Master's thesis

Two years

Huvudområde
M.Sc. Thesis
within Computer Engineering

Title
Scalable Device Mobility – Mobile DCXP

Author: **Ishfaq Hussain**

Campus Härnösand Universitetsbacken 1, SE-871 88. **Campus Sundsvall** Holmgatan 10, SE-851 70 Sundsvall.
Campus Östersund Kunskapens väg 8, SE-831 25 Östersund.
Phone: +46 (0)771 97 50 00, Fax: +46 (0)771 97 50 01.

Abstract

The continuously increasing Internet coverage and its availability has give rise to an issue that was once considered not important to take into consideration. Today a number of applications use the Internet to deliver time critical messages. The usage of wireless Access Points involves a considerable percentage to connect mobile devices to the Internet provider. However, these relatively cheaper Internet Access Points have their own disadvantages as compared to the GSM and ADSL. The access points cover a very limited area and thus in order to cover a wider area multiple access points must needs to be installed. In other words, as the user moves he/she is supposed to switch between access points. Nevertheless, the basic problem in such cases is involves packet loss during handover. In today's technological advancements these issues, though very small, are no more insignificant but are required to be handled properly. So protocols such as MobileIP, LISP, HOST have been proposed and are currently being used for such a purpose. Furthermore, in this thesis a mechanism to reduce such packet losses has been studied and proposed in relation to the SensibleThings Internet-of-things platform. A workaround solution known as Mobile DCXP has been proposed and implemented and comparisons with the existing system have been carried out. In addition, a generic solution has been discussed in detail and compared with the Mobile DCXP. However, the implementation of the generic solution has been deferred to the future. The concept of *Mobile DCXP has been illustrated with* proof-of-concept apps and an implementation of a simple Android Application known as IChat has been conduct. The IChat is a simple chat app that is used in the experiment to determine out the packet lost during handover and to carry out a comparison. Finally, based on the data collected from IChat, an evaluation of Mobile DCXP has been presented and a performance comparison with Mobile Proxy DCXP has been illustrated with charts. Furthermore, in the conclusion Mobile DCXP could minimize packet loss as compared to the existing system.

Keywords: DCXP, LISP, MobileIP, HOST, MediaSense, SensibleThings.

Acknowledgements

It has been such a privilege to work with academicians and other personnel at Mid Sweden University. It would not have been attainable and hard to imagine the environment had there not been very friendly people and nice students from Asia, Africa and, of course, Sweden. In all this period of time there are so many people that have assisted me to achieve my goals but attempting to mention all of them in here is quite impossible. However, I would like to give credit and appreciation to people who have given me much time and guided me in relation to this thesis work. Firstly, I would like to offer my gratitude to Victor Kardeby my supervisor for advising and have guiding me throughout the entire report. Secondly, I want to thank Majid Ali and his roommates for giving me advice and spending tea times together. Finally, the warmest appreciation goes to my examiner, Professor Tingting Zhang, who gave me many courses in the entire program, for here unfailing help and guidance through her tight schedules.

Table of Contents

Terminology

Abbreviations

ACK	Acknowledge.
CI	Context Information
CUA	Context User Agent
DCXP	Distributed Context Exchange Protocol
HIP	Host Identity Protocol
LISP	Location Identifier Separation Protocol
MD	Mobile DCXP
MDP	Mobile Exchange Protocol
MOBIKE	IKE2 Mobility and Multihoming Protocol
PMIPv6	Proxy Mobile IPv6

1 Introduction

Human beings possess amazingly voracious appetites for a better and easier life. At the end of the last century as well as in the previous decade we have witnessed a huge move with regards to the innovations in the area of Information Technology, electronics and electro-mechanical technologies among others. In the area of information technology, the Internet is one of the revolutionary innovations which arrived at its currently socially valuable technology through time. Moreover, in the current advancement of human beings' imagination and available supporting technologies, there is a buzzword known as Internet-of-things all around the Internet and in the research labs. In fact some companies have attempted to take advantage using it in business when it only at the conceptual stage than actually feasible and profitable when observed from the business point of view.

It has been a few years since the Internet-of-things overlay known as the MediaSense has been proposed from Mid Sweden University in cooperation with the European Union and other parties. Moreover, a number of projects related to the MediaSense have been carried out. The MediaSense project has produced components for the accumulation of context information from sensors and wireless sensor networks. The context information originates from numerous different sources such as sensors attached to mobile phones or home gateways. These devices are communicating via IP addresses. Using mobile devices with changing Internet access, these IP addresses may change without notice during a session. This becomes an issue when both the producer and the consumer change their point of attachment simultaneously.

In this thesis the concern is how to deal with context-aware applications in a situation where a user switches between networks. In addition, the thesis attempts to find a place for the developments, improvements and proposals within the SensibleThings platform.

1.1 Background and problem motivation

Applications that can change their behaviour based on the context of users are known as context-aware applications. These applications have a good market penetration with the introduction of smart phones and others similar related devices, which come with a multitude of embed-

ded sensors and built in actuators. This thesis is part of an effort to arrive at a next generation Internet-of-things architecture and its supporting protocols.

It has been observed that recently there are developments with regards to the utilization of the context-aware applications for example in an area where tourists go frequently, mobile workers, and adverts. The use of the technology within the modern society is in its infant stage. Moreover, common architectures and design principles are being developed and studied by different parties. The aim of thesis is to work on the implementation of the context-awareness with a focus on the mechanisms regarding on how to deal with a situation where such apps running on different devices manage to switch networks from one to the other smoothly without affecting its usage.

1.2 Overall aim

In this thesis the aim is to survey current mobility solutions and identify their shortcomings based on future requirements on mobility as exemplified from a scenario where nodes change their point of attachment arbitrarily in a distributed environment. Additionally, part of the overall aim is to propose a solution based on the SensibleThings architecture. Therefore, the thesis determines out a solution on regarding how to incorporate the mobility solutions in the existing SensibleThings Architecture. Finally, part of the aim has been to implement the proposed solution as an extension to the modules of the SensibleThings platform.

1.3 Concrete and verifiable goals

The goal of the project is to propose and implement an extension to the SensibleThings architecture that could accommodate mobile nodes that disconnect and connect in different networks.

The concrete and verifiable goals of the thesis have been divided into the following three goals:

- GOAL ONE: Study the existing mobility solutions and propose scalable mobility solution for the SensibleThings platform.
- GOAL TWO**:** Implement the proposed mobility solution as an extension for the SensibleThings platform.

- GOAL THREE: Evaluate the performance of the proposed mobility solution.

1.4 Scope

The focus of this thesis has been to study the existing mobility solutions with regard to protocols and technologies related to an IP network. Moreover, the thesis proposes a mobility solution that could be part of the existing SensibleThings architecture. Finally, the thesis implements an extension to the existing SensibleThings architecture so as to make the architecture more powerful in this specific case and fulfil future requirements of the scalable mobility. In addition, the proposed mobility solution will be evaluated and examined in detail. Comparison with the existing system will be made. To this end a simple application running on top of the SensibleThings platform will be developed.

1.5 Outline

Chapter one introduces the backgrounds of wireless access point handover issues. In this chapter, the problem, the aim, the concrete and verifiable goals and the scope of the project are briefly presented.

Chapter two discusses related works. In this section existing mechanisms to handle packet loss due to wireless access point handover have been discussed. Here protocols, solutions and optimization tasks have thoroughly covered.

Chapter three presents the overall methodology followed to arrive at the correct result and conclusion. In this section the task has been divided into smaller goals which could link up together and come to the intended results.

Chapter four discusses the design of implementations carried out. Moreover, it also explains in detail the proposed workaround solution (or the add-in layer extension) and discusses some other possible solutions. At this point proof-of-concepts, diagrams and tables have been used to illustrate the concepts.

Chapter five presents the implementation of sections. In this chapter the implementation of Mobile DCXP, IChat and proof-of-concept app have been presented in detail.

Chapter Six presents the results sections. The results sections presents the evaluation and comparison of the proposed Mobile DCXP with the existing Mobile DCXP proxy. Here graphs, tables and diagrams have been used for illustration purposes

Chapter Seven discusses the conclusion from the thesis work. In this chapter, future work and contribution of the thesis have been discussed.

2 Theory

The requirement for optimization and efficiency comes not at the beginning but at the later stages of development or any progress due to the very reason related to the nature of human learning instinct. The usage of radio technology for data communication and eventually as a networking device was one such undeniable success. However, in the beginning hardly anybody could imagine what might possibly be achieved as research ambition related to the wireless-access-point. Thanks to the kindness of time, the question we are asking today is not how to connect devices wirelessly rather it is about how to avoid packet loss during roaming and thus perform a smooth handover when devices switch from one access point to another.

Therefore, this chapter presents the theoretical backgrounds of the protocols, techniques and mechanisms proposed and being utilized currently in order to perform a smooth handover in wireless-access-point. In addition, a literature overview on the backgrounds of the concepts of the SensibleThings and Dissemination Context Exchange Protocol has been presented.

2.1 Core Issues and Problems

Currently, the usage of wireless-access-point is becoming more and more prevalent as well as valuable within the community. This can be totally attributed to the human deep rooted need for freedom and freedom of movement in particular, which is taken for granted in this case. Of course, Wireless-access-points are not recommended for all kinds of scenarios where communication needs to be built. Thus, wireless-access-points have their cons and also pros[1] However, the concern of the thesis is not to study about these technologies but to delve into them more deeply and work on the efficient handover during switching from one wireless access point to another access point.

In the existing IP address utilization, handover is carried out successfully however the session will not be retained. In other words, smooth handover where there is no packet loss cannot be carried out. The most pressing problem in the existing packet based networking protocols are Loss of trust, surge of unwanted traffic, choking routing systems poor support of mobility and multi-homing, lack of privacy and accountabil-

ity[2]. In order to carry out a smooth handover in wireless access-point switch a number of mechanisms have been proposed and different protocols have been ratified. In this section the most commonly accepted mechanisms are presented.

2.2 Proposed Solutions and Currently Utilized Mechanisms

Protocols proposed with regard to the issue of smooth handover in the wireless access-point serve different purposes, however all of solutions have one cause and that is to carry out a smooth handover or help assist in carrying out a smooth handover. Protocols for example HIP, LISP, Mobile IPv6 propose their own means to tackle the issue. However, Hierarchical MIPv6, PMIPv6 have a focus on the optimization of the existing Mobile IPv6.

2.2.1 Host Identity Protocol

The Host Identity Protocol (HIP) is an internetworking architecture and an associated set of protocols, developed at the IETF since 1999 and reaching their first stable version in 2007 [3]. HIP is an additional name space besides the two name spaces (IP and DNS) used in the Internet architecture [4]. HIP is cryptographic in its nature; it is a public key of an asymmetric public key pair [5]. HIP integrates IP-layer mobility, security, multi-homing and multi-access, NAT traversal and IPv4/v6 interoperability. Technically, the basic idea behind HIP is to add a new name space to the TCP/IP stack [3]. In the HIP layer Hosts are given identifications, Host identifications. Each host identity represents a unique host.

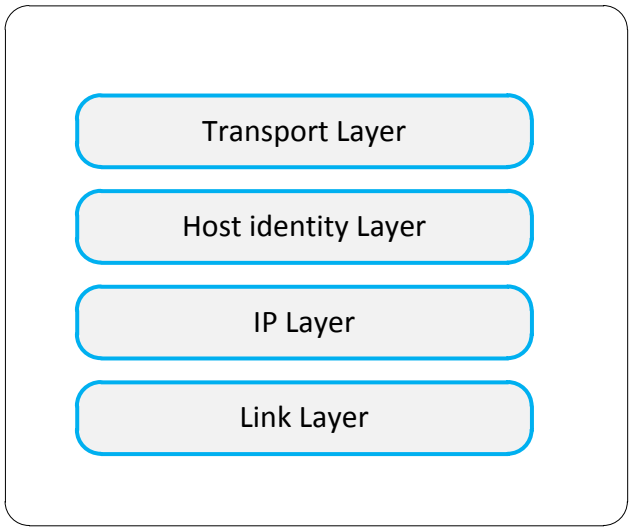


Figure 1 Location of HIP layer in the TCP/IP stack.

- **Benefits of HIP [5]**

HIP could provide the following benefits:

- **Non-mutable:** The address sent is the address received.
- **Non-mobile:** The address does not change during the course of an "association".
- **Reversible**: A return header can always be formed by reversing the source and destination addresses.
- **Omniscient**: Each host knows what address a partner host can use to send packets to it.

- **Mobility**

When a host moves to another network notification will be sent to its peers. The notification is performed through an HIP update packet containing a LOCATOR parameter. Acknowledgment will be sent back to the moving host. The update is retransmitted again in order to have more reliable communication in the case of packet loss [6].

2.2.2 LISP

LISP (Locator Identifier Separation Protocol) provides a set of functions for routers to exchange information used to map from non-globally routable End Point Identifiers (EIDs) to routable Routing Locators

(RLOCs) [7]. LISP has been proposed based on observations made from a different angle. The basic observation involves using a single address for both identifying the device and locating the device and this requires compromise involving of a topology based identifier assignment and no explicitly based identifier assignment. In order to carry out efficient routing there should be a topology based identifier assignment whereas in order to manage efficiently when a number of devices exist and to handle situations where devices require renumbering, it is advisable not to have explicit attachment of an identifier with the topology [8].

Moreover, due to the fact that LISP is map-and-encap protocol, there is no need to change the host stack [9]. See figure 2 Map-and-encap protocols appends header to the existing header.

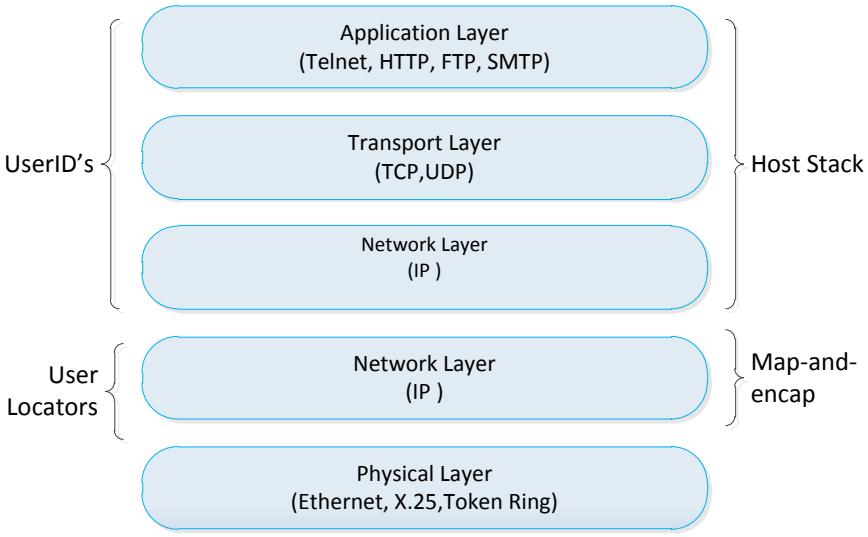


Figure 2 LISP [10]

Routing Scalability issue has been solved by assigning two types of numbers for each device's IP address: RLOCs - Routing Locators and EIDs - Endpoint Identifiers. RLOCs are assigned topology based; RLOCs are used for data forwarding and routing in the network. However, EIDs are assigned independently of the topology; EIDs are used for numbering [11].

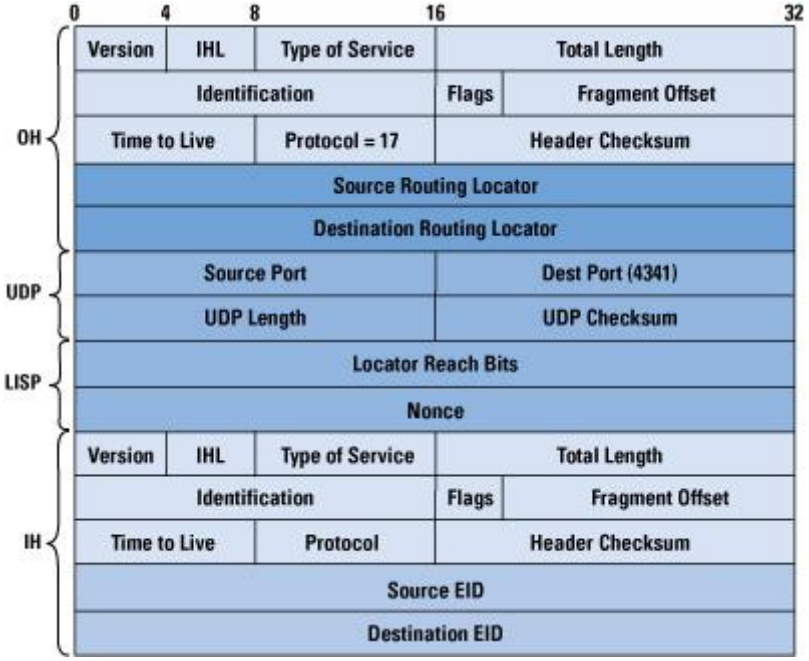


Figure 3 LISP Header Format [7][12]

- **Benefits of LISP [13]**

LISP provides the following benefits:

- Improved routing system scalability by using topologically-aggregated RLOCs
- Provider-independence for devices numbered out of the EID space (IP portability)
- Low-OPEX multi-homing of end-sites with improved traffic engineering
- IPv6 transition functionality
- IP mobility (EIDs can move without changing - only the RLOC changes!)

2.2.3 Mobile IPv6

Mobile IPv6 targets the offering of smooth hand over of mobile nodes during switching between access-points. Mobile IPv6 provides unbroken connectivity for mobile nodes when roaming between wireless access points in a different subnet in an operation known as L3 (Layer 3) handover [14]. Handover might be carried out on layer two or layer

three depending on the wireless access points involved. For wireless access points on the same subnet, the handover occurs on the layer 2 however to switch from one access point to the other, which belongs in a different subnet the handover is carried out in layer 3.

The Mobile node is identified by its two addresses in its entire process. The first address is known as the home address and the other is known as the care-of-address. The home address represents the IP address of the mobile node when it is attached to a home network. However, when the mobile node wanders around and becomes attached to another access point it will be assigned a care-of-address and this care-of-address will be registered to its home agent and correspondents. The association made between the home address and care-of-address is known as binding [7]. In this protocol until a binding update is sent to the home agent (see figure 4 below) the packets coming to the home address will be lost.

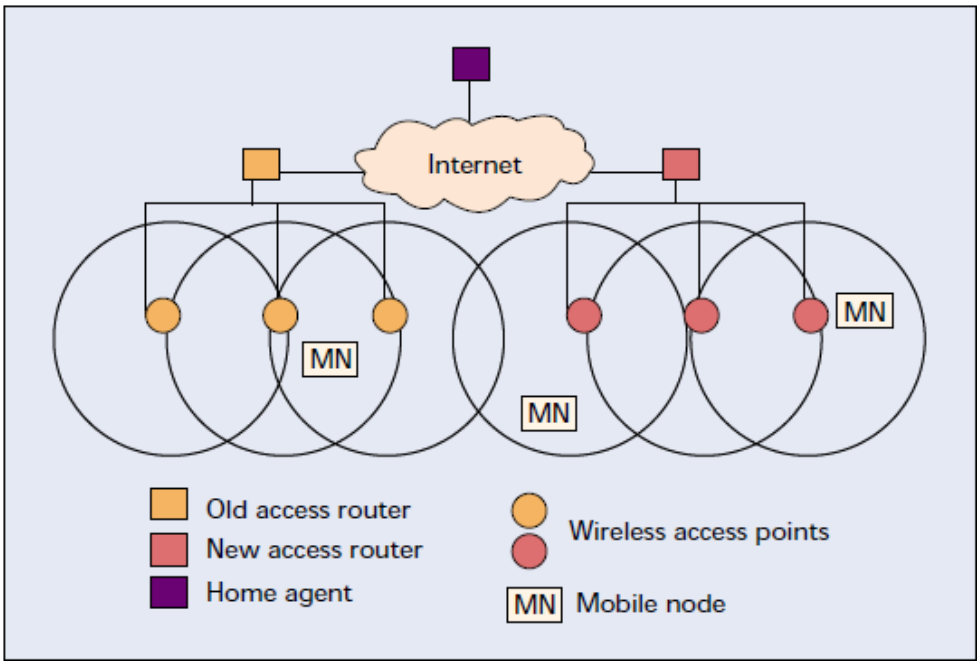


Figure 4 Mobile IPv6 Wireless Network Architecture [14]

Procedure

Here are list of steps taken to discover out a node which has moved to another network

1. MN detects that it has moved to another network through a periodic advertisement coming from the router.
2. Based on the messages being advertised from the router, the Mobile node obtains a new care-of-address
3. Mobile node performs duplication address detection (DAD) on its link-local address
4. Uses either Stateful or stateless address auto configuration
5. Mobile node performs DAD for the care –of-address.
6. Mobile node carries out bind update.

- **Benefits of Mobile IPv6**

The main benefit of this standard is that the mobile nodes (as IPv6 nodes) change their point-of-attachment to the IPv6 Internet without changing their IP address [15].

2.2.4 Hierarchical Mobile IPv6

The Hierarchical Mobile IPv6 (aka HMIPv6) follows similar concepts as in the Mobile IPv6 presented above. However, HMIPv6 introduces new functions known as Mobile Anchor Points (MAP) [16] and minor extensions. The Mobile node might send a packet to its Home agent immediately after update binding, however, in cases where the home agent and the Mobile nodes are far apart the home agent would be unable trace back to the Mobile node before receiving the binding update so packets will be lost. This drawback could have a significant impact on data communication where time a critical handover is taking place on.

The introduction of the MAP provides a solution to the issues within the Mobile IPv6:

- The mobile node sends binding updates to the local MAP rather than the home agent (HA) (which is typically further away) and correspondent nodes (CNs) **[17]**.
- Only one binding update message needs to be transmitted by the mobile node (MN) before traffic from the HA and all CNs is re-routed to its new location. This is independent of the number of CNs with which the MN is communicating [17].

- **Benefits of Hierarchical Mobile IPv6**

  Handover performance improvement due to the fact that local handover is performed locally which results in faster transition time and thus less packet loss.

  Reduces the mobility management signaling load on the network [18]

2.2.5 Fast Handover

Fast Handover is not a protocol standing on its own to handle the issue of wireless access point handover however, the protocol is aimed at improving the handover latency which might occur during the switching of a node from one access point to another using Mobile IPv6. The handover latency is caused by movement detection, new care of address configuration and binding update. Although the handover latency for Mobile IPv6 is small, it has an impact on the normal communication of nodes carrying out voice over IP and is thus intolerable in some situations. Fast handover is concerned with the following two questions: How to allow a mobile node to send packets as soon as it detects its new subnet link, and how to deliver packets to a mobile node as soon as its attachment is detected by the new access router [17][19].

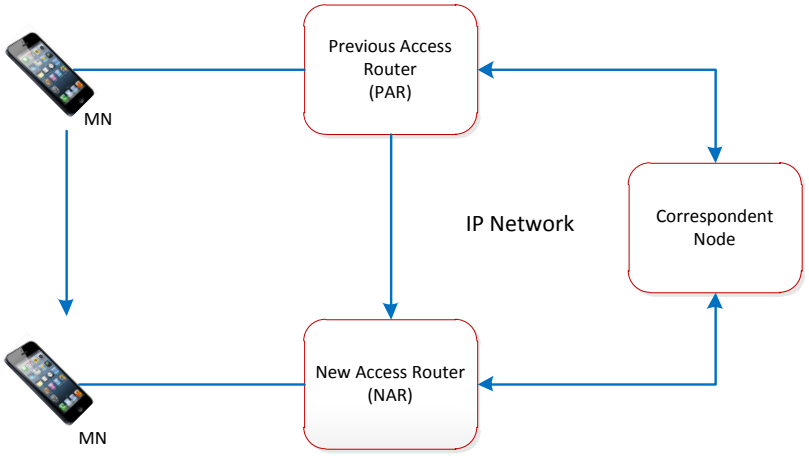


Figure 5 Handover Diagram[20]

Benefits of Fast Handover

The protocol reduces packet loss by combining packet tunnelling with buffering during the time the mobile node is switching between access routers [21].

2.2.6 MOBIKE

MobIKE also known as IKEv2 Mobility and Multihoming Protocol, allows the IP addresses associated with IKE2 (Internet Key Exchange) and tunnel mode IPsec Security Associations to change**Error! Reference source not found.**. IKE2 is used for performing mutual authentication, as well as establishing and maintaining IPsec Security Associations [22]

The main Scenario for MOBIKE is keeping the VPN user security associations without the need for re-establishing the task all over again later on [23].

MOBIKE also supports more complex scenarios where the VPN gateway also has several network interfaces [22].

Table 1 A Simple MOBIKE Exchange in mobile environment

INITIATOR	**RESPODER**
1) (IP_I1:500 -> IP_R1:500) HDR, SAi1, KEi, Ni, N(NAT_DETECTION_SOURCE_IP), N(NAT_DETECTION_DESTINATION_IP) -->	

```
                              <-- (IP_R1:500 -> IP_I1:500)
                                   HDR, SAr1, KEr, Nr,

                                       N(NAT_DETECTION_SOURCE_IP),

                                   N(NAT_DETECTION_DESTINATION_IP)
```

```
   2) (IP_I1:4500 -> IP_R1:4500)
      HDR, SK { IDi, CERT, AUTH,
                CP(CFG_REQUEST),
                SAi2, TSi, TSr,
                N(MOBIKE_SUPPORTED) }  -->

                              <--  (IP_R1:4500 -> IP_I1:4500)
                                   HDR, SK { IDr, CERT, AUTH,
                                             CP(CFG_REPLY),
                                             SAr2, TSi, TSr,
                                             N(MOBIKE_SUPPORTED) }

   (Initiator gets information from lower layers that its
attachment point and address have changed.)
```

```
3) (IP_I2:4500 -> IP_R1:4500)
      HDR, SK { N(UPDATE_SA_ADDRESSES),
                N(NAT_DETECTION_SOURCE_IP),
                N(NAT_DETECTION_DESTINATION_IP) }  -->

                              <-- (IP_R1:4500 -> IP_I2:4500)
                                                    HDR, SK {
                                       N(NAT_DETECTION_SOURCE_IP),

                                  N(NAT_DETECTION_DESTINATION_IP) }

   (Responder verifies that the initiator has given it a correct
IP address.)
```

```
   4)                         <-- (IP_R1:4500 -> IP_I2:4500)
                                  HDR, SK { N(COOKIE2) }

      (IP_I2:4500 -> IP_R1:4500)
      HDR, SK { N(COOKIE2) }  -->
```

2.2.7 PMIPv6

PMIPv6 (Proxy Mobile IPv6) is a protocol proposed by the IETF to reduce latency during handover and thus packet loss that occurs during handover in the MIPv6 protocol. PMIPv6 is intended for providing network-based IP mobility management support to a mobile node, without requiring the participation of the MN in any IP mobility related signaling [24]. PMIPv6 offers two new functional entities the Local

Mobility Anchor (LMA) and the Mobile Access Gateway (MAG). The MAG detects the mobile nodes attachment and provides IP connectivity. The LMA is an entity which assigns one or more Home Network Prefixes (HNP) to the MN and is the topological anchor to all traffic belonging to the MN [25].

The MN and LMA should have a local policy in place which makes sure that packets are forwarded coherently for unidirectional and bi-directional communication. The MN decides on the final IP flow mobility decisions, and then the LMA follows that decision and updates its forwarding state based on the decisions made [26].

- **Benefits of Mobile PMIPv6** [27]
  - The delay in sending signalling to LMA is lower as compared to sending signalling to a remote home agent in the case of Mobile IPv6.
  - Less overhead as compared to IPv6

2.3 The SensibleThings Project

The SensibleThings is a novel architecture for Internet-of-things application development. SensibleThings is a distributed architecture that enables Internet-of-things based on sensor and actuator information. The entire SensibleThings platform is divided into five layers as shown in figure 6

- **The Interface Layer:**
  The Interface layer is the public interface through which applications interact with the SensibleThings platform [28]
- **Sensor and Actuator Layer**
  The purpose of the sensor and actuator layer is to enable a generalized method to produce information and provide it to the SensibleThings platform [28].
- **Add-in Layer**
  The Add-in layer is intended for a developer who would like to add extensions to the platform or to carry out optimization works.

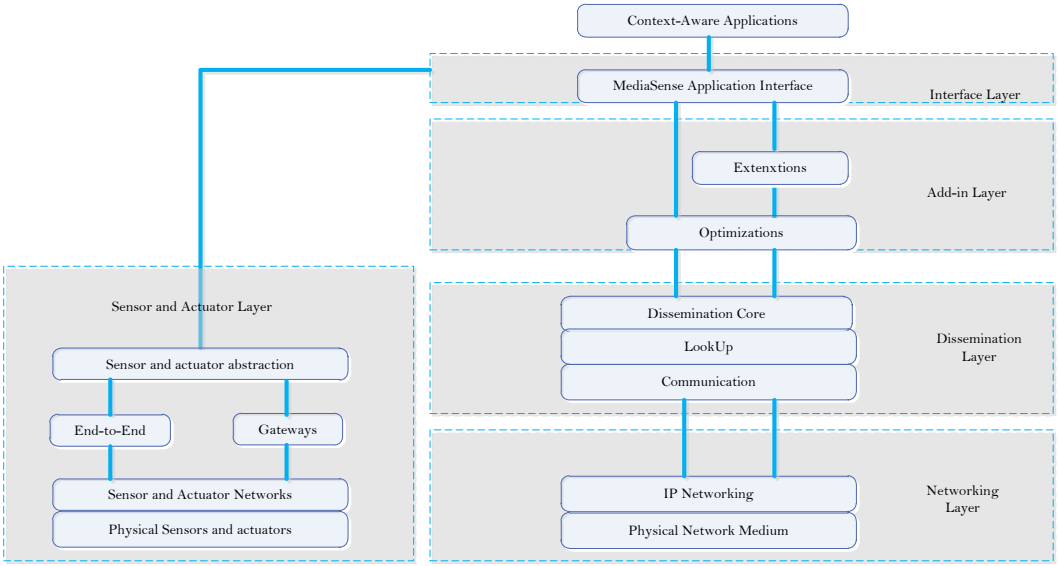


Figure 6 the SensibleThings Platform Architecture

- **Dissemination Layer**
  The dissemination layer is involved in disseminating information among the participating entities in the Internet-of-things.
  - Dissemination Core
  - Lookup
  - Communication

- **Networking Layer**
  The networking layer performs the connection of IP based networking communications. The networking layer is divided into two components: IP networking and physical network medium. The Sensible Things platform is independent of a particular networking medium and it is designed to run on heterogeneous networks.

2.4 Distributed Context Exchange Protocol – DCXP

The Distributed Context Exchange Protocol (DCXP) is a peer –to-peer protocol used within the SensibleThings framework to exchange context between users and entities. DCXP is a SIMPLE-inspired protocol with five primitives (REGISTER_UCI, RESOLVE_UCI, GET, SUBSCRIBE, NOTIFY) [28]. DCXP is nothing but an XML based application level protocol [29][30], which serves reliable communication of context information among nodes that participate in the overlay network **Error! Reference source not found.**. Although the term is a bit fuzzy, the

design of DCXP satisfies the real-time requirements for the provisioning of context information.

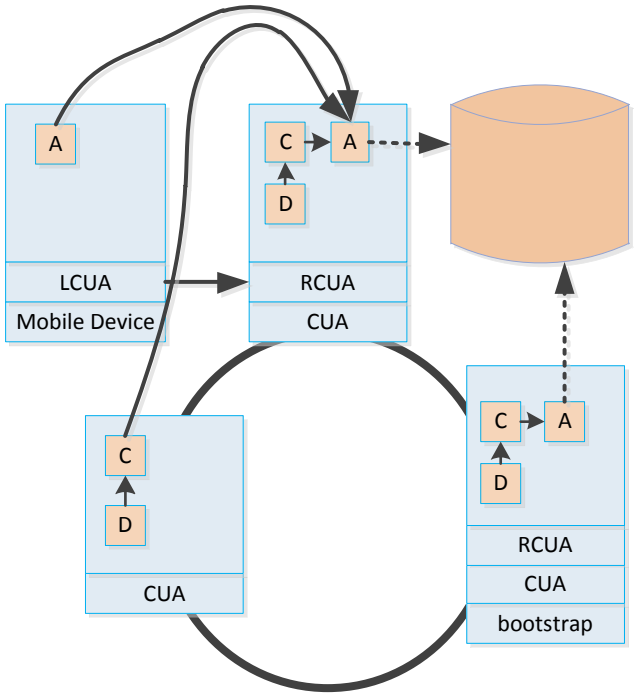


Figure 7 DCXP Architecture [28] CUA corresponds to a node in the DHT ring

- C- a database client
- D-data miner
- A-a database Agent
- RCUA- Remote CUA
- LCUA- Limited CUA

2.5 DCXP Messages

REGISTER_UCI:
A CUA (Context User Agent) uses REGISTER to register the UCI of a CI (Context Information) with the DS.

RESOLVE_UCI:

In order to find where a CI is located, a CUA must send a RESOLVE to the CS.

GET:
Once the CUA receives the resolved location from the Context Storage, it GETs the CI from the resolved location.

SUBSCRIBE:
SUBSCRIBE enables the CUA to start a subscription to a specified CI, only receiving new information when the CI is updated.

NOTIFY:
The source CUA provides notification about the latest information to subscribing CUAs each time an update occurs or if asked for an immediate update with GET.

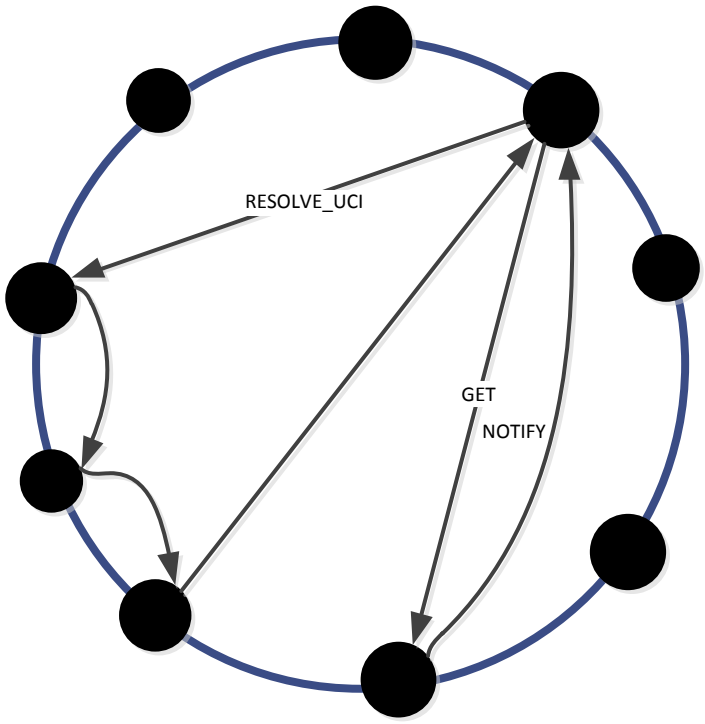


Figure 8 DCXP Signalling [28]

2.6 Mobile DCXP Proxy

Mobile DCXP proxy (aka MDP) is a server providing two important functions in the DCXP network.

1. Mobile devices must register on the MDP to gain access to the peer-to-peer network In addition, it processes computation and thus reduces the burden on the mobile devices.

2. Shields the peer-to-peer network from packet loss and other likely disruptions arising from the behavior of a radio communication link.

3 Methodology

In this section the approach adopted to achieve the aims pinpointed in the first chapter has been elaborated. Therefore, the overall plan has been to study the existing mobility solutions, which assists in gaining a sufficient enough knowledge base about the possible solutions and which could act as a springboard for proposing and implementing a scalable mobility solution for the SensibleThings platform. Then the thesis presents the design and implementation of mobility solutions for SensibleThings platform.

Accordingly, the overall order of approaching the challenges is first of all to study the existing related mobility solutions and present these mechanisms. Second of all, thoroughly examine and understand the SensibleThings platform. Third of all, implement scalability solution as an extension to the existing implementation. Finally, evaluate the performance of the solution that has been proposed. The approaches have been precisely presented and explained as in the following subsections.

3.1 The existing scalable mobility solutions

A brief study of the existing scalable mobility solutions is presented and discussed. The thesis covers those solutions which consist of better support as well as providing common and open solutions. The thesis has a target in this case to present all the relevant technologies and specifically to place emphasis on those technologies designed to encompass the architecture of the future Internet-of-things. Thus in this section, the plane is to cover LISP, HIP, Mobile IPv6, and other commonly used protocols. For this task, previous related research work papers and standards from organizations such as *ietf* have been used. Furthermore, previous research papers have been collected from *IEEE* and their corresponding webpage.

3.2 Scalable mobility solution for the SensibleThings platform

As explained in the first chapter, one of the goals in this thesis has been to design and implement a mobility solution for the SensibleThings platform. To this end, we a study will be made of the DCXP protocol which has been used to disseminate context information between nodes. More specifically, the focus is on the Mobile DCXP proxy server (MDP)

and the intention is to turn this part of DCXP into a scalable solution as well as providing better mobility. To make the concept and the proposal vivid, the thesis presents designs, proof-of-concepts and diagrams.

Furthermore, the aim of this subsection is to achieve two unique but closely related goals. In this case the aim is not just to achieve better mobility but also to make sure that the scalability is better.

3.2.1 Scalability

Scalability of a mobile node within the *SensibleThingsPlatform* is entirely dependent on the underlying DHT or algorithm. In the paper **Error! Reference source not found.** the scalability of platform which is of our concern has been achieved through scalable lookup as in P-grid and Chord which scale logarithmically as the whole entities grow. In addition, the proposal[1] for the thesis clearly stated that the thesis is required to focus on the implementation of a mobility solution extending the existing *SensibleThingsPlatform*. So the term scalability in this notion is to express the fact that when the number of nodes grow the mobility solution should scale with little or no performance degradation. Such a feature of the proposed solution has been discussed thoroughly and an attempt has been made to prove it be means of theoretical assumptions rather than experimental research.

3.2.2 Mobility

The mobility aspect of a node connected to overlay networks through wi-fi could be improved on the 2nd layer or 3rd layer of router/Access point depending on the situation at hand. For example, if the wireless access points are in the same subnet then it could be implemented on the 2nd layer but, if the access points are in a different subnet then the demand is for the 3rd layer. In addition to these, it is also possible to work on the application layer to achieve better mobility. Of course, the third option depends on the kind of application which is of concern. For example if the application could not tolerate even a very small latency of data packet then it is not possible to work on the top layer to optimize the efficiency. Nevertheless, as in the case with this thesis if a few seconds of delay (for example, a delay a node takes from one switch to another access point) are tolerable, then the third option could be quite a possible solution.

[1] Proposal …

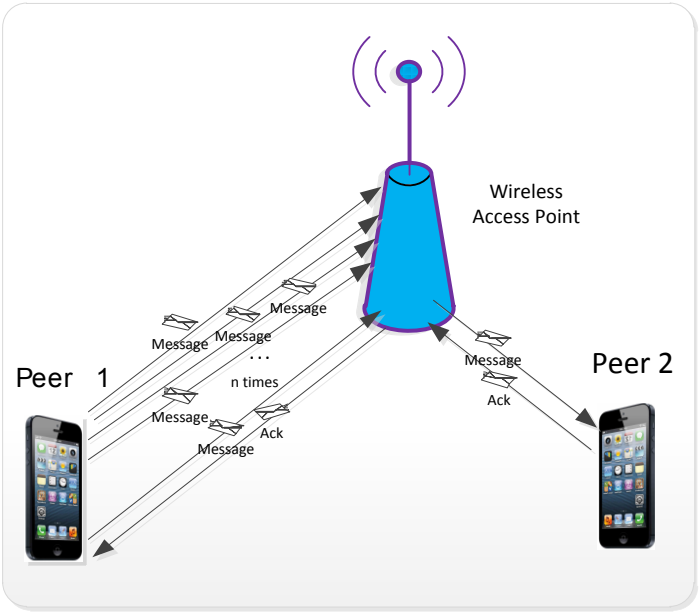


Figure 9 Mobility improvement

3.2.3 Implementation of Mobility Solution

After a thorough study, for example using proof-of-concepts, and close examination of the concepts used in the proposed solution, a solution was designed. Based on the proposed design, we implemented the solution using the Java programming language. The implementation which is to be done in this section ensures better mobility as compared to the existing implementation. The solution developed in this section is not a totally new solution built up from scratch but it could possibly be considered as an optimization to the existing system so as to enhance a smooth handover and better mobility. The ultimate goal of the solution is to decrease packet loss during handover for applications running on the *SensibleThingsPlatform*.

3.2.4 Development Environment and Utilized Tools

The existing implementations of *SensibleThingsPlatformCore* and *SensibleThingsPlatformImp* have been imported onto Eclipse. On *SensiblethingsPlatformImp* within the package *se.sensiblethings.addinlayer.extentions.mobiledcxp* the proposed modifications have been added. In addition , *SensibleThingsAndroidExnaample* have been downloaded from *www.SensibleThings.se* and used to check whether the modification works.

3.2.5 Implementation Approach

It has been set as a rule that any modifications or extensions on the *SensibleThingsPlatform* should not be carried out within other layers than the add-in layer. However, this seemingly unimportant rule keeps the consistency of the original concept. Therefore, as per the rule the approach followed has been to add a new package in the add-in layer.

3.3 Evaluate the Performance of the Proposed Mobility Solution

The last task has been to evaluate and prove that the proposed mobility solution could be applied in a real scenario. In the evaluation the existing mobility solution and the proposed solution has been compared. To this end an android app has been developed and run on a different scenario as well as hardware. In addition, a proof-of-concept application has been developed which illustrates that the idea in the proposed solution could actually work. In this part of the task, we will count the packet losses as compared to the packet losses in the existing system. The results will then be illustrated using diagrams and figures.

Hence, the network has been setup as shown in figure 11 below. In the network two mobile devices (peers), Wi-Fi link and developer laptop are included. The tools that have been used to determine out the packet losses are: *WirShark™* and *LogCat* on eclipse.

3.3.1 Android App Used for Testing

The design and implementation of proposed scalable mobility solution is to be provided in the next chapter. Then after the successful design and implementation, in order to show how the optimizations have really been achieved, a quantitative analysis has been conducted. At this stage there are two systems to compare. Nevertheless, since these are merely extensions on the overlay network, there should be an application which could run on these overlay networks. Therefore, for testing purposes an Android Testing application has been developed. The application developed is not large rather it is an application which is sufficient to provide assistance in relation in gathering statistics. See figure 10 below.

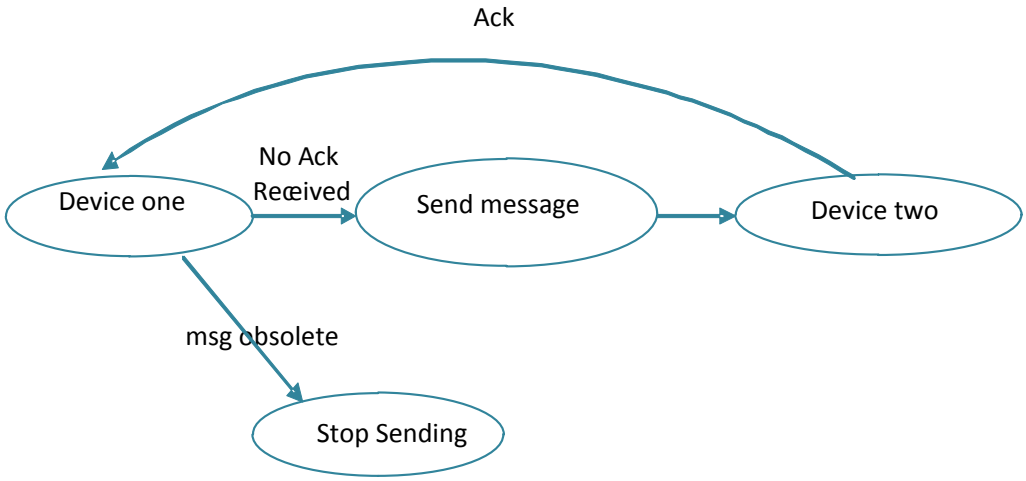


Figure 10 Test app state machine diagram

3.3.2 The Experimental Setup

At this stage, the android app has been installed on two mobile devices, *TP-Link 3G/4G Wireless N Router* has been set on, *LogCat* on *Eclipse IDE* is opened, *Wireshark* has been set ready for launch and the digital timer has been set to ready. The network has been shown in figure 11.

Parameters to Measure (Packet Loss):

Messages sent and received. Peer2 sends and Peer1 receives. Using the *LogCat* it is possible to view how many times message retry has been made and when the final success occured. In addition, *Wireshark* also shows the details of the list of packets exchanged.

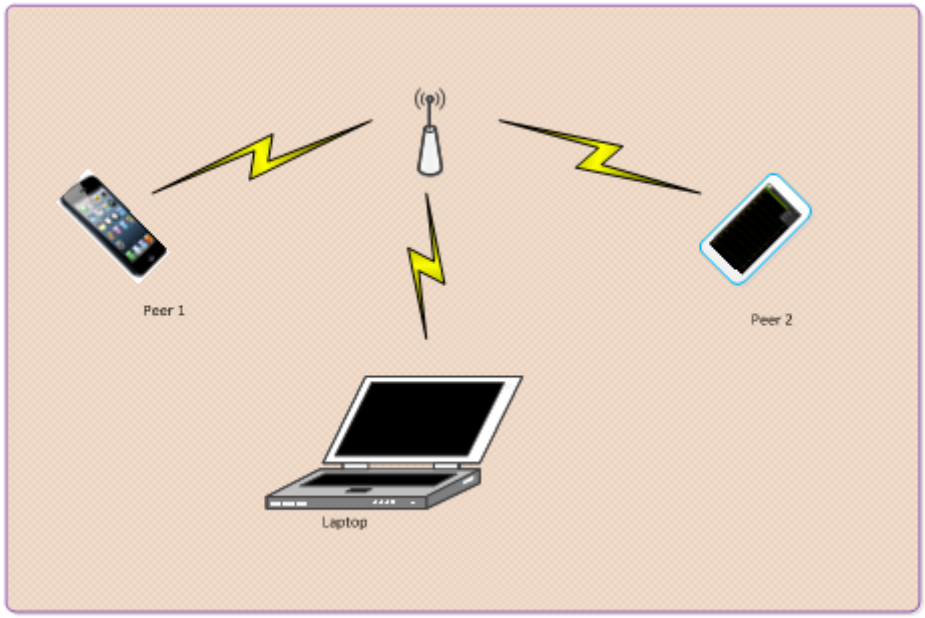


Figure 11 Data gathering Mechanism

Parameters to Measure (Jittering):

Jittering is the variance of measurement of Ping messages to a particular resource. In order to measure jittering we need to send ping messages from the laptop to remote device continuously throughout the experiment. The sending node and the laptop are connected throughout the experiment so when the second device gets disconnected due to handover then it is not possible to ping from the Laptop either. Therefore, sending ping message from the first device yields the same result as sending ping message from the laptop.

3.3.3 Testing and Experimental Data gathering Strategy

Three different sorts of data have been gathered to measure the performance of the solution we implemented and then later on to compare it with the solution from *SensiblThingsPlatform*.

Packet Loss

In the experiment set up as shown in figure 11 we have two mobile nodes. To count the packet loss, the first node which is connected to the laptop has been made to send continuous stream of packets to the other device. While the other node is roaming it gets connected and disconnected depending on the strength of the radio Signal. When the second device gets packets, we have acknowledgement messages send back to

the first device. Therefore, using *LogCat* and *WireShark™* which are already running on the laptop the number of these acknowledgements and lost packets could be gathered.

Jitter

In the experiment to examine the jittering, the same experimental set up shown in figure 11 has been used. Furthermore, in order to collect the necessary data to examine the jittering, the laptop which is connected to the first tab has been made to send continuous ping messages as the other device is roaming and at a time it is gets disconnected as the signal strength drops. So here the time the Ping Message takes to conduct a round trip of the entire path has been collected. In this experiment it is not expected to arrive at different round trip times as the network is LAN except that when the device become disconnected during the handover. A jittering test has been included for comparison as it is one of the measures of network quality however, the implementation carried out is not expected to improve jittering because the protocols implemented on the 2nd and 3rd layers have not been changed.

Subjective Testing

In the subjective testing thirty students from Mid Sweden University (MIUN) have been assigned a task to run and examine the differences of two applications one at a time. The results of the evaluation have been noted by means of filled questionnaire.

The following is list of procedures for the subjective test:

Step 1:

Assign the other peer as the subject to examine the differences.

Step 2:

Choose appropriate spots within the hotspot coverage. Start from a strong signal and take four spots till handover occurs

Step 3:

In the experiment create a chat session and let the second device roam around. The chat messages are a predefined list of messages which should be repeated two times: first on the existing system

and then on the proposed modifications. Furthermore, during the chatting the user should not type but just press *Send* button.

Step 4:

Repeat the experiment with the existing System.

Step 5:

Fill the questionnaire. The questionnaire is listed under Appendix A.

3.3.4 Evaluation and Comparative Analysis

The data gathered in the experiment explained in this section has been subjected to qualitative analysis. The statistical data from the experiment has been illustrated on different informative diagrams. Moreover, based on the results, important conclusions and discussions have been drawn and made.

3.4 Tools and important equipments used in the Performance Evaluation

The tools utilized in the experiment to gather the necessary data and make observations on the mobility extension implemented has been categorized into two: Software tools and hardware

3.4.1 Software

During the development of the Android apps, we used the Eclipse development environment was used. To draw diagrams and figures, *Microsoft Visio 2010* has been used. In addition, networking tools, for example *WireShark™*, has been utilized to examine the packets exchanged.

Software Tools:

- **Eclipse (LogCat)**

  Eclipse has been used to implement the test android app, proof-of-concept and one of the powerful tools on Eclipse which is *LogCat h*as been used to observe details of messages going out to the other end.

- **WireShark™**

  In the experiment all the network traffic has been captured with *Wireshark™*. The *wireshark™* provides tools for examining the packets.

- **Microsoft Visio 2010 / Excel 2007**

  In here the data gathered from the experiment are brought together for analysis and to draw illustrations.

3.4.2 Hardware

During the testing of the implementations, we have used a Computer, an Access Point, two mobile devices as well as a camera.

Hardware Tools:

- **Computer**:

  Specification: Laptop Packard Bell (PB), 2.20GHz, 2GB memory, 64-bit OS.

  The laptop has two important functions in the experiment. First we implemented the testing app on the laptop. Second the jittering test has been carried out from the laptop, which is attached to the sending node, to the other end.

- **Access Point:**

  Specification: TP-Link TL-MR3420, IEEE 802.11b/g/n, IEEE 802.3/3u, 2.4GHz, 300Mbps

  The access point has been used as a wireless router when the nodes are attached and detached as the devices roam.

- **Mobile Devices:**

  Specification: Samsung galaxy tab 7.0 Model number: GT-P3110 and XTouch Model number: X704.

- **Camera**:

  Specification: IPhone 4 8GB.

Camera has been used to take photographs of the experiment for reporting purposes.

3.5 Theoretical analysis of tasks achieved in this thesis

The aim in this section is to evaluate the overall achievements in this thesis and also to theoretically analyze the approach followed to achieve the goals. Therefore, in this section the efforts and advanced techniques in the thesis work have been discussed.

3.5.1 Theoretical analysis of output of the thesis

The research carried out in this thesis work aims to propose and implement a scalable mobility solution on the existing SensibleThings platform. Accordingly, two alternatives have been proposed, one is within the scope while the other falls outside the scope. The actual outputs from this task have a new mechanism to increase the scalable mobility on the SensibleThings platform known as Mobile DCXP which is an improvement to the previous Mobile DCXP proxy. The implementation of Mobile DCXP has been conducted and an evaluation of our solution has been performed. In addition, a comparison of the existing system and the proposed solution has been carried out. A mobile application has been developed to study the scalable mobility and furthermore a proof-of-concept application has been created to show the concept of Mobile DCXP (MD).

3.5.2 Ethical Deliberations

The research result from this thesis work is intended to offer a positive contribution for researchers and applications developers who are interested in the *SensibleThingsPlatform*. Furthermore, methodological procedures in the thesis are demanded for a scientific approach. Therefore, we have followed standard referencing rules to avoid common mistakes, for example plagiarism. In addition, during the implementation there has been compliance with the, corresponding rules for reusing piece of codes.

4 Design

The SensibleThings platform is an all-in-all module for the Internet-of-things. The ultimate goal of the platform is to serve as a fully distributed overlay network. The dissemination layer of the platform is responsible for the look up, join/leave and resolving the address. In order to carry out these tasks as intended the Distributed Context eXchange Protocol (DCXP) has been utilized. Therefore, our target in these sections has been to design a suitable mechanism regarding how to smoothly perform context information dissemination in those cases when a nodes detaches itself from a wireless access point and joins another access-point.

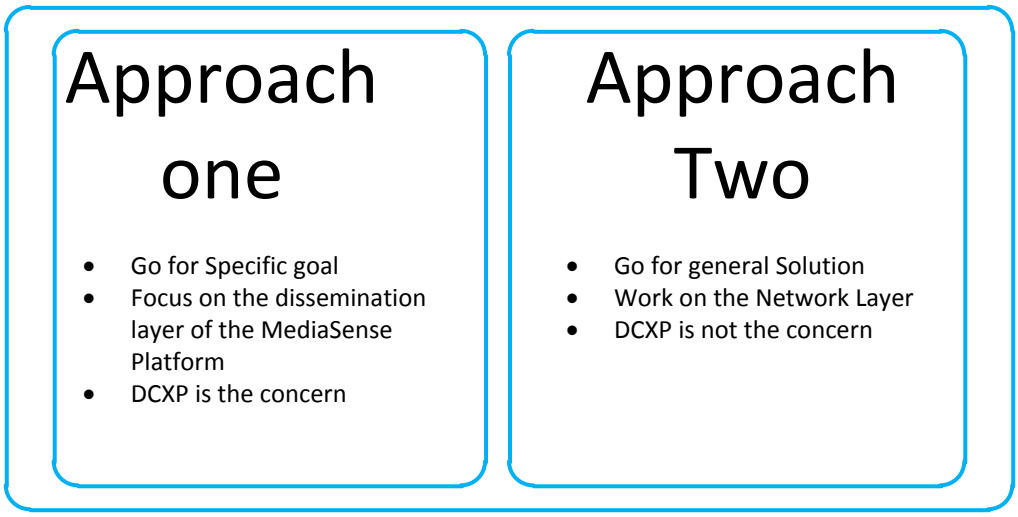


Figure 12 Generic approach and work around approach to prevent packet loss.

4.1 Proposed Mobility Solutions and Important Points to Examine

One of the root causes for problematic issues related to handover within access points has been the fact that the devices, addresses are topology dependant. It is not a mere random choice but the topology dependence nature of the device identifier offers efficient routing of devices within a network. Moreover, if the assignment of topology dependant identification is unable to assist in relation to smooth handover then it is convincing to add another address which is not topology dependant. So using the former address provides efficient routing within the network. However, using the latter address offers a smooth handover. Therefore, the

second approach in this thesis is to deal with the real root cause of mobility issues. The second approach works on the network layer of the devices. The implementation has been deferred for future work and only the proposal has been discussed. Furthermore, the general approach takes quite similar assumptions with the existing LISP protocol. However, the ultimate concern in the approach two is to find a solution for the smooth messaging on the SensibleThings platform rather than to work for a general solution.

The first approach is to circumvent or at least reduce data loss during handover on a DCXP protocol. So this approach takes into consideration that data loss that occurs in the lower layers. However, if a node acknowledges on receipt of data, then it could be possible to keep the node updated as long as the device stays connected to the overlay network. This approach in effect does not keep the node connected during handover rather it is concerned with on the update of information and prevent data loss.

Approach one is more of optimization work than quite a new approach to prevent packet loss during access point switch. In the existing system at a time when a node undergoes handover, the messages arriving at that particular time will be lost and the system returns the message, unreachable error. However, approach one provides a little bit more time and opportunity until the message becomes obsolete. Therefore, as compared to Roger Norling's**Error! Reference source not found.** implementation approach one does not throw *DestinationNotReachable* error immediately after missing the destination but it does provide a little extra time. However, at the end if the node is missing and is not a matter of handover or some kind of short disconnection due to the nature of radio link then it throws *DestinationNotReachable.*

4.1.1 Extended Comparison of the two approaches

This subsection presents the extended comparison of the two approaches mentioned above. See table 2 below.

Table 2 Extended Comparison of approach one and approach two

	Approach One	**Approach Two**
Features	Implemented on the dissemination layer of the	Implemented on the network layer of the

	SensibleThings platform. In this case additional primitives are proposed. Moreover, approach one has nothing to do with the lower layer protocols.	five layer model. This approaches attacks the problem from the root cause and this could be extended for other platforms.
Scalability	Approach one fully accomplishes its intended real task only if the latency taken by Mobile IP is too small to outdate the relevancy of a message intended to be transmitted to a disconnected node. So if the message being transmitted could be retransmitted within the time slot allocated then Approach one is fully Scalable.	Approach two is a kind of solution similar to LISP but follows its own implementations and ideas. Therefore, in this case the concern is not just to reach the messages to the intended node but also to avoid any packet drop that may occur when handover occurs
Packet drop	Yes. packet drop occurs This is more of an optimization task. Packet loss is expected but less as compared to the current implementation.	Packet drop occurs but is very minimal and scalablity of mobility is increased significantly. See LISP in section 2.2.2
Advantages	Easy to implement and could be used as a work around solution. In this case, a little modification on the existing SensibleThings platform could suffice.	Better Solution as compared to the other approach. It could be used in other similar platforms.
Disadvantages	During an extended delay, the relevancy of a message	Not easy to implement and takes longer time

	to be retransmitted becomes outdated so that means even if it is possible to retransmit the packet the messages are old enough to be considered irrelevant. So eventually, a lot of packet drop occurs.	from design phase to testing phase as compared to the other approach. So this approach can't be considered for implementation in this thesis
Related works and Expectations	We expect using this approach to increase the scalability of SensibleThings a little bit further if not to 100%. The achievements are presented in the Results section.	Related works for example LISP,HIP. The expectations in this case would be quite similar protocols as in these existing but our case would work on optimization.

4.2 Approach one – Focus on Dissemination of Context Information

In this approach the target has only been to determine out a solution for the smooth dissemination of messages when handover in a wireless network occurs and the belief is that DCXP is the only source of problems. So the problem has to be treated on the dissemination layer. The fact that the dissemination of information occurs in real time carries a meaning it is only within a certain period of time the difference that information is considered valid for dissemination. Based on this fact the question arises *"Is it not possible to retry dissemination of packets that have been lost due to handover if the context information is valid?"*Well if that retransmission is possible it means that there is a chance that in the retry the peers shows up in another wireless access-point within the threshold of validity of the information and is able to confirm its legitimacy. The exact answer for question entirely depends on the structures of DCXP.

4.2.1 DCXP

DCXP is a protocol for a peer-to-peer real-time context data exchange, in other words, real-time context exchange refers to the communication

delay which is insignificantly very minimal and extended latency is not tolerable. Besides, radio link disruptions and issues related to packet loss are handled through the MDP (Mobile DCXP Proxy). Furthermore, the delays for exchange of messages on DCXP are expected to be a few seconds.

However, in a real environment, mobile devices do not recieve continuous and smooth access to the peer-to-peer network. In real environments, handover takes place during mobility and which means, in most cases, packet drop occurs.

4.2.2 MDP

MDP acts as a server to shield the DCXP network from packet loss due to its radio link nature. Thus MDP is part of the DCXP network which is the actual concern of the thesis. It has been explained in section 2.6 that MDP is a server where each of the nodes is required to register when they join the peer-to-peer network. MDP could be considered as a node by itself but having a much better computing capability as compared to the real nodes. Thus here there is a server which could possibly prevent packet loss somehow. However, such an idea of employing a server as one option to work on the packet loss due to radio link nature is not scalable but only ensures mobility. Therefore, the secondary function of MDP required to be addressed in a distributed manner not by a centralized severer if the need is to achieve scalable mobility. The thesis detaches the task of mobility from the MDP server and proposes a distributed solution to achieve mobility as well as scalability at the same time. However, the primary task which is to register nodes and act as the main computation center is left unchanged in the proposed modifications though this would mean there is still scalability friction because of the presence of a central server.

4.2.3 Mobile DCXP (MD) – Our Solution

MD is a distributed approach to handle the problem of packet loss arising due to the nature of radio communication. Each of the nodes is responsible for retaining the mobility of nodes. Besides, such a system is scalable as compared to the previous solution. The mechanism in this section is to introduce additional primitives that help to trace out the fate of a message which has been disseminated. Accordingly, if a packet ends up nowhere due to handover, then retransmission must to follow

or if the packet successfully reached the destination then there would be no retransmission. See the pseudo code below.

```
Node node;
Message message1, message 2, acknowledge;

// Function 1
function Disseminate (Message m, Node[] nodes)
{
// send messages to all the nodes in the member list.
}

// Function 2
function Redisseminate (Message m, Node[] nodes)
{
// Nodes=  unacknowledged nodes
if (message is not obsolete)
// send messages to all unacknowledged nodes
retry Disseminate(m,nodes);
else
//if the message is obsolete due to extended handover delay or due to the node itself,
don't retransmit
return;
}
```

4.2.4 Proof-of-Concept

In the proposed solution the real aim is to carry out retransmission of messages in a proper time to those nodes going through handover and thus experiencing packet loss. The real headache is that the messages become obsolete and irrelevant in short period of time. Besides, as already mentioned, retaining a smooth follow of data is unthinkable as long the device can be disconnected during the handover. In other words, this would mean if the latency of handover is greater than the time for the message relevancy, then our proposed solution does nothing more than prevent packets loss and makes it available at improper time. However, since the time taken to switch from one access-point to another is very minimal, approach one could be an alternative work-around solution. As illustrated in figure 13 four active nodes and two access points have been shown. In addition, the dotted red line demarks the point where a possible packet drop out occurs. Nodes 1,1',2' are static and node 2 is on move. As can be seen in figure 13 c node 2 jumps on the red line and become disconnected from the networks. However,

as it moves further to the left (figure 13 d) it becomes connected to the network again. The time that must be ascertained in this example is the period of time t between d and b. If, for example message m destined to node 2, had been dropped in the time t, then the source node does not receive an acknowledgment so it needs to retransmit as far as the message is relevant. The relevancy of a message matters for protocols such as DCXP where the real target is to disseminate context information which is continuously evolving in most cases.

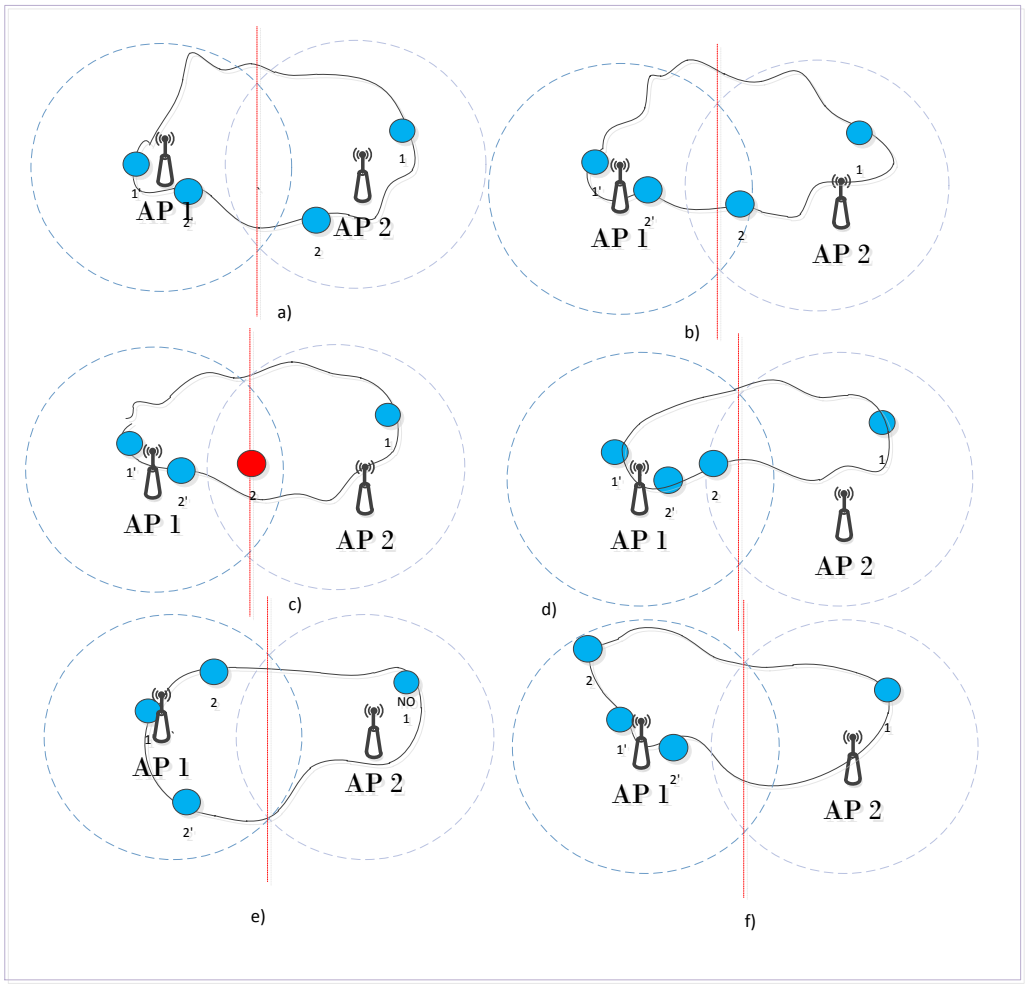


Figure 13 Handover

In figure 11b, node 2 is on the verge of disconnection from access point 2 .The assumption is that immediately after node two is no longer connected and the time is T1. Furthermore, node 2 is get connected with another IP adress possibility but the same ID through access point 1 at time T2. Therefore, the time taken for the proper handover in this case is T= (T2-T1). If node 1, for example, was attempting to reach node2 in

figure 11b, the packet loss will inevitability occur. However, the solution in this case is that node 1 keeps sending to node 2 until the node shows up as shown in figure 11d. The limitation on the number of retries could be based on the maximum time the handover might take or the deadline of the message relevancy.

4.2.5 Mobility Extension on the SensibleThings Platform

The SensibleThings platform is a fork project from the noble MediaSense platform. The MediaSense is an end-to-end platform for the internet-of-things. DCXP takes the core task for context data dissemination. On the SensibleThings platform presented in section 2.5, extension and optimization work should be carried out on the Add-in layer thus retaining the entire architecture unaltered. Since the intention in the proposed solution is to add one more primitive into the existing list of primitives, then the modification could have been done conveniently on the dissemination layer and more specifically on the dissemination core.

The existing system (as shown in figure 14a) throws a destination unreachable error in the case when the destination node is not accessible because of a number of reasons including a very short disruption due to access point handover. Therefore, in this case the packets are dropped and the destination node is not able to obtain the message even if the switch takes a few seconds.

However, in the proposed modification the source nodes send the message again and again until the current messages become outdated or until the node receives the message. Once the message arrives at the destination node, an acknowledge message is sent back to the source node (See figure 14b).

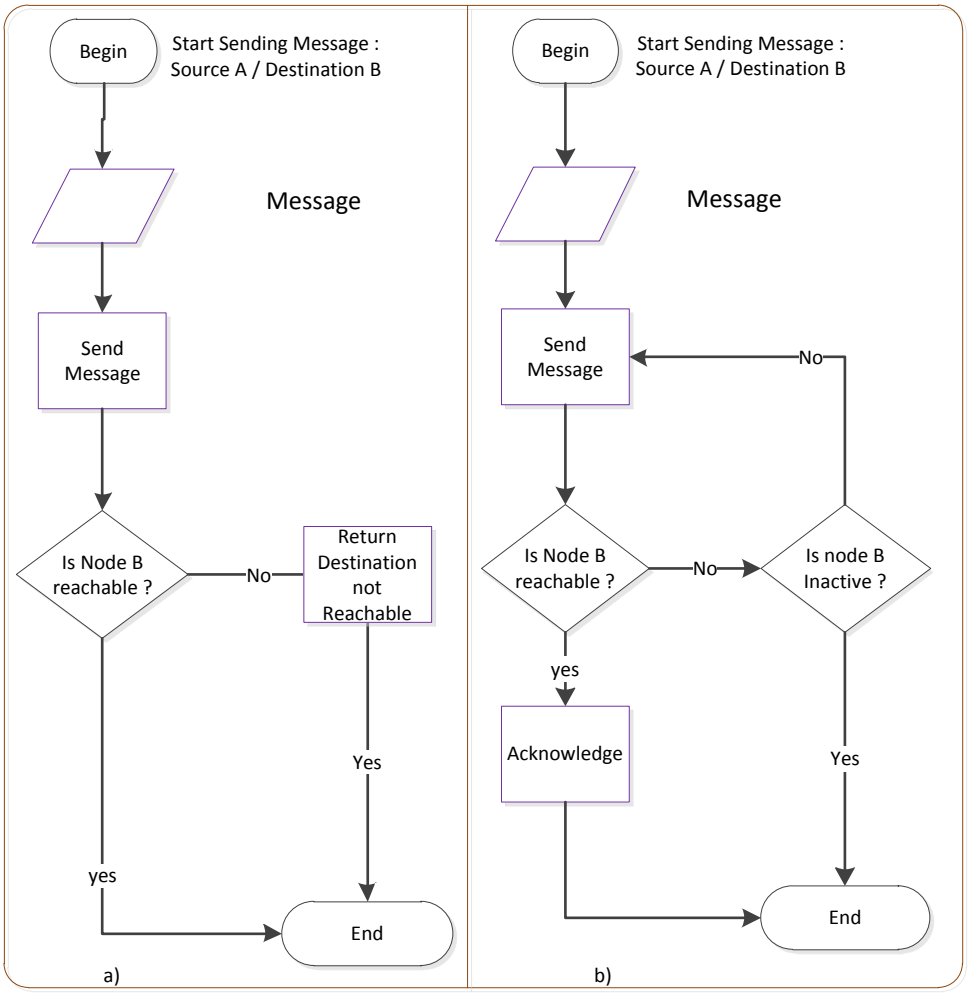


Figure 14 The SensibleThings Message Forward Flowchart a) the Existing Message Forwarding as shown using flowchart b) the Proposed modification to enhance scalable Mobility

4.3 Approach two

The second approach solves the problem from the root. The cause of the problem has been the well known issue of latency and packet loss in the existing wireless network handover mechanisms. Therefore, in this section we propose a general approach for the SensibleThings platform. The implementation of the second approach is deferred for a future task but we have explained all the necessary techniques in this section.

4.3.1 Mobility Solution

We believe that the observation made by the Location Identifier Separation Protocol proposers is the suitable beginning to start with the scalability problems in the existing SensibleThings platform. As shown in figure 13 The device could be in one of the positions as shown on the

position 1, AP1 could only provides the service, however, on positions 2 and 3 both AP1 and AP2 provide the service. Moreover, in position 4 only AP2 could provide a service. So in this diagram on the positions 2 and 3, there is a session disconnection problem as the devices move to the other end. If we give each device two addresses i.e. one topology dependant and another topology independent address, it is possible to have smooth a handover in positions 2 and 3.

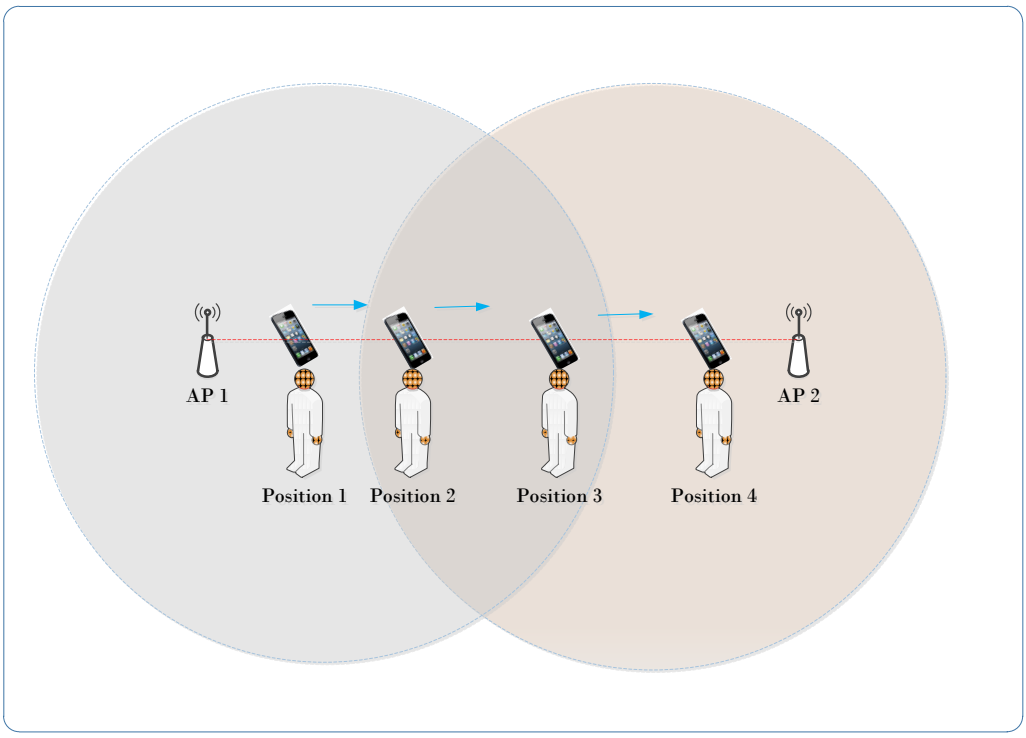


Figure 15 Handover

4.4 Summarization

In this section the proposed solution, alternative solution and the existing system have been discussed briefly. Moreover, a comparison of approach one/ proposed solution and approach two / deferred generic solution has been explained. Furthermore, the proposed solution named Mobile DCXP (MD) has been presented and a proof-of-concept for MD has been shown in detail.

5 Implementation

In this section of the report the implementations of what has been named the Mobile DCXP or MD have been briefly presented. Figure 4 below shows the implementation conducted in this section on the top of the existing SensibleThings platform core. The IChat Android application has been developed to evaluate the Mobile DCXP proposed. Therefore, IChat runs on the top of the SensibleThings platform. Furthermore, the extensions added on the Add-in Layer of the SensibleThings have been used in the application.

In addition, the proof-of-concept explained in detail in the Chapter four has been implemented in this section. Therefore, the proof-of-concept app has been intended to show the concept of Mobile DCXP more clearly.

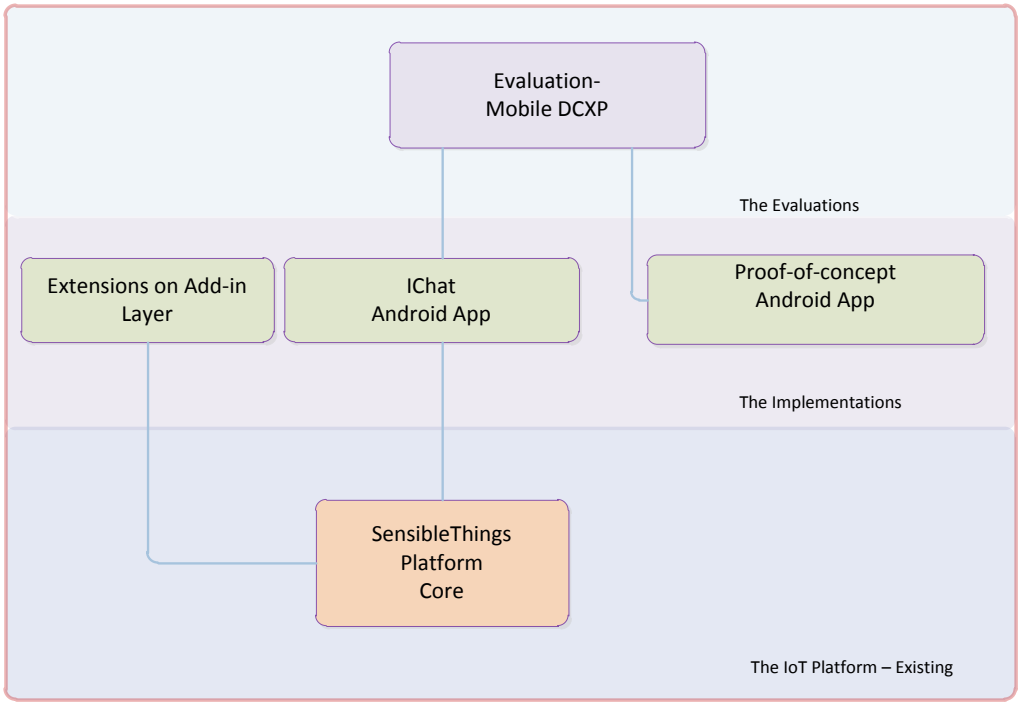


Figure 16 Overall diagram of separation of tasks and the implementations carried out

All subsections of the implementations shown in figure 14 have been explained in detail in this Chapter. However, the evaluations section has been presented in Chapter 6 and Chapter 7.

5.1 Extensions on Add-in Layer

In the add-in layer we have added functions that could enable '*DestinationNotReachable'* error to be thrown not just after the first transmission failure but after going through continues retries within the given time period.

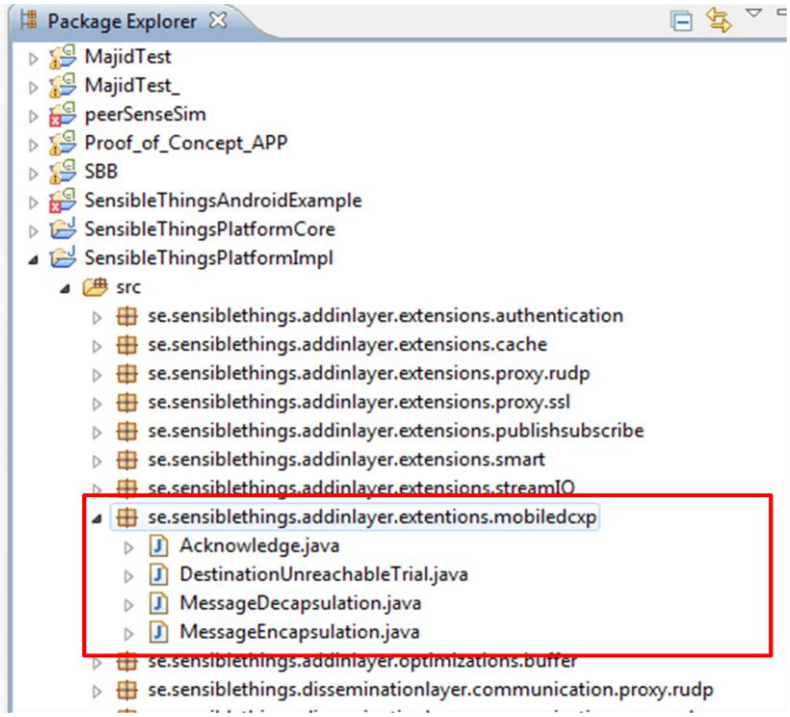


Figure 17 Extensions on Add-in Layer

Functions and Codes Added

Acknowledge:

Acknowledge is the sixth primitive we have proposed to have a better scalable mobility. This function performs an exactly similar task as in the Message.java but the difference is *Acknowledge* is intended to confirm whether the packets have arrived.

DestinationUnreachableTrial:

In the existing system when a node is disconnected all of a sudden an error "*DestinationNotReachable*" is thrown and after that even if the disconnection is just for an instance as in the case when this occurs during the handover, the destination node does not have a second chance to receive a message. In the TCP socket when there is a problem with the destination of a packet then an error is caught and thus han-

dling this error has been properly conducted in the existing implementation, however, it is not efficient. The work we have done in this part could be categorized under optimization as the real need is to handle the *DestinationNotReachable* error in such a manner it would optimize the efficiency of the message dissemination in time where there is high mobility.

MessageEncapsulation:

The code developed in this section is to send messages having another format which is the message to be transmitted in the existing SensibleThings Platform. So the task in this section is to add additional sections to the Message and then retrieve the extra information on the end without changing anything on the existing System.

MessageRetrival:

In this part of of the code the "Decapsulation" is performed at the other end. So the extra information attached to the message is retrieved and dealt on accordingly.

5.2 IChat Android Application

IChat is a simple Android application developed in order to study and determine out the advantages of having mobile DCXP on the existing SensibleThings platform. So IChat will be connected to the existing SensibleThings Platform and the packet loss will be examined as well as following the same procedure will be followed with mobile DCXP included on the SensibleThings platform. The Very IChat application does not offer an advanced idea but it is just an ordinary chat application having additional an auto generated message exchange (or chat). Therefore, the application allows the amount of packet loss to be studied with and without mobile DCXP.

5.3 Proof-of-Concept App

The proof-of-concept application illustrates the concept of the Mobile DCXP shown in the diagrams in section 4.2.1 in a clearer manner. The application in this section simulates the events happening during handover between two wireless access points and four connected nodes. Therefore, the messages exchanged and the retransmission trial can be observed more clearly.

6 Results

The results section illustrates in detail the experimental results using graphs and diagrams, and the developed applications (IChat and proof-of- concept applications). In this part of the thesis, comparisons of existing system with the proposed system have been presented in detail. The improvements achieved using Mobile DCXP have been presented in figures. In addition, the extensions developed to achieve Mobile DCXP (MD) have been explained in detail.

6.1 Comparison and Evaluation

To examine the performance gain of the proposed system over the existing system with regard to the packet loss, an experiment. In the experiment has been conducted, we assigned unique numbers (Identifications) for each of the messages exchanged. The identification could help to array the incoming messages as well as providing information regarding which packet is missing. Therefore, the two systems have been tested as the signal strength of the first access point becomes low and eventually dies. In this case, there is one important factor which is the time of relevancy of the message. In other words, during the period handover occurs the messages intended to reach the node might no longer be important. When the message of interest is the context information coming from underlying sensors and sensory networks, the experiment should consider the context information relevancy factor. Therefore, in this case since the SensibleThingsPlatform is intended for real time context-exchange, after a couple of extended trial it is necessary to stop sending the message. So in this case if the node is taking longer than the maximum time limit for real time communication delay then there happens the inevitable packet loss occurs. In fact there was the slight chance that a packet loss might occur in the proposed system. However, based on the kind of applications, the retransmission of the lost packet for a further extended period of time is better by far than just leaving the lost packet and turning to the new packets.

In this experiment we used the network (the SensibleThingPlatform with DistributedLookup intended for the connections without NAT transversal). Furthermore, the network (see Figure 18) we set up for the experiment consisted of two tablets (Samsung galaxy tab 7.0 Model

number : GT-P3110), AP TL-MR3420 and (XTouch Model number: X704) as well as a laptop to examine the packets coming in and going out.

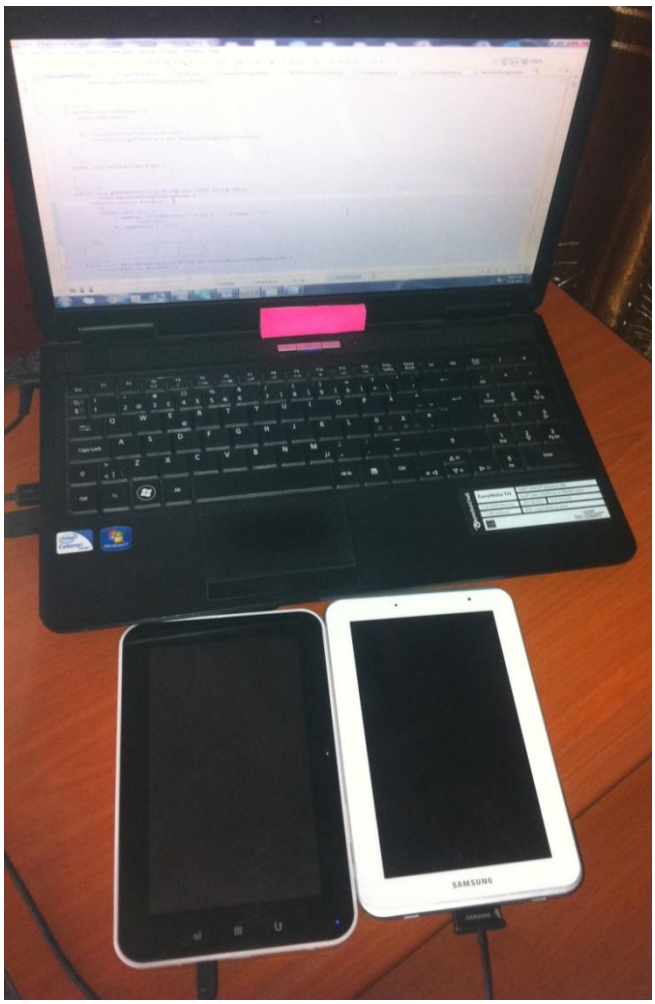
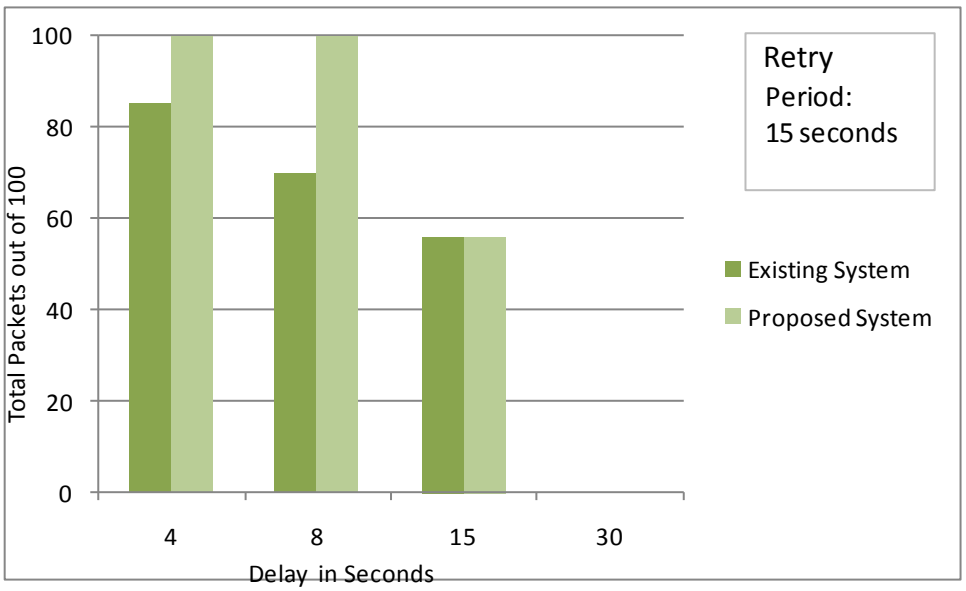


Figure 18 the Peer-to-peer network.

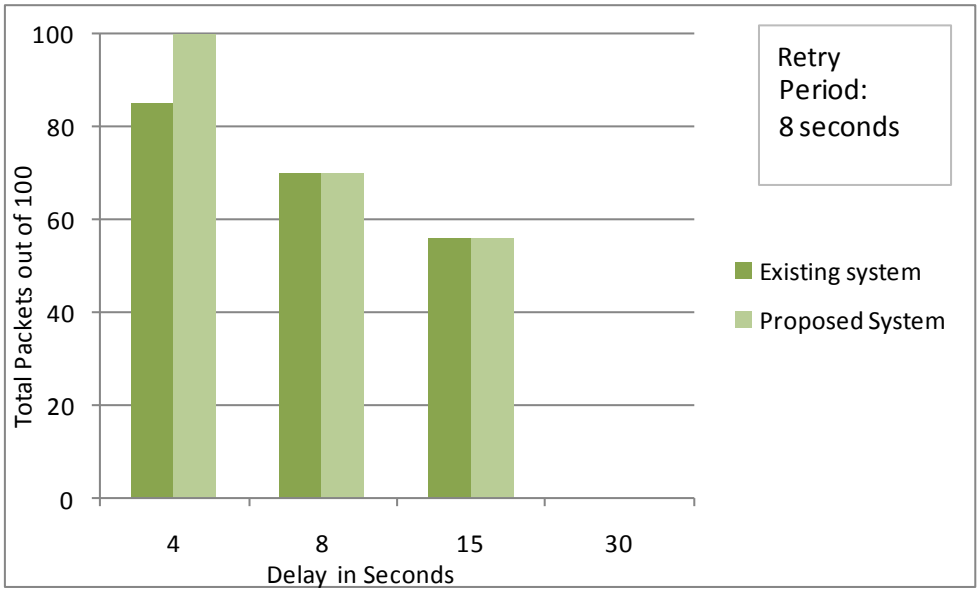
6.1.1 Packet Loss

In order to count the number of packet losses in the proposed and existing systems, automated chatting session is set up while the second device is roaming.

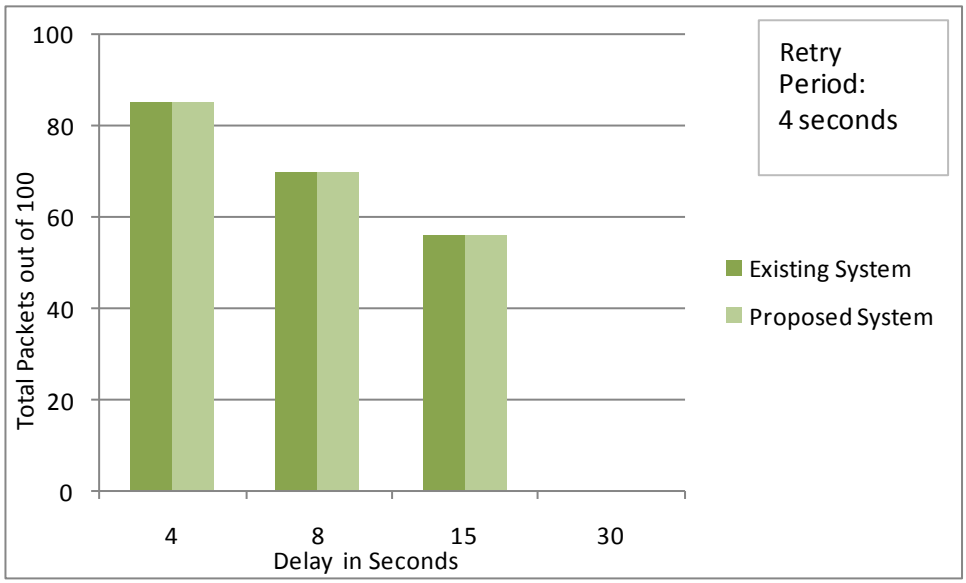
In the existing system, packet loss does occur and there is no way to stop that. However, in the proposed system each peer retries the transmission of lost packets again and again until the message becomes obsolete. In this case, the measurement of the probability of packet loss as it happens in the extended delay has been shown. So we have collected the percentage of packet loss for delay times of 4sec, 8 sec and 15sec. A total number of 100 messages have used in the study and each message is assigned a unique number.


100
80
60
40
20
0
Total Packets out of 100
4
8
15
30
Delay in Seconds
Retry Period: 15 seconds
Existing System
Proposed System

a)


100
80
60
40
20
0
Total Packets out of 100
4
8
15
30
Delay in Seconds
Retry Period: 8 seconds
Existing system
Proposed System

b)



c)

Figure 189 Bar Chart illustrating the packet losses in the existing system and proposed system.

The charts in the figure 19 show the gains in the performance of the proposed system over the existing SensibleThingsPlatform. The result clearly shows that as delay of Wi-Fi handover increases the gain of the existing system decreases. Though it could have been a possible solution to consider further extending the retry period that does not fulfil the requirement of the *Sensiblethingsplatform*. Therefore, the probability of packet loss has been decreased but a breach in a relation to loss packets still exist in the cases of extended time taken by the handover. However, under normal circumstances it does not take a longer period than that mentioned in this research.

6.1.2 Jitter

In this experiment jitter of connection between two devices used in the experiment has been shown. In the methodology section, it was mentioned that jittering has been included as it is one of the quality measurements for the network however since the modifications are entirely done on the application layer, no difference is anticipated between the two systems. See figures 20 , 21 and 22.

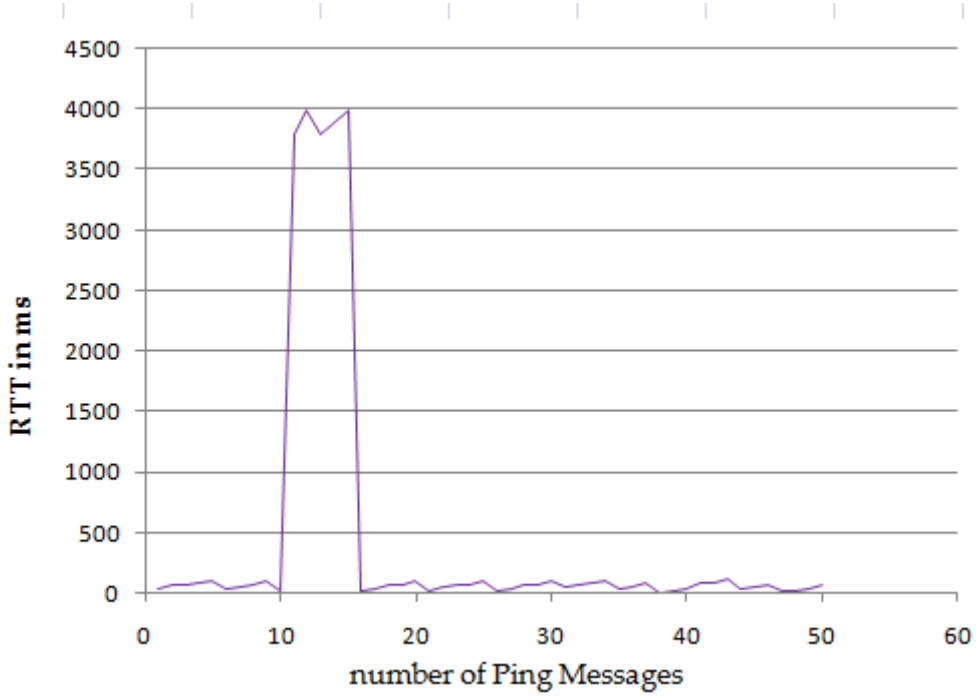


Figure 20 Jitter : Existing System

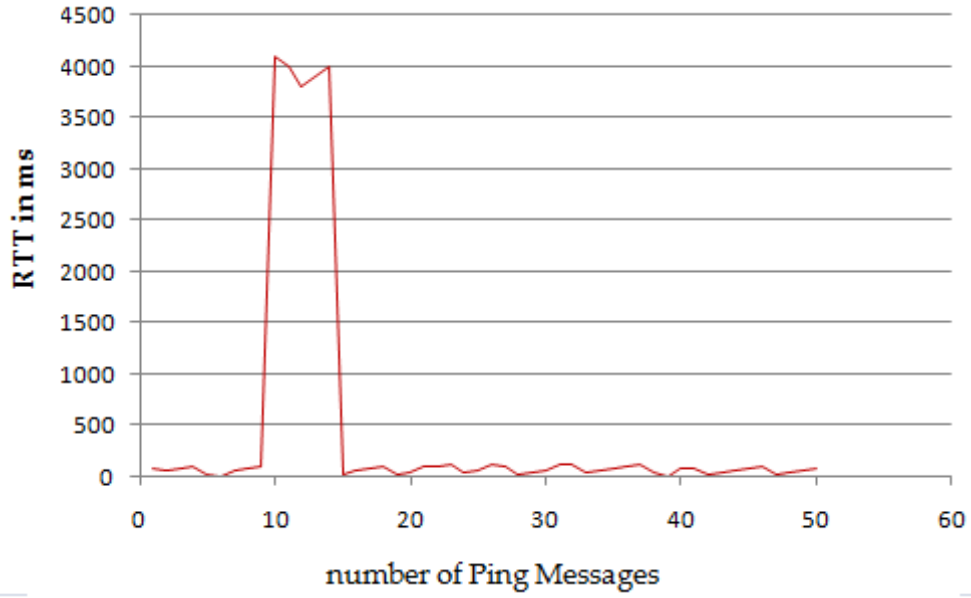


Figure 21 Jitter: Proposed System

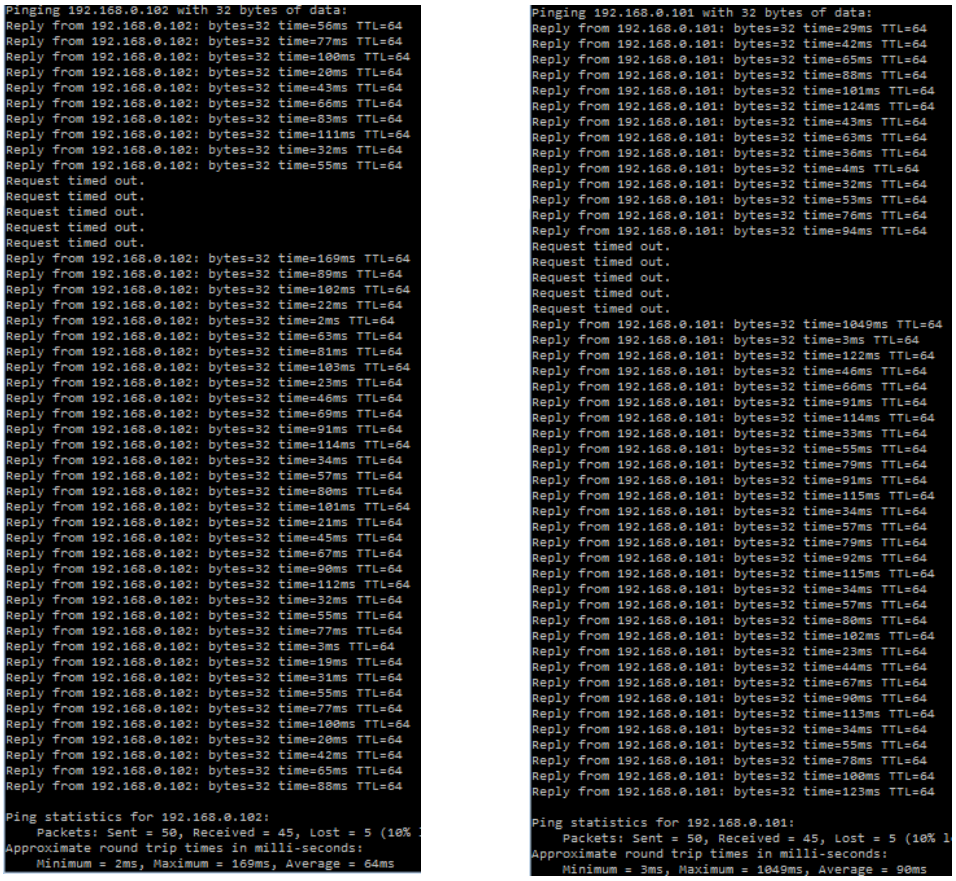


Figure 22 RTT times for the existing and proposed system

6.1.3 Subjective Testing

Results from subjects who participated in the survey suggests there is a real performance gain as compared to the existing system. The exact figures for the subjective testing are shown in the following diagram: figure 23

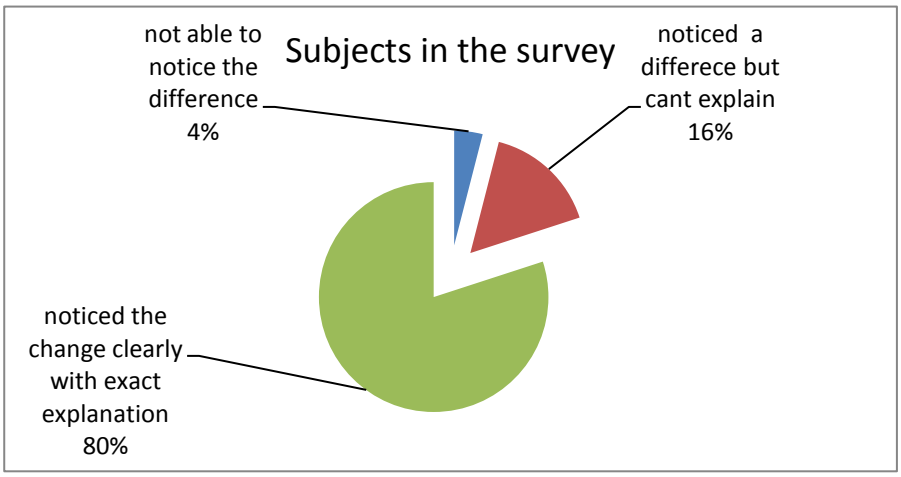
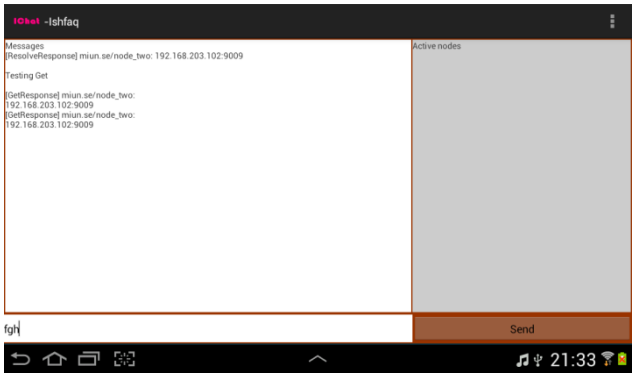


Figure 23 Survey result

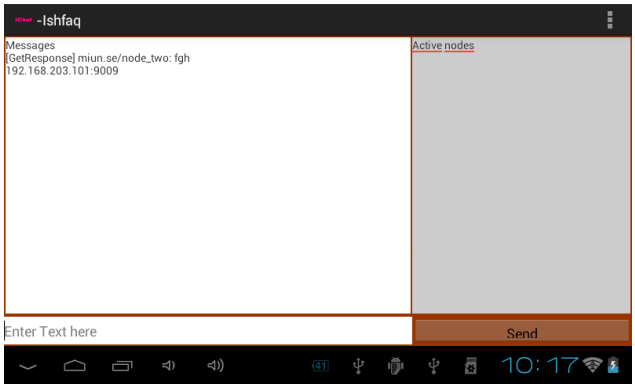
The subjective testing results confirm that the changes done on the *SensibleThingsPlatform* have indeed brought a performance gain. As shown in the figure, 96% of all the subjects have noticed the difference. Furthermore, 80% of all the participants pinpointed the modifications in the proposed system.

6.2 Application Developed for testing purpose: IChat Android app

The IChat sends and receives messages as in any minimal chatting applications. It is an Android application which allows the user to strike the p2p communication, the intended purpose of IChat is just to help us test the Mobile DCXP and compare it with the existing SensibleThingsPlatform's Proxy Server. Therfore, the basic features of the chat application have been added to the IChat as shown in the figure below. The area labeled by the "Active nodes" takes the two peers (a bootstrap which is device1- Samsung galaxy and the normal node which is XTouch tab) used for testing.

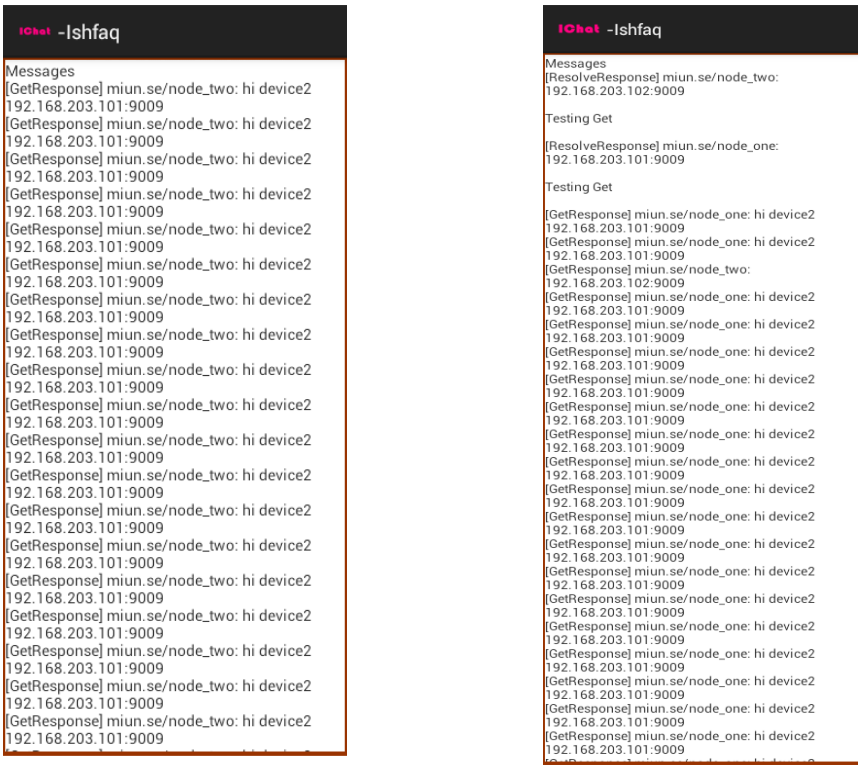


a)



b)

Figure 24 IChat Android based testing application

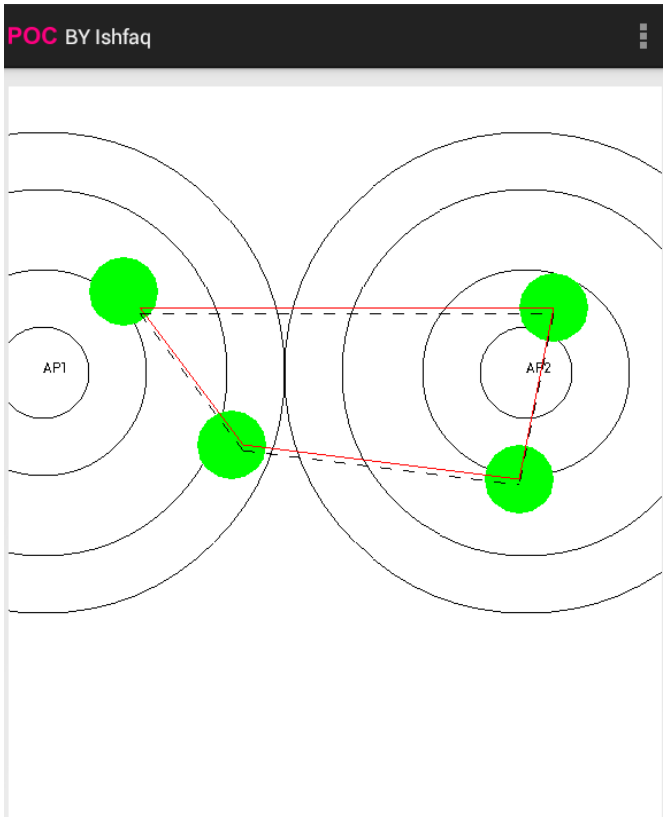


a) b)

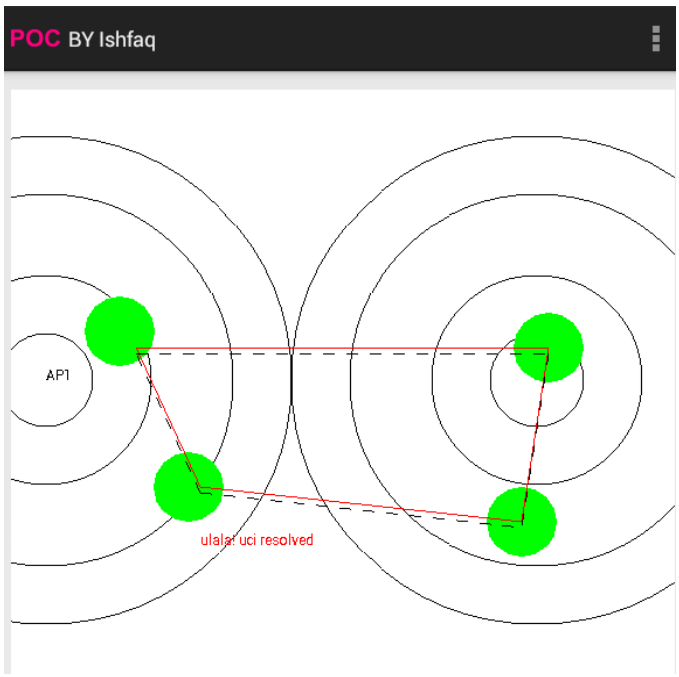
Figure 25 Two nodes showing the messages exchanged between them. a) the bootstrap node whose ip address is 192.168.203.101 b) the other node which shows the normal node whose ip address is 192.168.203.102. In this figure the bootstrap node send a message "hi device2" and both of them will receive that message.

6.3 Proof-of-concept Application

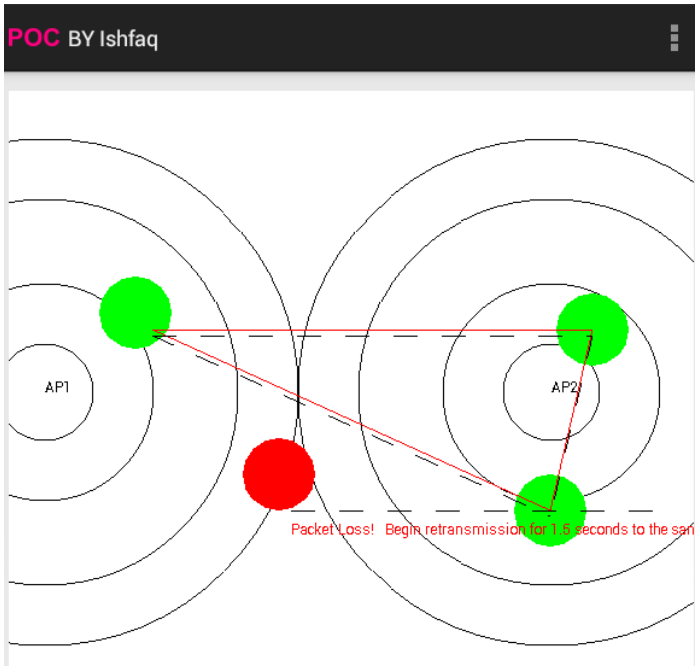
In this section the proof-of-concept application has been shown with different states. The implementation performed in this part of the report is directly taken from the detailed explanation of the report presented in section 4.2.1.



a)



b)



c)

Figure 26 Proof-of-concept applications

7 Conclusions

The tasks clearly outlined in chapter one and carried out in this research wok could rather be presented as optimization work. This is because the solution presented in this thesis work could exactly be presented as a workaround solution to arrive at a better mobility for the nodes connected over the SensibleThingsPlatform.

We have proposed two solutions for the research questions mentioned. The workaround solution of Mobile DCXP (MD) as it is referred to specifically has been chosen to show the performance gain over the currently existing version of the platform with regard to packet loss. However, the other solution which could handle the problem from the root cause and which could yield a better result has been deferred for future work.

The tasks we have conducted in this research work could be summarized as in the following relative to what has been outlined for in the proposal and the introduction chapter.

Goal One Achievement:

In the chapter two of this report existing mobility solutions have been presented briefly. Host Identity Protocol, LISP, Mobile IPv6, Hierarchical Mobile IPv6, Fast Handover,MOBIKE, PMIPv6 have been studied. Moreover, the SensibleThingsPlatform and its functionalities have been presented in detail.

In chapter four of this research work solutions to handle the research question have been proposed and thoroughly discussed. In this chapter two approaches haven forwarded and compared. Proof-of-concept apps and diagrams have been used to illustrate the concept of Mobile DCXP (MD).

Goal Two Achievement:

In chapter five the Add-in layer extensions have been implemented and the techniques have been shown. In addition, in the implementation chapter a brief explanation has been presented on the application used to evaluate the proposed solution.

Goal Three Achievement:

In the results section we have carried out experiments to find out the size of packet losses in the existing system and the proposed system. In this section the IChat Android app, and proof-of-concept app have been presented in detail.

Overall Aim Achievement:

Although the Mobile DCXP (MD) is a scalable mobility solution which could rather be considered as an optimization work, the results shows that we have achieved a better performance. Therefore, observing the goals listed in the Introduction chapter one by one, the aim has been achieved. In Chapter four we have presented an alternative solution which is a rather big project (for example as in LISP) and thus might not fit into an AV level course. So considering the time and resources allocated it is only possible to choose the other alternative.

7.1 Discussion

Mobile DCXP or MD has been proved to offer performance gain as compared to the Proxy Server in the existing system. The fact that this app has not been tested in an environment where there are many of disconnections and connections occuring may not change the result arrived at but in that case it could be necessary to handle the issue related with changing IP address.

Moreover, the writing of the codes does not take more than what has been allocated in the time plan however connecting and bringing the app to a working state has been very problematic. The main reason being the size of the code for SensibleThingsPlatform and the networking that was required to be set up. The documentations for users interested to communicate on the platform is not sufficiently detailed. The other bugbear and holding back condition has been when setting up the communication with platform within the LAN, the code directly goes to the ***RudpProxycommunication*** rather than ***Rudpcommunication*** that part took longer to trace and finally we had to disable one of the conditions See the piece of code below.

The other difficulty was to trace out a bug and the time the platform takes to converge in cases where there is a problem with the underlying networking hardware. See the figure 27 below for problems encoun-

tered in this regard. The application does not throw an error but remains busy which misleading as if converging is taking place when there is an error in the socket.

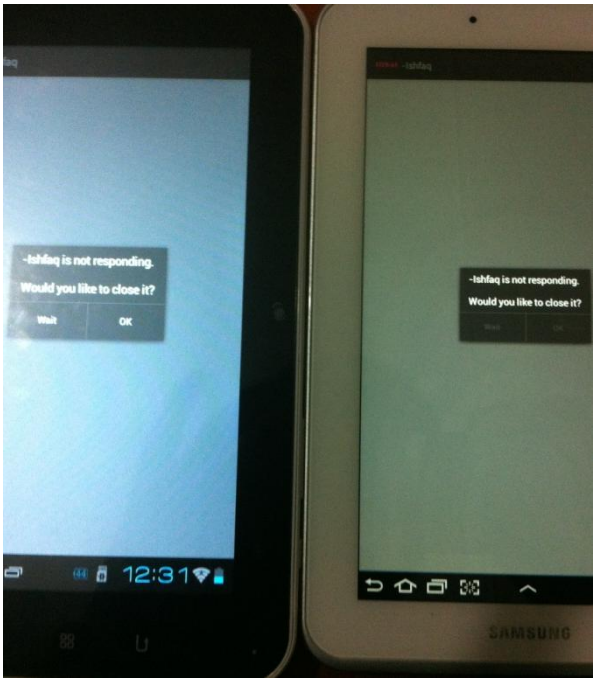


Figure 27 indefinite time for convergence. Not hint to trace out the actual error.

```
 * The SensibleThings platform itself, which exposes all functionality towards
 * the application developers.
 *
 */
public class SensibleThingsPlatform {

    private DisseminationCore disseminationCore = null;
    private AddInManager addInManager = null;
    private SensorActuatorManager sensorActuatorManager = null;

    /**
     * Initializes the SensibleThings platform. Must be called before using the
     * any other functions. This is now the suggested way to start
     * SensibleThings. It uses normal DHT lookup and RUDP. But switches to Proxy
     * automatically if you are behind NAT.
     *
     * @param listener
     *            The SensibleThingsListener, for all callbacks
     */
    public SensibleThingsPlatform(SensibleThingsListener listener) {

        //if (isBehindNat()) {
            // This forces Proxy when behind NAT
        //  initalize(LookupService.DISTRIBUTED, Communication.PROXY_RUDP);
        //} else {
            // Or else, normal DHT and RUDP
            initalize(LookupService.DISTRIBUTED, Communication.RUDP);
        //}
```

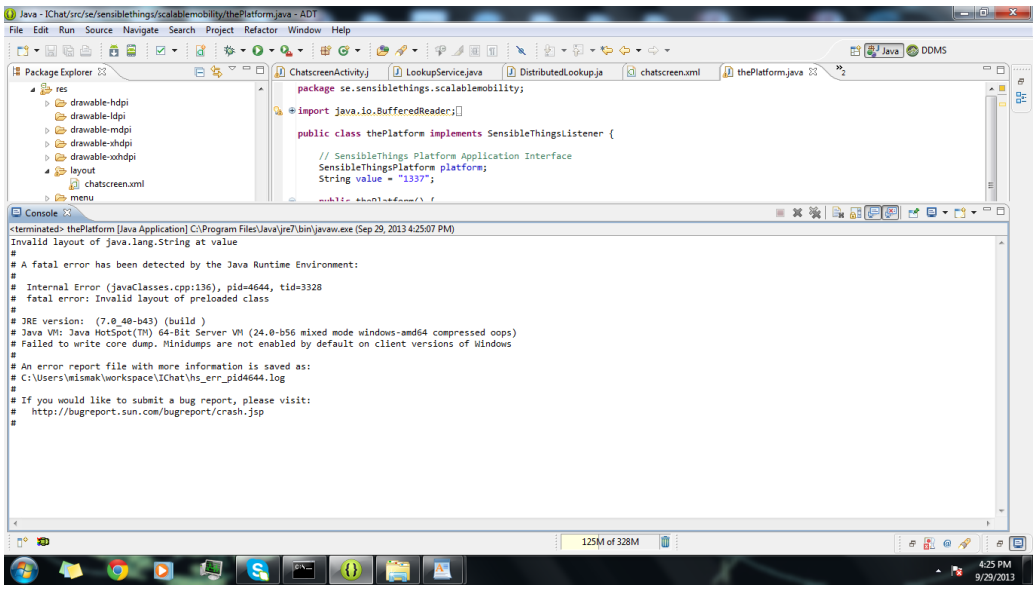


Figure 28 errors

7.2 Scalability

The *MediaSense* platform from which the *SensibleThingsPlatform* as a fork has been proposed to meet requirements which have been listed as core requirements for Internet-of-things Platforms: Scalable , fully distributed, fast, current, lightweight, stable and extensible. Furthermore, scalability of *Mediasense* overlay network is entirely dependent on the underlying lookup service built on the DHT and therefore, modifications at the application layer does not impair the existing scalability of the *SensibleThingsPlatform*. As a result, the size of nodes could not affect the performance of the proposed solution.

7.3 Contribution and impact

The actual work carried out in this research work contributes immensely both to the existing system and for future research looking to pursue with a better and generic scalable mobility solution for the SensibleThingsplatform. So for the existing system we have achieved a better performance gain with regard to packet loss during Wi-Fi hando-ver and other short radio link disruptions. In addition, in the long run the thesis has laid down some necessary information for future re-searchers and students.

7.4 Ethical Deliberations

The results of the thesis work contribute immensely to developers interested to build applications on the top of SensibleThingsplatform. The reduction of packet loss supports the apps running on the top of

SensibleThingsplatform to deliver optimal performance as compared to the existing System. In addition, the thesis recommends a common solution to increase the scalability at the network layer as in LISP. This could decrease the extra time needed to implement a protocol similar to LISP for SensibleThingsplatform.

Moreover, throughout the entire thesis work due considerations have been given to conform to copyright rules and thus avoid plagiarism.

7.5 Future work

The ultimate future work related to this thesis has been as mentioned again and again throughout the report the implementation of approach two presented in chapter four. In this approach a kind of scalable mobility solution similar to LISP is worth considering in the future.

In addition, some other smaller tasks which have not been tested within the scope of this work for example, testing the Mobile DCXP within environments where a number of nodes exist and some other factors could be part of the task worth considering.

Finally, during the implementation of the IChat android app we have found out that applications and even the ***SensibleThingsPlatformExample*** App which could be downloaded for testing purpose from the www.SensibleThings.com is a little bit unstable in the case of network error and keeps the resources indefinably busy unless forced to stop the process. In this case the app does not crash and an exception is not captured either. Exception should have been caught after a couple of tolerable resource busy staff. This very problem could have come from the time the platform takes to converge as we have found out later on in the properly working system. If that is so, this problem could be expressed in brief as "indefinite time to converge" which occurs during network error. This could be one more task worth considering for future work.

References

[1] Wireless Networking Basics, Chapter 2, Version 1.0, NetGear Inc. America Parkway, CA95045 USA, Tutorial published on 2005

[2] Dr. Pekka Nikander,"Evolution of Networking: Current Problems and Future Directions", Published in IEEE SecureComm 2007- Third International Conference on Security and Privacy in Communications Networks, at Nice, France, 17-21 Sept. 2007.

[3] P. Nikander, A. Gurtov, T. R. Henderson "Host Identity Protocol (HIP): Connectivity, Mobility, Multi-Homing, Security, and Privacy over IPv4 and IPv6 Networks", IEEE Communication Surveys & Tutorials, Volume: 12, Issue: 2, published April 2010

[4] F. Al-Shraideh,"Host Identity Protocol", Published in International Conference on Mobile Communications and Learning Technologies, April 2006, DOI:10.1109/ICNICONSMCL.2006.112

[5] R. Moskowitz, P. Nikander, "Host Identity Protocol (HIP) Architecture" ICSA Labs and Ericsson Research Nomadic Lab, RFC: 4423, page 1, 16, Retrieved June 10, 2013

[6] Ved P. Kafle, Masugi Inoue, "Locator ID Separation for Mobility Management in the New Generation Network", Published in Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, Volume: 1, Number: 2/3, October 2010

[7] CISCO tutorial on Locator/ID Separation Protocol (LISP) Overview, online available on http://lisp4.cisco.com/lisp_over.html, Retrieved June 19, 2013

[8] D. Farinacci, V. Fuller, D. Meyer, D. Lewis, "The Locator/ID Separation Protocol (LISP)", Internet Engineering Task Force, ISSN: 2070-1721, RFC: 6830, January 2013

[9] Ping Dong, Hongke Zhang, "MobileID: Universal-ID based Mobility in Locator/ID Separation networks", IEEE International Conference on Communications and Mobile Computing, 2010

[10] http://www.cisco.com/web/about/ac123/ac147/images/ipj/ipj_11-1/111_lisp_fig1_lg.jpg, retrieved June 12, 2013

[11] Alberto Castro, Martin German, Macelo Yannuzzi, Xavi Masip-Bruin, "Insights on the Internet routing scalability issues", Advanced Network Architectures Lab, Universitat Politecnica de Catalunya (UPC), 2009.

[12] http://www.cisco.com/web/about/ac123/ac147/images/ipj/ipj_11-1/111_lisp_fig2_lg.jpg, retrieved June 12, 2013

[13] Damien Saucez, Luigi Iannone, Olivier Bonaventure, Dino Farinacci, "Designing a Deployable Future Internet: the Locator/Identifier Separation Protocol (LISP) case", IEEE Internet Computing, Published December 2012

[14] N. Montavont, T. Noël,"Handover Management for Mobile Nodes in IPv6 Networks" IEEE Communication Magazine, Volume 40, Issue 8, Published August 2002

[15] K. Das, "Mobile IPv6 "http://www.ipv6.com/articles/mobile/Mobile-IPv6.htm

[16] Hee Young Jung, Eunah Kim, JongWha Yi, HyeongHo Lee, "A Scheme for Supporting Fast Handover in Hierarchical Mobile IPv6 Networks", Published in ETRO Journal, Volume 27, Number 6, December 2005

[17] H. Soliman, C. Castelluccia, K. ElMalki, L. Bellier "Hierarchical Mobile IPv6 (HMIPv6) Mobility Management ", Network Working Group, October 2008, http://www.ietf.org/rfc/rfc5380.txt, Retrieved June 19,2013

[18] Sangheon PACK, Yanghee CHOI, "A study on Performance of Hierarchical Mobile IPv6 in IP-Based Cellular Networks", Published in IEICE Transactions on Communications, Vol. E87-B, No. 3, 2004

[19] Seonggeun Ryu, K. Lee, Youngsong Mun, "Optimized fast handover scheme in Mobile IPv6 networks to support mobile users for cloud computing", Published in The Journal of Supercomputing, February 2012, Volume 59, Issue 2, pp 658 - 675

[20] R. Koodli, "Fast Handovers for Mobile IPv6", RFC 4068, Nokia Research Centre, Network Working Group, November 2009

[21] S. Forsström, "Enabling Adaptive Context Views for Mobile Applications Negotiating Global and Dynamic Sensor Information" Mid Sweden University, published 2011, Page: 22

[22] P. Eronen, "IKE2 Mobility and Multihoming Protocol (MOBIKE)", RFC 4555 (Proposed Standard), Network Working Group, June 2006

[23] John Moore,"Secure and Seamless Network Handoffs", March 16, 2007, available online on: http://www.itsecurity.com/features/secure-and-seamless-network-handoffs-031607/

[24] S. Ryu, G. Kim, B. Kim, and Y. Mun,"A Scheme to Reduce Packet Loss during PMIPv6 Handover considering Authentication", Published in IEEE International Conference on Computational Sciences and Its Applications (ICCSA 08), June 30 2008, Perugia, Italy

[25] Mark Grayson, Kevin Shatzkamer, Scott Vainner,"IP Design for Mobile Networks", Chapter 6, ISBN-13: 978-1-58705-826-4

[26] CJ. Bernardos, "Proxy Mobile IPv6 Extensions to Support Flow Mobility", NETEXT Working Group, August 2013

[27] F. Guist, C. J. Bernardos, S. Figueiredo, P. Neves, T. Melia," A hybrid MIPv6 and PMIPv6 distributed mobility management: The MEDIEVAL approach", Published in 2011 IEEE Symposium on Computers and Communications (ISCC), July 2011, Kerkyra, Greece

[28] T. Kanter, S. Forsström, V. Kardeby, J. Walters, U. Jennehag, & P. Österberg, "MediaSense–an Internet of Things Platform for Scalable and Decentralized Context Sharing and Control", Published in The Seventh International Conference on Digital Telecommunications (ICDT 2012), April 29, 2012, Chamonix/Mont Blanc, France

[29] T. Kanter, S. Forsstrom, S. Pettersson,P. Osterberg, J. Walters, V. Kardeby,"The MediaSense Framework", Published in IEEE

Fourth International Conference on Digital Telecommunications 2009 (ICDT 09), Colmar

[30] J. Walters, Theo Kanter, R. Norling, "Distributed Context Model in Suppoet fo Ubiquitous Mobile Awareness Services", Second International ICST Conference, S-Cube 2010, Miami, FL, USA, December 13-15, 2010, Revised Selected Paper

Appendix A: Questions in the subjective Testing List

1. *Did you observe any differences of some sort on the behaviour of the two applications?*

    a. *No*

    b. *Yes.*

    If yes, please answer the rest of the questions properly

2. *Did you have any message you couldn't receive from the list of messages you need to receive?*

    Yes / no

    If yes, how many messages failed to receive?

3. *Did you get any unexpected error such as disarray of chat messages?*

    Yes/no

    If yes, could you mention how many times such incident occurred during chat session?

4. *Did you notice any differences when handover between hotspots occurs?*

    Yes/no

5. *Which application do you prefer to use if you are allowed to text chat with your peers as you do in facebook but no Internet cost?*