeverApplication is to send messages to clients, receive acknowledgements and keeping the record that which clients sent acknowledgement and which clients did not.

ClientApplication This is the client application class. Simulator can have maximum of 1200 clients and this class will run for all selected clients. For our measurements we have selected clients varying from range 100 to 900. This class sends initial location updates to server and sends acknowledgements after message is received successfully.

TrafficOffloadingSettings This class stores all necessary information which is defined by the settings file. These settings will be used for different algorithms.

TrafficOffloadingApplication This class is backbone of our architecture. It handles the traffic offloading on server by handing different messages such as data messages, acknowledgements and location updates. It handle many lists such as list of clients to whom server will send message, list of clients who has sent acknowledgements and list of clients who did not send acknowledgement. This class also initiate the creation of data message.

5.3.2 Random Algorithm Architecture

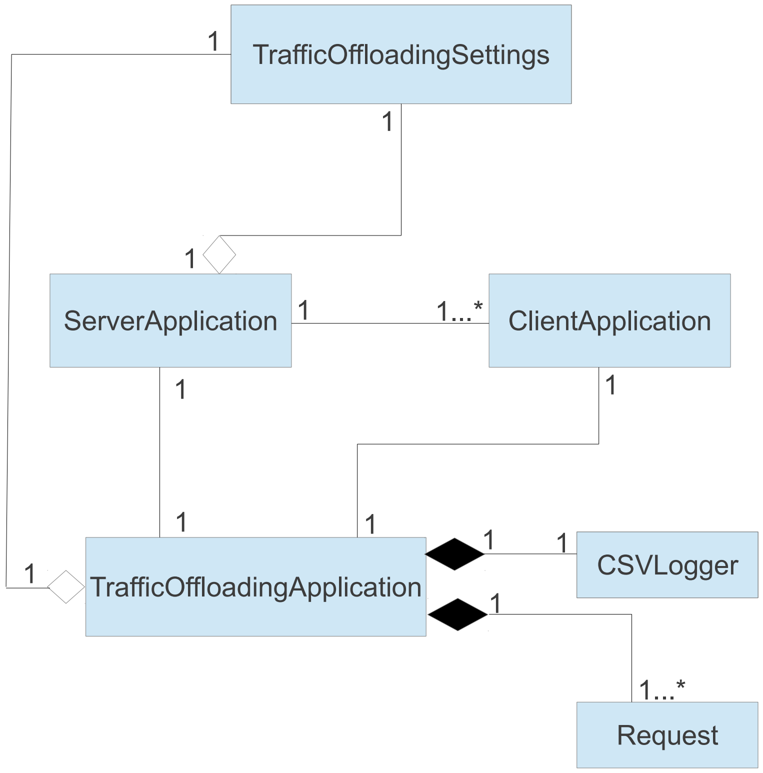


Figure 5.5: Random Algorithm Architecture

Random algorithm is the first basic algorithm that is implemented on the simulator. This algorithm does not have complex architecture as this algorithm does not require any target set selection strategy. This algorithms sends message to randomly decided targets. There are two classes which are linked to TrafficOffloadingApplication and are not part of basic architecture.

Request This class creates the data message which will be transmitted by server to the clients. This class also creates list of clients who has sent acknowledgements and list of clients who did not send acknowledgement. Lists are modified after acknowledgement is received.

CSVLogger This class prints the values of the simulation results as comma-separated-values on the console.

5.3.3 Device Selection Algorithm Architecture

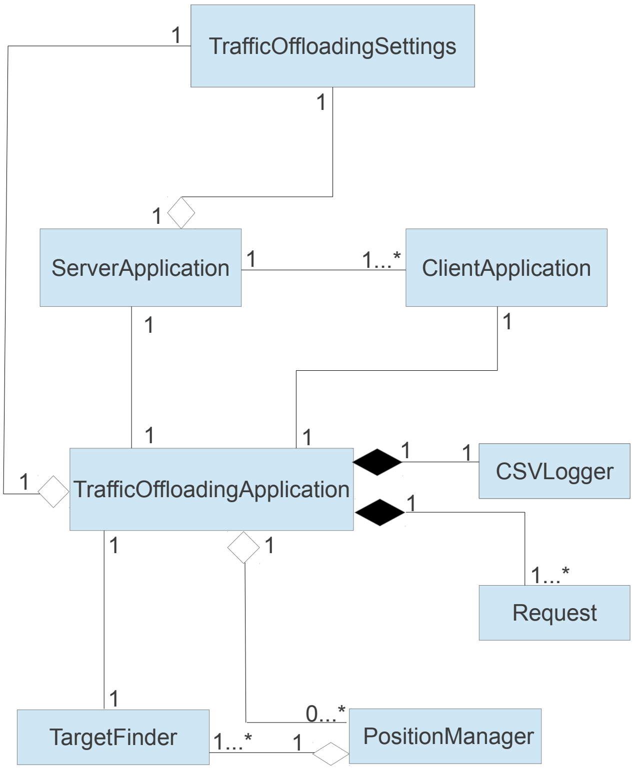


Figure 5.6: Device Selection Algorithm Architecture

This is the common architecture for all device selection algorithms. Now ServerApplication class will send message to clients in the target set which will be decided by one of the implemented algorithm.

TargetFinder This class is very important class of the architecture. All this class finds the target set with the help of device selection algorithms. All device selection algorithms such as Distance Algorithm, Distance Extended Algorithm, Pathlength Algorithm and SameStreet Algorithm are defined in this class. It decides the target set of mobile nodes.

PositionManager This class keep track of current position of the mobile nodes.

5.4 Summary

The complete architecture of Small Scale Implementation as well as Large Scale Implementation was discussed in this chapter. We discussed about server and client application classes of the small scale android implementation. We also discussed about the implementation architecture of The ONE simulator. We discussed the architecture for Random algorithm and other Device Selection algorithms. Random Algorithm does not need any strategy to decide target set so there is no TargetFinder class for random algorithm. Other Algorithms; Distance, Distance Extended, Same Street and Path Length algorithm have different strategies for the selection of target set which are implemented in TargetFinder class.

Chapter 6

Evaluation

In previous chapters we discussed the concepts related to Mobile Traffic Offloading and our Device Selection Algorithm for Mobile Traffic Offloading. Now we will discuss the results obtained from these algorithms and then we will compare those results using different comparison parameters. Initially in this chapter we will discuss and compare the results obtained from Small Scale Implementation on Android platform. Then we will discuss and compare the results obtained from Large scale implementation which was done on The ONE simulator.

6.1 Results For Small scale Implementation

Set up for this Small Scale Implementation using two android based smart phones and desktop has been explained in chapter 3. Now we will compare the results obtained from this simulation. We have obtained results under Wi-Fi interface and UMTS mobile interface. Our Goal is to measure the change in message delay with change in message size and distance between two mobile nodes. We have measured message delay under mobile interface and and under Ad-hoc interface. For mobile interface we did measurements under Wi-Fi interface and under UMTS interface and Bluetooth was used for Ad-hoc mode of communication. We also changed the distance between two mobile nodes to add mobility factor with our measurements. In next parts of this section we will discuss the results obtained after our measurements. First we will discuss the measurements under mobile interface(Wi-Fi and UMTS) then we will discuss the results obtained under Ad-hoc interface and finally we will compare our results.

Areas of Interest Our area of interest is to check the change in message delay time on changing the message size and distance between two android phone. We will check the message delay under mobile interface and Ad-hoc interface.

Message Delay Message delay is expected to increase with increase in message size. Messages of four different sizes were used for measurements.

Measurement Specifications In our measurement graphs x-axis represents the messages of different size in bytes and y-axis represents message delay in seconds. Three different measurements signifies the change in distances between the two mobile nodes. We have measurement for the distance 1 meter, 3 meter and 5 meter.

6.1.1 Results Under Wi-Fi Interface

Message Delay under Wi-Fi interface

Following figure shows the change in message delay with change in message size an distance between the mobile nodes under Wi-Fi interface. As per our expectation there is continuous increase in message delay with increase in message size.

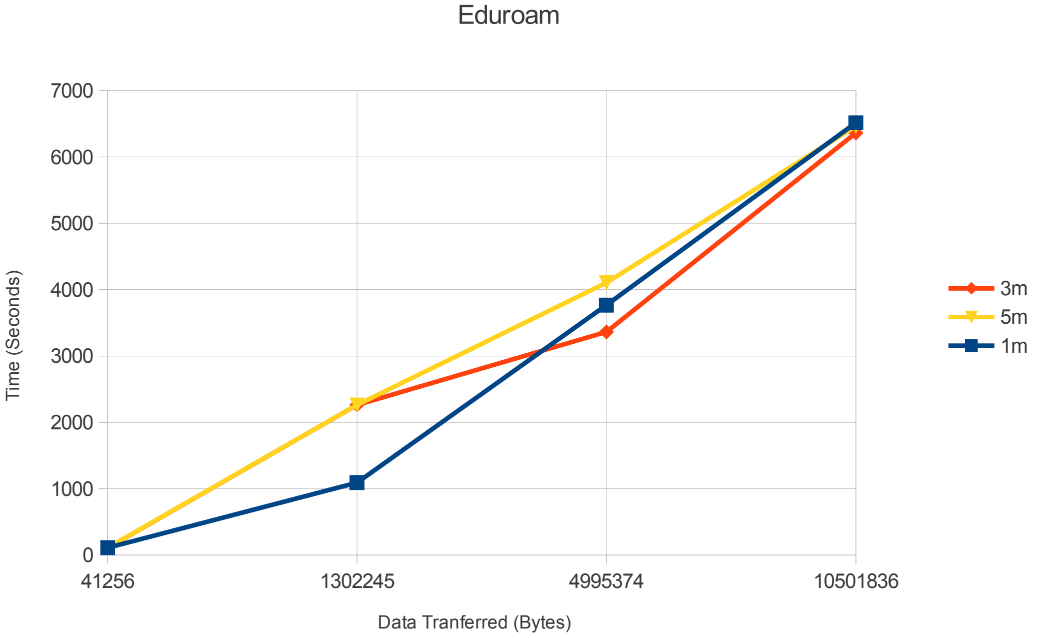


Figure 6.1: Message Delay under Wi-Fi(Eduroam) interface

6.1.2 Results Under UMTS Mobile Interface

Message Delay under UMTS Mobile Interface

In previous part of this section we discussed the results obtained under Wi-Fi interface. For next measurements we used same setup but we changed the mobile interface. Now server will send message to first android phone via UMTS mobile interface following figure shows the message delay resulted under UMTS mobile interface.

Following figure shows the change in message delay with change in message size an distance between the mobile nodes under UMTS mobile interface. As per our expectation there is continuous increase in message delay with increase in message size.

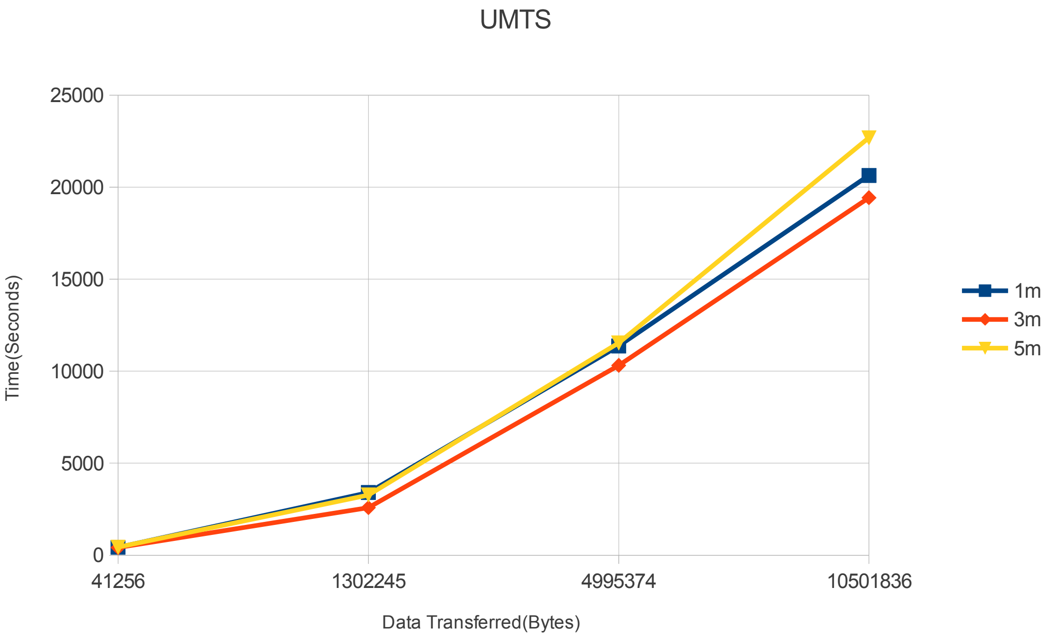


Figure 6.2: Message Delay under UMTS interface

6.1.3 Message Delay under Bluetooth interface

Above figures shows the comparison between the message delay when message was transmitted via mobile interface(Wi-Fi and UMTS). For both of the cases explained above message is transmitted via Ad-hoc interface after it is received successfully via mobile interface. We used bluetooth for Ad-hoc mode of communication.

Figure shows that there is continuous increase in message delay with increase in message size.But There is some some amount of randomness which is associated with Ad-hoc mode of communication. This randomness can be seen in graph for the message of size 4995374 bytes. This randomness remained same while taking measurements for Wi-Fi as well as for UMTS mobile interface. If we compare the message delay via mobile interface with message delay with Ad-hoc interface then we will find that message delay is less in case of Ad-hoc interface.

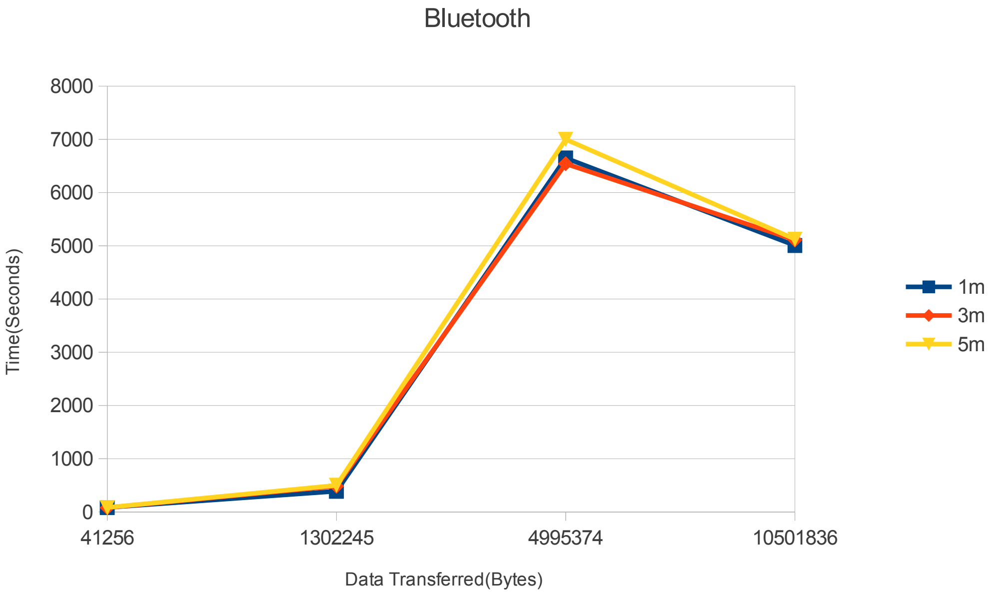


Figure 6.3: Message Delay under Bluetooth interface

6.1.4 Comparison

Following figure shows the comparison of message delay when message was transmitted via UMTS mobile interface and message delay when message was transmitted via Wi-Fi interface.

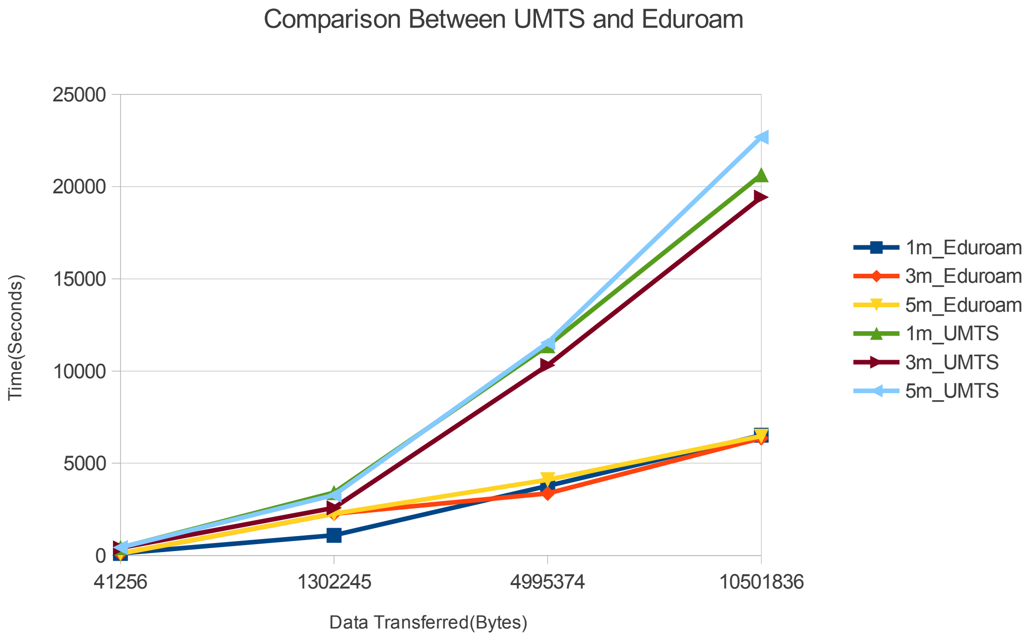


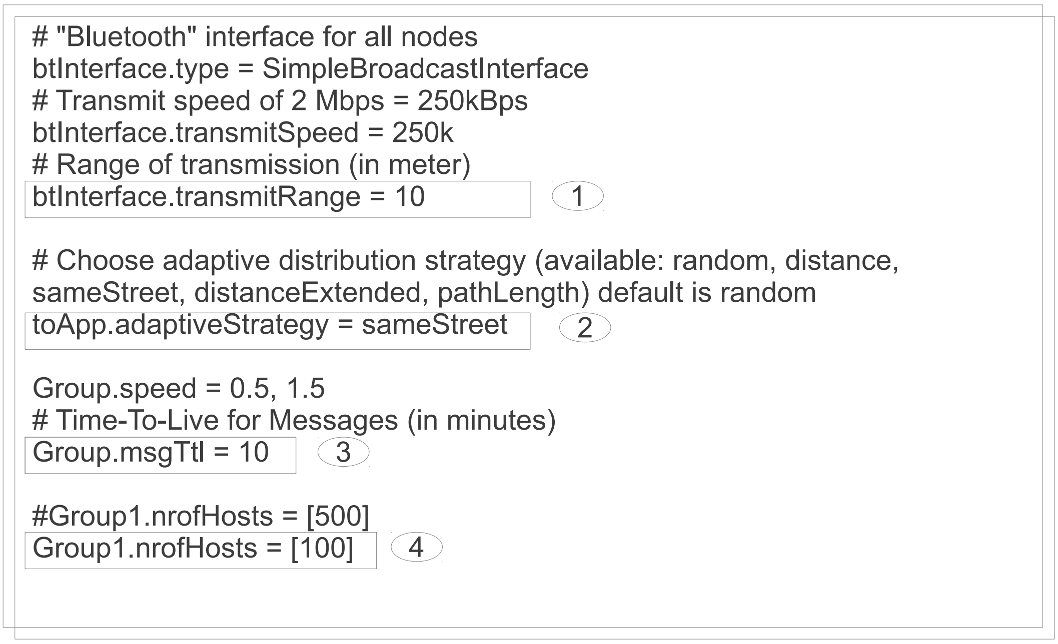
Figure 6.4: Message Delay under UMTS and Wi-Fi(Eduroam) interface

6.2 Results for Large Scale Implementation

Large Scale implementation was done on Java based The ONE simulator. We have done brief description about simulation setup and simulator in Chapter 3 and Chapter 4. In this section we will discuss and compare the results obtained after simulation. Results discussed in previous section are for small scale setup with two mobile nodes bur for this set set we will be using maximum of 900 mobile nodes. We have implemented device selection algorithms for mobile traffic offloading. Main aim of our algorithms is to reduce the traffic from the mobile network by increasing the number of Ad-hoc messages.

Now we will discuss about the settings file which contains many important settings related to simulator.

Settings File: Simulator contains a settings file named as *TrafficOffloading_settings.txt* this file provides all necessary parameters which are required before starting simulation. Following figure shows a brief view of settings file. File includes many other settings related to simulator but in order to reduce the complexity we will only discuss about the settings which are important for our simulations.



```
# "Bluetooth" interface for all nodes
btInterface.type = SimpleBroadcastInterface
# Transmit speed of 2 Mbps = 250kBps
btInterface.transmitSpeed = 250k
# Range of transmission (in meter)
btInterface.transmitRange = 10

# Choose adaptive distribution strategy (available: random, distance,
sameStreet, distanceExtended, pathLength) default is random
toApp.adaptiveStrategy = sameStreet

Group.speed = 0.5, 1.5
# Time-To-Live for Messages (in minutes)
Group.msgTtl = 10

#Group1.nrofHosts = [500]
Group1.nrofHosts = [100]
```

Figure 6.5: Simulation Settings

Brief description about the Simulation Settings:

1. It is communication range for Ad-hoc mode of communication in meters. 10 m is the typical range for Bluetooth. We have also done measurements with Ad-hoc range of Wi-fi Direct which has maximum range of 150 meters.

2. We have implemented in five algorithms on the simulator. Algorithm must be specified in the settings file in order to obtain the corresponding results. Random algorithm is consider as default algorithm.

3. This settings signifies the time to live for message. It is the expiration time for the message. Message must be delivered before its expiration time.

4. This setting allows user to change number of mobile nodes. Simulator has maximum of 1200 mobile nodes. For our measurements we have used mobile nodes ranging from 100 to 900.

6.2.1 Comparison Parameters

We have done simulations with Ad-hoc range of 10 meters which corresponds Bluetooth Ad-hoc range and 100 meters which corresponds Ad-hoc range of Wi-Fi Direct. In next parts of this section we will first discuss the results with Bluetooth Ad-hoc range and then with Wi-Fi Direct Ad-hoc range.

Comparison of measurements will be done on the basis of certain parameters. Now we will discuss about all those parameters.

1 Ad-hoc Coverage:

Ad-hoc coverage is a very important parameter for our results. As we discussed before that main aim of our work is to reduce the traffic in mobile network by shifting some amount of traffic to Ad-hoc network. So Ad-hoc coverage means number of Ad-hoc message as compare to total number of message which were sent during simulations.

$$AdhocCoverage = \frac{AdhocForwardings}{TotalMessages} \tag{6.1}$$

2 Message Delay

We have discussed Flowchart representation of Simulator implementation in section 4.3.2. As shown in flowchart that few minutes before message deadline is regarded as Panic Zone also we have discussed in section 1.3 that our approach tolerates delay but we should try to avoid it. One possible way to reduce the message delay is by reducing the messages in Panic Zone.

3 Energy Efficiency

Our approach uses UMTS messages and Ad-hoc messages both messages consume different amount of energy. This energy consumption calculation is done for mobile nodes as they have limited resources. Table 6.1 [30] provides amount of energy consumed by different messages, these values are for message of size 1000 Bits. As shown in table that energy consumption is very high in UMTS messages as compare to Ad-hoc messages. Hence, reduction in total UMTS messages will result in huge reduction in energy consumption. Most of the energy consumption is done due to misuse of communication techniques.

Table 6.1: Energy Model

Message Type	Energy[mj]
UMTS-Send(1000 Bit)	80
UMTS-Receive(1000 Bit)	40
Adhoc-Send(1000 Bit)	2
Adhoc-Receive(1000 Bit)	1

Energy Consumption Calculation Energy calculation is done on the basis of values given in table. We have used different UMTS and Ad-hoc messages in the form of data messages, acknowledgements and location updates. Now we will discuss about all the messages that will consume energy and finally we will describe the equation to calculate energy.

1. **UMTS Message:** These messages will be sent either from server to mobile nodes or from mobile nodes to server via UMTS interface.
   - Location Updates: (*UMTS_lu*) In the beginning of every approach, mobile nodes will send location updates to server in order to update server about their existence and their current location. Messages will be considered as UMTS-Send message from above table as these messages will be sent from mobile nodes to server.
   - Data Messages: (*UMTS_dm*) Messages sent from server to mobile nodes via UMTS interface. Messages will be considered as UMTS-Receive message from above table as these messages will be sent form server and received by mobile nodes.
   - Acknowledgements: (*UMTS_ack*) All mobile nodes will send acknowledgements back to server after successfully receiving the data messages. Messages will be considered as UMTS-Send message from above table as these messages will be sent form mobile nodes and received by server.
2. **Ad-hoc Messages:** Messages forwarded from few mobile nodes to other mobile nodes via Ad-hoc mode of communication. So there will be both Adhoc-Send messages (*Ad-hoc_msg_sent*) and Adhoc-Receive messages (*Ad-hoc_msg_received*) as mentioned in above table.

The total amount of energy consumed for different approaches will be calculated as:

$$\begin{aligned} Totalenergy = UMTS\_lu \times 80 + UMTS\_dm \times 40 + UMTS\_ack \times 80 \\ + Ad - hoc\_msg\_sent \times 2 + Ad - hoc\_msg\_received \times 1 \end{aligned} \quad (6.2)$$

In next part of this section we will compare all the parameters that we have discussed in this section.

6.2.2 Results with Ad-hoc range of Bluetooth

Ad-hoc Coverage Comparison

We have discussed the importance of Ad-hoc coverage in section 6.2.1. Now we will compare the Ad-hoc coverage obtained from different algorithms.

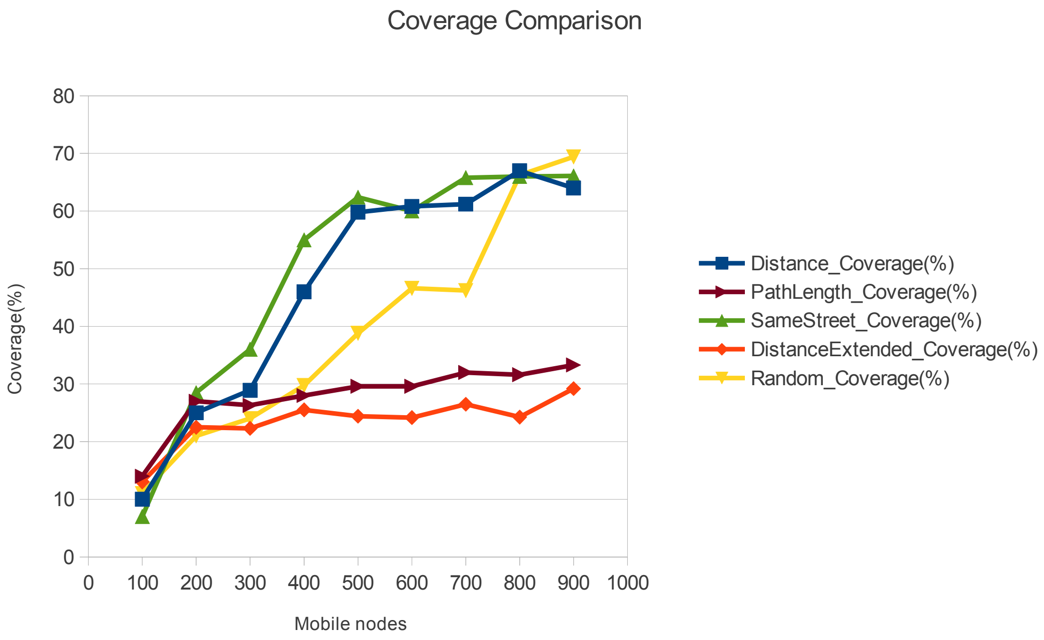


Figure 6.6: Ad-hoc Coverage under Bluetooth Ad-hoc range

Figure shows the results obtained after implementing five different algorithms on the simulator. X-axis represents the numbers of mobile nodes and Y-axis represents the percentage of Ad-hoc coverage. Ad-hoc coverage represents the reduction in data traffic on the mobile network. Our all algorithms aim to reduce the data traffic on the mobile network by increasing number of Ad-hoc messages. Surprisingly Random Algorithm discussed in section 4.5.1 also provides good results. As shown in above figure the SameStreet Algorithm and Distance algorithm are showing excellent results whereas average results are obtained form Distance extended Algorithm and Path Length Algorithm.

This difference in results is due to value of probability which will be considered while calculating the Target set. Distance Algorithm and Same Street Algorithm discussed in section 4.5.2 and section 4.5.4 respectively, value of probability is either 0 or 1. But for Distance Extended and Path Length Algorithm discussed in section 4.5.3 and section 4.5.5 respectively we are considering polynomial value of the probability. This makes our target set selection NP-hard we used Greedy algorithm discussed in section 4.4.1 to avoid this problem. According to Greedy Algorithm mobile node will be added to target set even if it covers atleast one other mobile node. Hence, Distance Extended and Path Length Algorithm results less number of Ad-hoc message. These algorithms give better coverage with large Ad-hoc range which will be discussed later in this section.

Message Delay Comparison

In section 6.2.1 we discussed about the Message delay and how it is important. In section 4.3.2 we discussed about Panic Zone. Delay can be reduced by reducing the number of messages in Panic Zone.

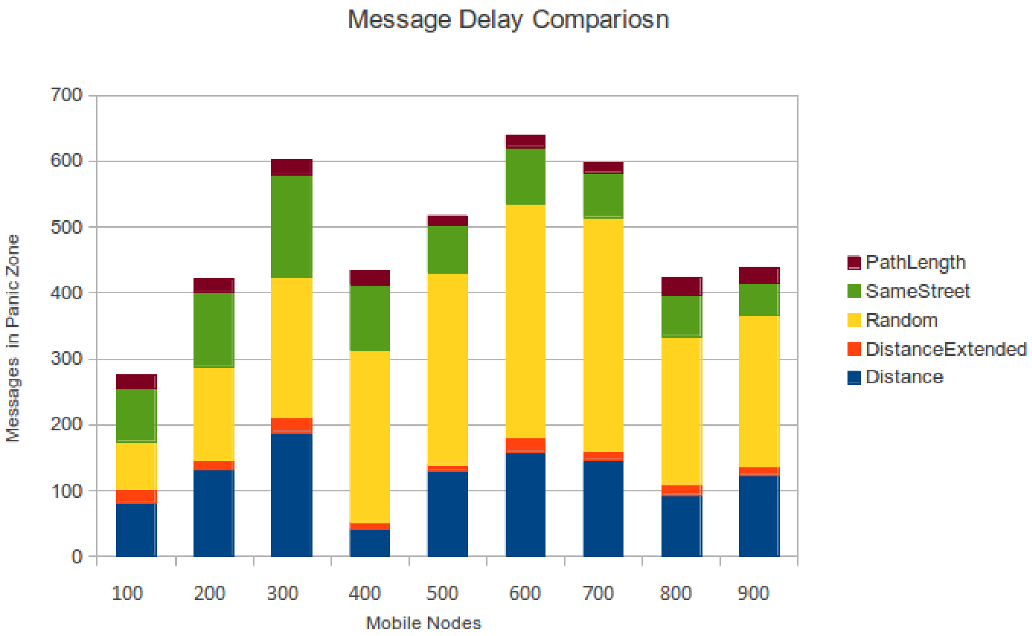


Figure 6.7: Message Delay Comparison

As per our expectation Random algorithm algorithm has highest number of messages in Panic Zone. If we discuss the performance of other algorithms then we will find that

Distance Extended Algorithm and Path Length Algorithm are showing very good performance with very less messages in Panic zone. As we discussed before that in Distance Extended and Path Length algorithm we use polynomial value of probability which makes it complicated but it also makes algorithm more reliable.

Energy Efficiency Comparison

In section 6.21 we discussed about energy consumption and method to calculate the energy consumption for mobile nodes. Following figure compares the energy consumption by mobile nodes for all algorithms.

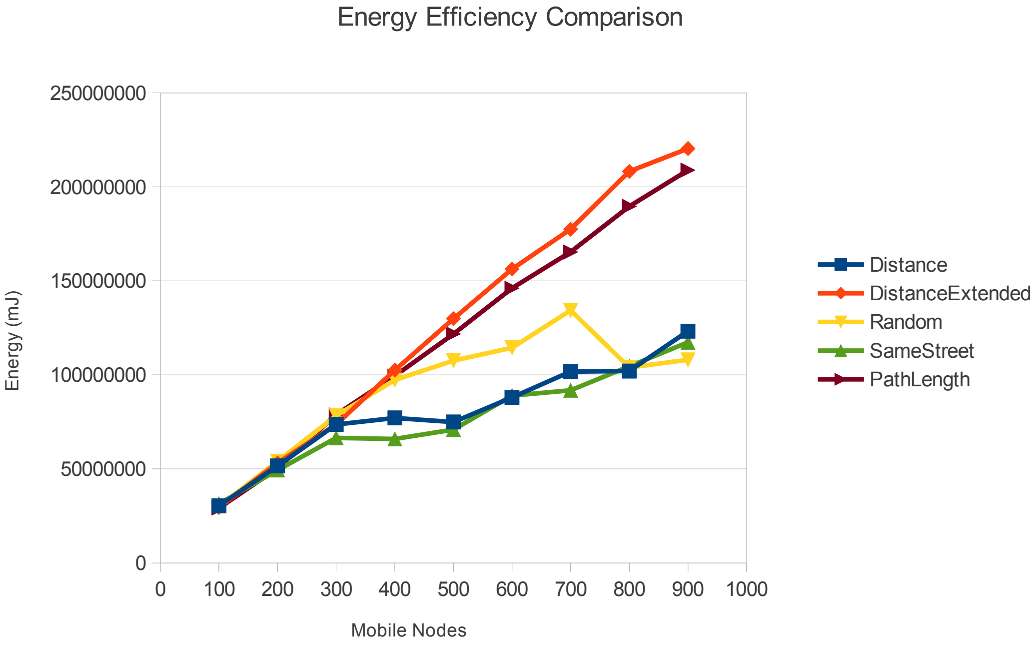


Figure 6.8: Energy Efficiency Comparison

The results obtained in the form of energy consumption are according to our expectation. According to equation(6.2) UMTS messages consume large part of energy. So energy consumption will be directly proportional to UMTS messages. Since Distance Extended Algorithm and Path Length Algorithm has highest number of UMTS messages as shown in figure 6.8. Hence, These algorithms consume more energy as compare to other algorithms.

6.2.3 Results with Ad-hoc range of Wi-Fi Direct

Till now we compared performance of all Device Selection Algorithms for Mobile Traffic Offloading (Distance, Distance Extended, Random, Path Length and Same Street) on the basis of different parameters such as Ad-hoc Coverage, Message Delay and Energy Consumption. We found that Distance Algorithm and Same Street Algorithm are showing very good results as compare to Distance Extended and Path Length Algorithms for Ad-hoc coverage and Energy Efficiency. We also found that Distance Extended and Path Length algorithms are more reliable if we compare the message delay as number of message forwardings are very less in Panic Zone. Now we will discuss the difference in results with Ad-hoc range of Wi-Fi Direct

Ad-hoc Coverage Comparison

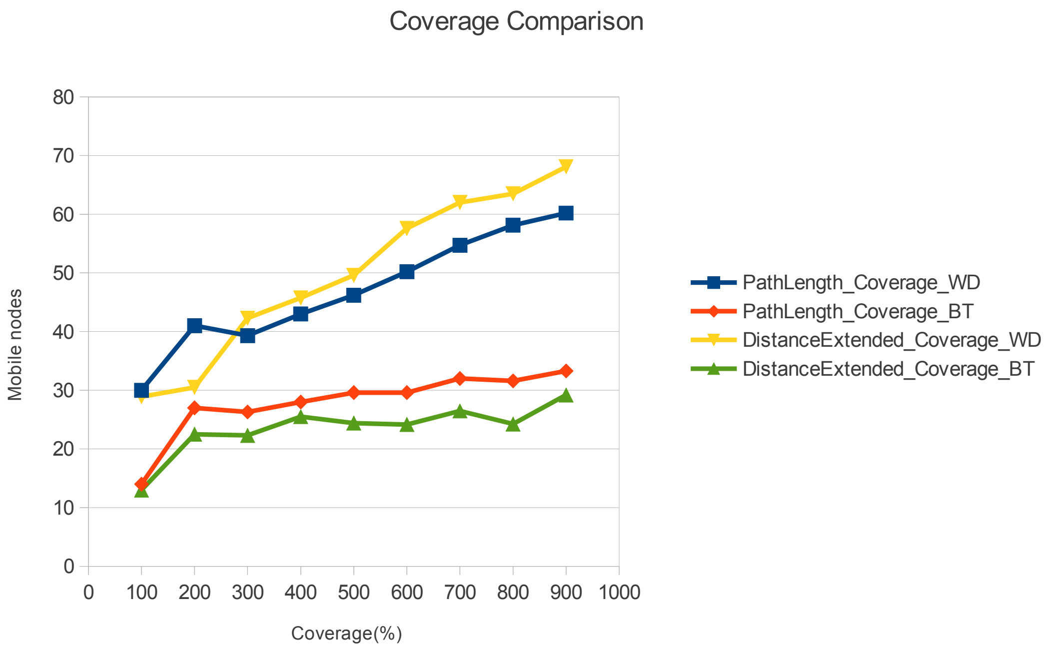


Figure 6.9: Ad-hoc Coverage under Wi-Fi Direct Ad-hoc range

In section 1.3 we discussed that Reliability and Delay tolerance are two important challenges for our algorithms. We found that Path Length and Distance Extended are very reliable and have less message forwarding in Panic Zone but these algorithms have less

Ad-hoc coverage with Bluetooth Ad-hoc range. Above figure shows the results of Distance Extended and Path Length algorithms with Ad-hoc range of Wi-Fi Direct. Wi-Fi Direct can have maximum range of 150 meters, but considering the distortions in realtime network we have done measurements with Ad-hoc range of 100 meters. We will compare the results obtained under the Bluetooth Ad-hoc range and Wi-fi Direct Ad-hoc range.

Figure 6.9 is comparing the results obtained under Ad-hoc range of Wi-Fi Direct (WD) and Bluetooth (BT). The parameter Ad-hoc coverage is same as defined in section 6.2.1. According to our expectation there is drastic improvements in results under the Ad-hoc range of Wi-fi Direct.

Message Delay Comparison

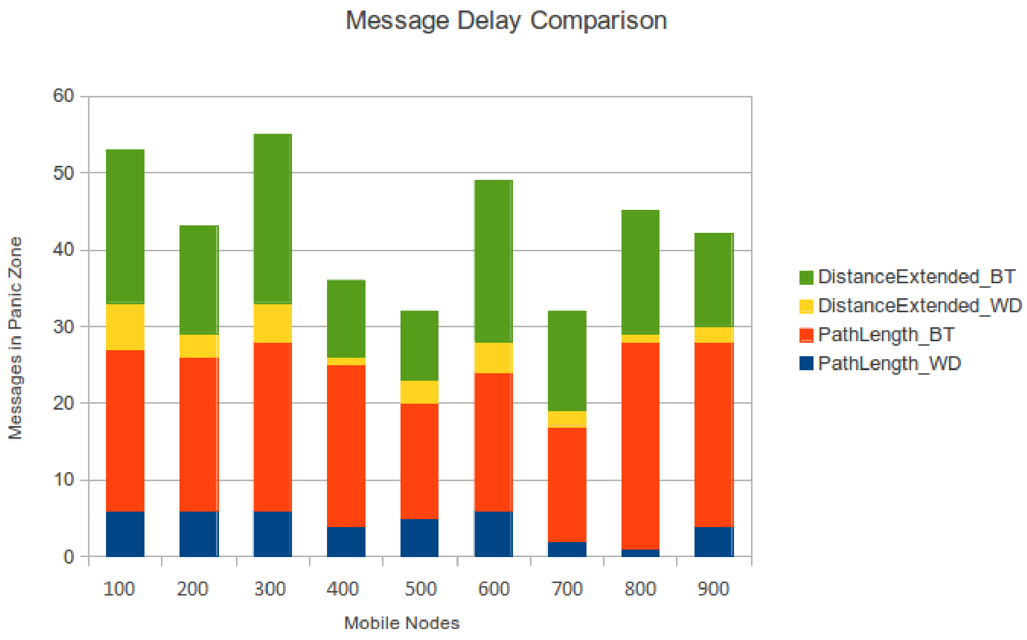


Figure 6.10: Message Delay Comparison under Wi-Fi Direct Ad-hoc range

As we have discussed before that this comparison is done on the basis of number of messages sent in Panic Zone. As shown in figure 6.7 Distance Extended and Path Length Algorithm are already showing good results under Bluetooth Ad-hoc with very messages in Panic Zone.

In figure 6.10 we observed that performance of our Algorithms has become better with further decrease in number of messages in Panic Zone.

Energy Efficiency Comparison

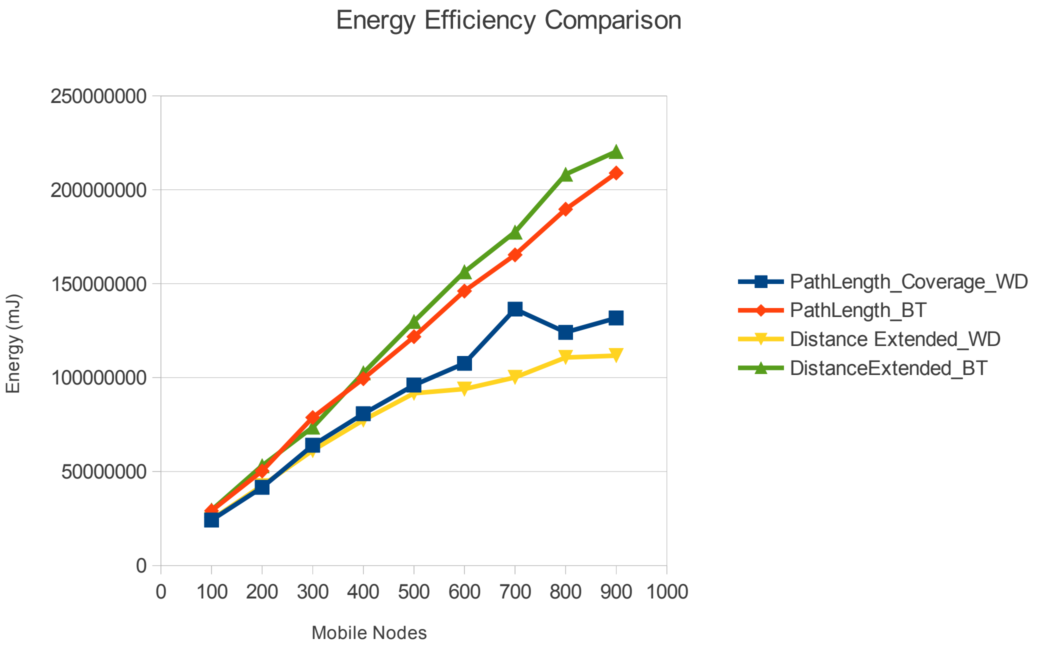


Figure 6.11: Energy Efficiency Comparison under Wi-Fi Direct Ad-hoc range

Figure 6.11 is showing the amount of energy consumption by mobile nodes with Path Length and Distance Extended Algorithm under Ad-hoc range of Bluetooth and Wi-Fi Direct. We have discussed before that energy consumption is directly proportion to number of UMTS messages. In figure 6.9 we have seen that there is huge decrease in number of UMTS messages under Wi-fi Direct Ad-hoc range. Hence, energy consumption is also reduced.

6.3 Summary

In this chapter we discussed different parameters to compare the performance of our algorithms. In section 6.2.1 we defined all comparison parameter(Ad-hoc Coverage, Message

Delay and Energy Efficiency). After comparing all these parameters under Bluetooth Ad-hoc range in section 6.2.2 we concluded that Same Street and Distance algorithms are showing very good Ad-hoc coverage. We also concluded though Path length and Distance Extended algorithms have less Ad-hoc cover but still these algorithms are more reliable with very less messages in Panic Zone.

On considering the reliability factor of Path Length and Distance Extended algorithms which is very important for our approach. We checked the performance of these algorithms with Ad-hoc range of 100 meters which is equivalent to Wi-Fi Direct, we compared these algorithms using same comparison parameters. We also compared the obtained results with Ad-hoc range of Bluetooth and we found that the Path Length and Distance Extended algorithms are showing very good results with Ad-hoc range of Wi-Fi direct.

Chapter 7

Summary and Future Work

7.1 Summary

The thesis deals with algorithms to reduce the data traffic on the mobile network by shifting some amount of traffic to Ad-hoc network. We actualized the concept of Mobile Traffic Offloading by using two Android based smart phones and one server. We also implemented this concept on The ONE simulator. According to our approach mobile network will send data to only few mobile nodes which is called as Target Set of mobile nodes and these mobile nodes will forward data to rest of the mobile nodes via Ad-hoc communication. We have discussed many algorithms to select an efficient target set which can effectively reduce the traffic from mobile network. All algorithms aim to reduce the data traffic by increasing the message forwarding via Ad-hoc modes of communication. .We have implemented total five algorithm; Random Algorithm, Distance Algorithm, Distance Extended Algorithm, Path Length Algorithm and Same Street Algorithm. Every algorithm has used different approach to select target set.

Random Algorithm does not have any logic to select Target Set, so it randomly decides Target Set of mobile nodes. Distance and Distance Extended Algorithms consider the current position of mobile nodes and compare the distance between with Ad-hoc range in order to find the possibility of Ad-hoc communication between them. Distance Algorithm does not consider the future position of mobile nodes but Distance Extended Algorithm considers the probabilistic future position of mobile nodes based on the current position of mobile nodes and checks the possibility of Ad-hoc communication. Same Street Algorithm considers the current position of mobile nodes and checks if mobile nodes are on same street or not, this algorithm does not consider the future position of mobile nodes. So the probability is 1 if mobile nodes are on the same street and 0 if they are not on the same street. Our final Algorithm is Path Length Algorithm. This algorithm considers the shortest path between mobile nodes which is decided on the basis of Dijkstra's Algorithm. Path length Algorithm also considers the shortest path but is advance version of Same

Street Algorithm as it is not limited to one street and also considers the future probability of Ad-hoc communication which is decided on the basis of current shortest path between the mobile nodes. The future probability for Path Length Algorithm and Distance Extended Algorithm is inversely proportional to the current shortest path and distance respectively between the mobile nodes which means that if mobile node are far from each other then there is less probability of Ad-hoc communication between them in future.

All these Algorithms except Random Algorithm will decide the probability of coverage based on their corresponding logic. This coverage values will be treated as NxN matrix where N is number of mobile nodes. Elements of this coverage matrix will have value form range[0,1]. This matrix will be given as input to greedy Algorithm and Target set will be obtained as output of this greedy algorithm. All algorithms assure 100% message delivery rate. Evaluation of algorithms is done on the basis of different comparison parameters such as Ad-hoc coverage, Message Delay and Energy Efficiency. Evaluation of algorithms revealed that algorithms with probability values 0 and 1 (Same Street and Distance) have better results with Bluetooth Ad-hoc range but are less reliable. Evaluation also revealed that Algorithms with polynomial value of probability (Distance Extended and Path Length) are more reliable and give very good results in terms of Ad-hoc coverage and energy consumption by changing the Ad-hoc range from Bluetooth to Wi-Fi Direct.

7.2 Future Work

In our approach we have used coverage matrix (section 4.4.1). Each element of coverage matrix is the probability of Ad-hoc coverage between the mobile nodes of corresponding row and column. Our approach adds mobile node in Target Set (section 1.2.1) if it covers at least one other mobile node. In future it is possible to define a parameter which decides a minimum number of mobile nodes that one mobile node should cover before it is added to the Target Set.

This new parameter can help us to achieve our goal of reducing the number of messages via cellular interface. Adding this new parameter will not affect the delivery rate as all missing nodes will receive message in Panic Zone (section 4.3.2). It is difficult to make any fixed decision about the minimum number of mobile nodes that should be covered by one mobile. The one parameter that can affect our decision is the total number of mobile nodes.

Bibliography

[1] Patrick Baier, Frank Dürr, Kurt Rothermal. TOMP: Opportunistic Traffic Offloading Using Movement Predictions.

[2] http://www.verisi.com/resources/mobile-internet-global-usage.htm //12.March 2013

[3] Lingyang Song, Jia Shen. Evolved Cellular Network Planning and Optimization for UMTS and LTE. //page 365

[4] Aravindhan Venkateswaran. Mobility Prediction Based Intelligent Algorithms for Mobile Ad Hoc Networks // Pages 64-69

[5] [Chandrasekhar08] Chandrasekhar, V. ; Andrews, J. ; Gatherer, A.,FemToCell networks: a survey, Communications Magazine IEEE, 2008, Volume 46 Issue:9, Pages 59-67

[6] Cellular Traffic Offloading through WiFi Networks Savio Dimatteo , Pan Hui, Bo Han , and Victor O.K. Li

[7] An Overview of Opportunistic Communication http://www.scribd.com/doc/40199823/An-Overview-of-Opportunistic-Communication //15. March 2013.

[8] http://www.bluetooth.com/Pages/Bluetooth-Home.aspx // 16.March 2013

[9] http://www.digitaltrends.com/computing/wi-fi-direct-mimics-bluetooth-but-with-faster-speed-longer-range/ //16. March 2013

[10] http://www2.alcatel-lucent.com/blogs/corporate/2012/12/metrocells-101-what-are-they-where-would-you-use-them-and-how-do-you-set-them-up/ // 16. March 2013

[11] Vasilios A. Siris and Dimitrios Kalyvas. Enhancing Mobile Data Offloading With Mobility Prediction and Pre fetching.

[12] C. Kalampokis, D. Kalyvas, I. Latifis, and V.A. Siris. OptiPath: Optimal Route Selection Based on Location Data Collected from Smartphones.

[13] Aruna Balasubramanian,Ratul Mahajan and Arun Venkataramani. Augmenting Mobile 3G Using WiFi.

[14] Marco Valerio Barbera, Julinda Stefa , Aline Carneiro Viana, Marcelo Dias de Amorim, and Mathias Boc. VIP Delegation: Enabling VIPS to offload Data in Wireless Social Mobile Network.

[15] William Su, Sung-Ju Lee and Mario Gerla. MOBILITY PREDICTION IN WIRELESS NETWORKS.

[16] Bo Han, Pan Hui, Member, IEEE, V.S. Anil Kumar. Mobile Data Offloading Through Opportunistic Communication and Social Participation.

[17] David Kebo Houngninou(Under guidance of Prof. Raj Jain). Femtocell: Indoor Cellular Communication Redefined.

[18] S. Kashihara, M. Y. Sanadidi and M. Gerla. Mobile, Personal Data Offloading to public Transport Vehicle.

[19] Yong Li, Guolong Su, Pan Hui, Depeng Jin, Li Su and Lieguang Zeng and Multiple mobile data offloading through delay tolerant networks.

[20] Kanchana Thilakarathna, Aline Carneiro Viana, Aruna Seneviratne and Henrik Petander. The Power of Hood Friendship for Opportunistic Content Dissemination in Mobile Social Networks.

[21] http://www.itu.int/ITU-D/ict/statistics/ // 19. March 2013

[22] Ari Keränen, Jőrg Ott and Teemu Kärkkäinen. The ONE simulator for DTN protocol evaluation.

[23] http://delay-tolerant-networks.blogspot.de/p/one-tutorial.html // 29. March 2013

[24] http://www.netlab.tkk.fi/tutkimus/dtn/theone/ // 30. March 2013

[25] http://www.cs.sunysb.edu/ skiena/combinatorica/animations/dijkstra.html

[26] Alan Dcmers. Dan, Greene, Carl Hauser, Wes Irish, John. Larson. Scott Shenkcr, Howard Sturgis, Dan Swinehart and Doug Terry EPIDEMIC ALGORITHMS FOR REPLICATED DATABASE MAINTENANCE.

[27] http://www.openhandsetalliance.com/androidoverview.html //24.03.2013

[28] http://developer.android.com/training/basics/activity-lifecycle/starting.html //24.03.2013

[29] John Whitbeck, Yoann Lopez1, Jer emie Leguay, Vania Conan and Marcelo Dias de AmorimRelieving the Wireless Infrastructure: When Opportunistic Networks Meet Guaranteed Delays.

[30] H.Weinschrott , F.Dürr and K. Rothermal. Efficient Capturing of environmental data with mobile rfid readers. In mobile data Management.Pages 41-50,2009.