University “Politehnica” of Bucharest

Automatic Control and Computers Faculty,
Computer Science and Engineering Department

MASTER THESIS

A Critical Analysis and Application on the current state of Mobility in Wireless Networks

Scientific Adviser:
SI. Dr. Ing. Elena Apostol

Author:
Mititelu Ștefan-Cristian

Bucharest, 2016

Universitatea ”Politehnica” București

Facultatea de Automatică și Calculatoare,
Catedra de Calculatoare

LUCRARE DE DISERTAȚIE

Analiza si Aplicatie Critica asupra problemelor de Mobilitate in Retelele Wireless

Conducător Științific:
SI. Dr. Ing. Elena Apostol

Autor:
Mititelu Ștefan-Cristian

București, 2016

Contents

List of Figures

List of Tables

Abstract

Mobile devices are becoming "a must" in our day to day society. With this current growing necessity, new challenges appear in order to satisfy end user needs. One of the biggest challenges in the mobile environment is the handover from one network to another. There is an ongoing effort concentrated on this specific issue. There are many solution developed and improved over time at different OSI Layers.

This paper investigates one handover solution at each OSI Layer and focuses on Application Layer, specifically on SIP protocol. In this context, we simulate several such solutions. We analyze and compare the results for those solutions.

In the end, we propose a new, centralized, handover trigger approach based on geolocation. As more and more devices have GPS capabilities, the new handover approach could switch the responsibility from the device, to the Internet Service Providers. The latter might take better handover decisions given the fact that they have the overview of their network. Our solution is implemented on top of Kamailio open source SIP server.

Chapter 1

Introduction

1.1 Context - The Mobile Environment

Due to the need of flexibility, the usage of wireless technologies are becoming more and more widespread nowadays. Subjects are not confined to a fixed position anymore. They can move at their own will. Thus, new issues arise, that never happened before in a wired environment. For example, a wireless moving device will not receive data as soon as it switches between wireless networks. However, mobility does not refer to a person's movement only. Switching to a better signal is also correlated with mobility. Take for example a stationary smart phone which might switch to 3G because of the repeated fluctuations in the Wifi signal.

There are several types of mobility, sorted either by category of user or by availability of services. The former concentrates on the subjects while the latter focuses on service management.

Category of user mobility are described next:

- **Terminal Mobility** refers to a single device capable of moving between different networks. Mobile IP[11] and Mobile Stream Control Transmission Protocol[13] are examples of network and transport layer terminal mobility mechanisms.
- **Personal Mobility** refers to the capability of a person to access the network without depending on the gateway or the device it uses. Therefore, a person could be reached at the company PC by day and personal phone by night.
- **Application Mobility** refers to the capability of relocating applications while moving to a new terminal, such that a person can access that application regardless the device.
- **Session Mobility** refers to maintaining the current sessions while either subjects, applications or terminals are moving from one network to another. For example, the conference session is maintained when switching from voice-video enabled to voice-only enabled terminals.
- **Role Mobility** refers to switching between two or more roles of a person, particularly switching between services that come with each role.

Availability of services mobility are presented next:

- **Continuous mobility** refers to maintaining services while moving.
- **Discrete mobility** refers to maintaining services only on limited areas .

The Mobile Environment consists of different types of networks, varying from traditional telephony networks to Wifi, WIMAX, 3G, UMTS, LTE networks. One of the most promoted wireless technology is Wifi. Wifi networks could be grouped in infrastructure and ad-hoc networks, as depicted in Figure 1.1.1. In infrastructure networks the message flow is managed by a central entity. In ad-hoc networks no central entity is needed. In this thesis, we handle mainly Wifi networks.

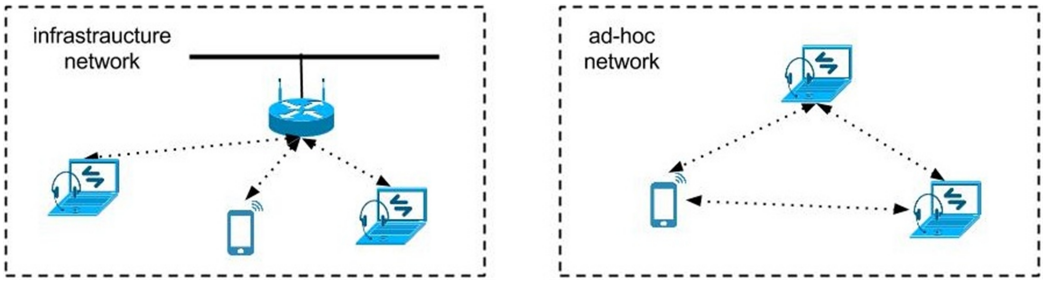

Figure 1.1.1: Infrastructure and ad-hoc networks

1.2 Motivation

The nowadays technologies evolve to offer more and more user flexibility. The end user is not bound anymore to a personal desktop computer and prefers some of the other options like laptops, smart phones or PDA. Those devices offer more freedom to the user when speaking about accessing them. However, a lot of new technical issues appear because of this mobility freedom, issues like network access while traveling, identifying and locating the moving users and delivering data to them.

Many applications need stable links in order to transfer RTP packets like voice or video. It is desirable to offer the end users a stable platform with reliable services, even when they are moving. Thus, new challenges arise when trying to decide when and how to switch to a new wireless cell in order to have the best signal at a given time.

Due to the growing mobile telephony usage and growing popularity of VoIP technology, we decide to focus on Application Layer, particularly on SIP[14] protocol because it is widely used in the VoIP systems. Moreover, we choose to investigate the Application Layer handover because, opposed to lower layers handover, the application is able to adapt to the new environment conditions. For example, when moving from a high throughput wireless connection to a lower one, the application might decide to change the encoding or resize the rendered video screen in order to keep providing reasonable services.

1.3 Objectives

We aim to find the fastest approach for lowering the data flow interruption, such that it will be unperceivable to the end user. This thesis investigates some of the SIP handover solutions and improve them with a new handover trigger solution.

In the first part of this thesis we simulate and compare some current SIP handover solutions. We describe each scheme and simulate it using SIPp, a software SIP simulation tool. In the end we provide the simulation result and a small conclusion based.

In the second part of this thesis we propose an alternative way of triggering SIP handover, based on a centralized approach, using a SIP server. We describe how this is done and measure the time delay introduced by this approach. In the end, we decide if this is a feasible approach or not.

1.4 Summary

This thesis is divided into two parts. The first part discusses and evaluates existing SIP handover schemas. The second part proposes a new centralized handover trigger approach, based on Kamailio**Kamailio**[5] open source SIP router.

Chapter 2 groups and describes mobility types. Also distinguishes between centralized, distributed and replicated mobile network architectures and summarizes their specific features.

Chapter 3 details one handover scheme for each of the OSI Layer 1 to Layer 5. It starts from ARUBA[16] single channel deployment, continues with an alternative approach to IAPP[1][10] protocol. Following, it discusses Mobile IP[12] and Multipath TCP[6] ideas. IT ends with describing the Session Layer Mobility[7] framework.

Chapter 4 focuses exclusively on Layer 7 OSI, SIP based handover approaches. It starts from considering reINVITE solution and enhances it with proactive and proactive extension ideas; also it discusses REFER and JOIN based handover possibility. Afterwards, it tests, gathers results and compares each of the analyzed schema using SIPp traffic generator, in a basic peer to peer test setup.

Chapter 5 states the proposed handover trigger idea based on Kamailio open source SIP router. It presents the Kamailio geolocation setup and message flow. It shows how Kamailio server triggers handover when the mobile node reaches out the AP area. This proves that one can implement geolocation handover with Kamailio, as long as Kamailio has an overview over the network i.e. AP distribution.

Finally, chapter 6 states some conclusions related to the geolocation feature and advances further ideas like SIP user access based on geolocation or extending soft phones to send geolocation to Kamailio.

Chapter 2

Related Work

This chapter investigates some of the current handover solutions starting from the Layer 1 OSI to Layer 5 OSI. The chapter starts with a Layer 1 solution, based on ARUBA network, followed by a Layer 2 solution based on Distributed Neighbor Discovery Protocol. The chapter continues with Mobile IP and Multipath TCP as a Layer 3 and Layer 4 approaches. The chapter ends with Session Layer Mobility, a Layer 5 approaches.

2.1 Layer 1 - ARUBA Single Channel Model

ARUBA proposes paper[16] in which examines both the good and bad parts of having single channel WLANs deployment. The paper starts from basic, adaptive model and advances to single channel model and the co-channel interference issues. The paper discusses some architectural models, discussing both their advantages and disadvantages. The goal is to build enterprise WLANs, considering capacity, scalability, robustness, flexibility and performance. Some performance analysis are presented, in order to clarify marketing statements.

The idea of the single channel model is to have Access Points working on the same frequency, distributed over the area which needs connectivity, as shown in Figure 2.1.1. In the most implementations, the end user does not know to which AP is actually connected, so the client does not engage in handover decision. It is the network's role to decide to which AP to deliver data, such that the end user does not notice anything while moving between APs.

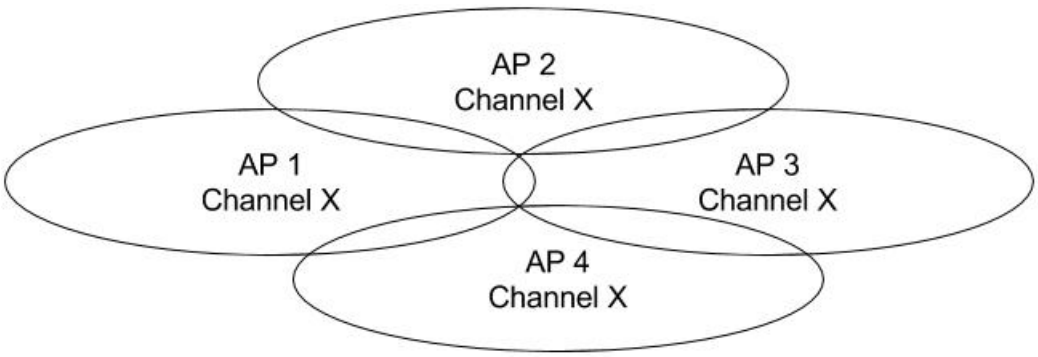

Figure 2.1.1: ARUBA Single Channel WLANs

The first advantage of the single channel model is that the client is not involved anymore in

handover decision. However, switching the handover decisions from client to network is not a simple task and new challenges arise. The basic handover flow is pictured in Figure 2.1.2. To accomplish this type of handover, the APs are modified to send the same BSSID in the beacon frames. Additionally the current selected AP must trace and increment sequence numbers smoothly, for the client to receive a continuous data flow.

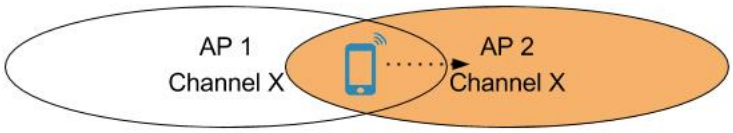

Figure 2.1.2: Handover between APs

The second advantage is that there is no need for channel planning because only one channel is used. However, other problems appear related to co-channel interference, not present in the adaptive model(which used orthogonal channels for adjacent APs). The coverage and interference areas of an AP can not be fully predicted(Figure 2.1.3). The transmissions from one AP may interfere with the transmissions from other AP, on the same channel, leading to packet errors.

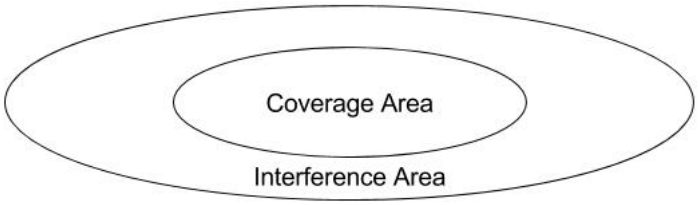

Figure 2.1.3: AP Coverage and Interference areas

To solve co-channel interference one can use spatial separation(bigger distances between APs) or temporal separation(control the adjacent APs so they don't transmit at the same time). The single channel model uses the latter idea for solving the APs interference, improved with QoS.

2.2 Layer 2 - Distributed Neighbor Discovery Protocol (DNDP)

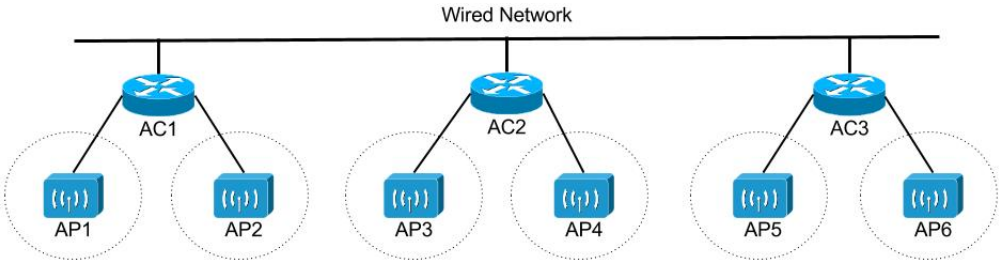

Figure 2.2.1: DNDP Architecture

The protocol comes as an alternative to IAPP[1][10] or CARD[9] protocols and uses the distributed approach. Thus, it needs some intelligence on the network equipments in order to take the handover decisions. The proposed architecture for this protocol is shown in Figure 2.2.1. Many Access Points(APs) can be connected to the same Access Controller(AC).

DNDP uses two tables for taking decisions:

- **Candidate Access Controllers(CAC)** table which is build in a distributed fashion and contains "own AP<=>neighbor AC" mapped pairs. The purpose of this table is to help in context transfer.
- **Neighbor Access Points(NAP)** table contains "neighbor AP<=>neighbor AC<=>timer" mapped pairs. The purpose of this table is to help in deciding if this AC is still candidate for the AP.

If the Mobile Node(MN) moves between the APs of the same AC, there is no need for performing a context transfer. The handover happens when moving between the APs of different ACs.

Figure 2.2.2 shows how DNPD works. Firstly, MN sends register upon moving, which is detected by the new AC. Following, the new AC notifies the old AC. Next, the old AC updates CAC table and notifies back the new AC. Finally, the new AC updates NAP table. Upon timer expiration, the new AC is no longer candidate for the AP and the entry is removed from NAP table.

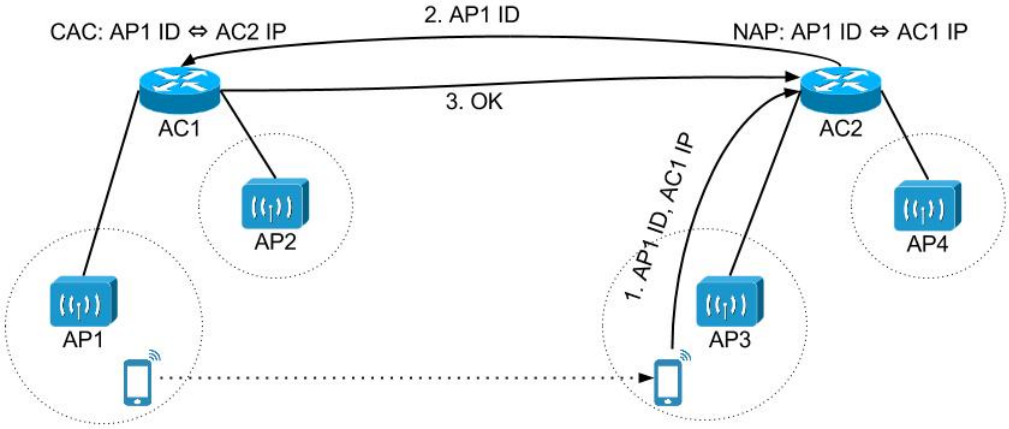

Figure 2.2.2: DNDP Logic

2.3 Layer 3 - Mobile IP (MIP)

Mobile IP[12] is a Layer 3 protocol designed by IETF and provides IP handover mechanism for devices which move from one network to another. It is best suited for active users who might move a lot thus switching between networks very often. Mobile IP was designed to support seamless and continuous Internet connectivity. Mostly useful in wireless networks, it is also found in wired networks. For a better understanding let consider the wireless networks because the most roaming devices are wireless.

2.3.1 MIP entities

- **Mobile Node(MN)** is the device which moves and switches between networks. MN has 2 network interface cards, one used for the home network and another one for the foreign network.
- **Correspondent Node(CN)** the device to which MN communicates. CN may move but for simplicity consider it stationary.

- **Home Agent(HA)** is the default gateway in the home network of MN. HA must implement all the basics of a router, including tunneling; may provide a DHCP server.
- **Foreign Agent(FA)** is the default gateway in the foreign network of MN. HA must implement all the basics of a router, including tunneling; may provide a DHCP server.

2.3.2 MIP functionality

- Suppose MN is placed in the home network. It connects to the home network using the home interface and it gets a home IP address either by DHCP or static IP. This home address is never deleted or changed, even when MN is in the foreign network. The message generated by MN is sent to HA which forwards it to CN. The response travels back to HA which forwards it back to MN. This is the situation of classic routing and there is no need of the second interface(foreign interface).
- As MN moves and arrives in the foreign network an IP is assigned to the foreign interface, either by DHCP or by FA. This address is called Care-of-Address(CoA), it changes within different foreign networks.
- MN updates its location by sending a Registration Request to HA which is now aware of the Care-of-Address. The message generated by MN still has home IP address as its source address and CN address as its destination address. Because routing occurs based on destination address, the message is send through foreign interface to the default router i.e. FA which forwards it to CN.
- CN does not know that MN is roaming and is not aware of the MN's foreign location. MN is a static node to him, thus the response is sent to HA. HA tunnels the message by using layer 3 encapsulation to add the CoA over the current message destination IP and then forwards it to FA. The latter performs routing table lookup based on the tunnel header, drops the tunnel header of the message and forwards it back to MN. The below Figure 2.3.1 depicts how MIP works

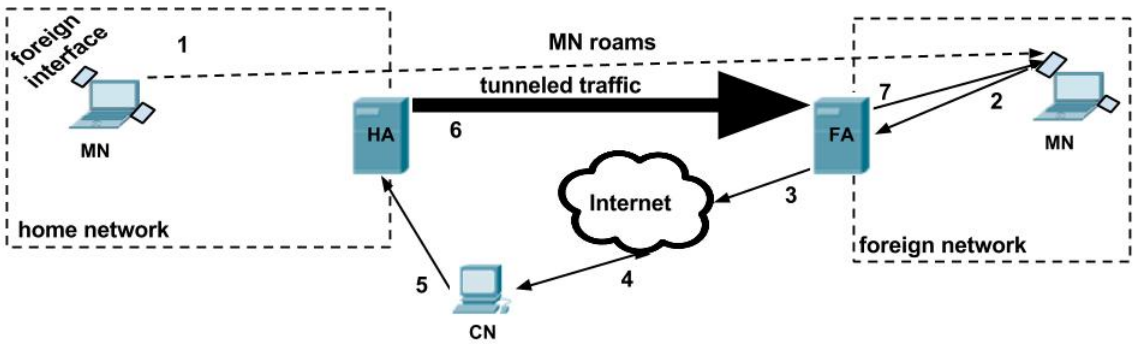

Figure 2.3.1: MIP Functionality

Because of the detoured traffic from CN to MN through HA, triangular routing problem arises as Figure 2.3.1 shows. To perform tunneling about 20 bytes must be added to each packet which decreases the size of the useful payload. In addition, the detour brings in an unwanted latency, not accepted in real time traffic. To overcome this problems, the Route Optimization(RO) was proposed. Basically, the HA informs the CN of the CoA of the MN. Now, the CN can transmit directly to MN, thus reducing latency problems that were present even if the CN and MN were close to one-another. However, this solution has also some disadvantages because CN must know to receive and store CoA and to tunnel messages; a delay still exists until HA informs CN of the MN's CoA and CN still relies on the old FA until the receives the updated CoA.

2.4 Layer 4 - Multipath TCP (MTCP)

MPTCP[6] is a relatively new transport layer protocol also designed by IETF to support multi homing. In addition to SCTP, it supports data transfer on multiple paths at the same time. Therefore, the total throughput is the sum of the throughput on each path. It is compatible with standard TCP and it can be used to perform seamless session handover.

After the connection to the new AP and the attainment of a IP address, MN sends a SYN Join message to initiate the new path and waits for the SYN Join acknowledge. Once acknowledgement received, data for the same session can be sent simultaneously on both paths, as pictured in Figure 2.4.1.

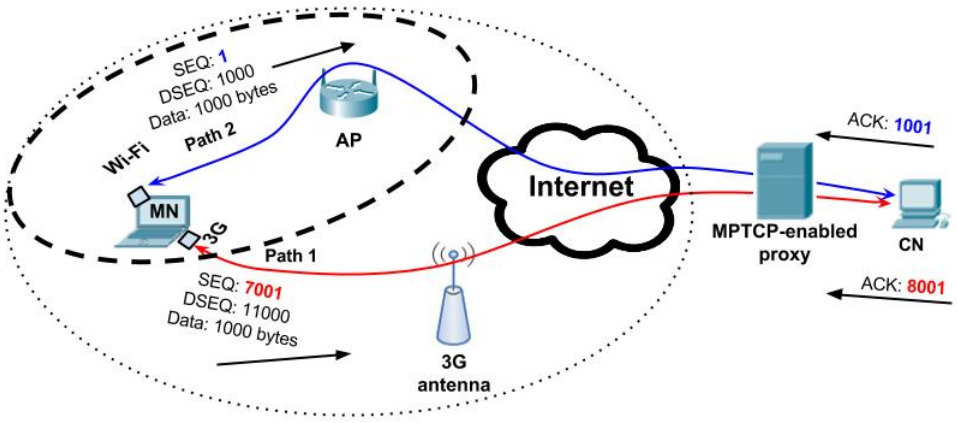

Figure 2.4.1: MPTCP Functionality

2.5 Layer 5 - Session Layer Mobility (SLM)

SLM[7] is another mobility approach applied over the Transport layer namely at the Layer 5 OSI. SLM is actually a framework whose purpose is to switch session streams between two end devices, thus maintaining current TCP sessions while moving.

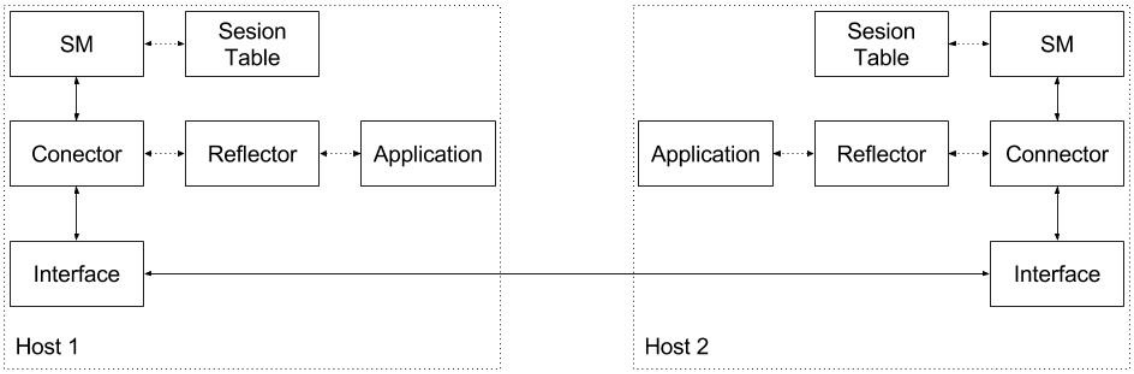

Figure 2.5.1: SLM Functionality

As depicted in Figure 2.5.1 a Session Manager (SM) administers all the connections of a host. It starts or ends sessions, by creating or destroying connectors, and also maintains session state

in a session table. The Connector sends data down the socket and also triggers the peer's SM to create a local connector for establishing a socket. The Reflector interfaces data between the application and the Connector in such a way that the application is not aware of the changes that might occur while moving.

To distinguish between sessions, session IDs or group, session IDs are used. When switching to another device, the local SM must signal to the target SM, requesting the handover for an application session ID. The target SM then creates a new Connector and signals the peer SM with the above received ID. In the end, the peer SM reconnects the session to the target SM. Hence, only path between the two Connector has to change.

They tested the SLM architecture on 2Mbit/s MCS WLAN networks using Java implementation. A "standard" PC running Linux was playing an MP3 from a server at a rate of 0.5Mbit/s. The measured time needed for TCP to re-establish the connection was around 25ms which. This is the time needed to switch the TCP sessions and is expected to increase with the increasing distance between the mobile node and server.

Chapter 3

Case Study and Analysis of Current SIP Handover Solutions

This chapter evaluates and compares some of the most important Session Initiation Protocol handover ideas. We have chosen SIP protocol due to its versatility and robustness as a signaling protocol and its increasing usage in VoIP technologies.

The chapter starts with the analysis of each SIP handover idea, pointing to both advantages and disadvantages. The chapter continues with the case study and testing of each solution, using SIPp[8] traffic generator, in a basic case study setup. The chapter ends with a SIP handover time comparison of the handover for each investigated idea.

In order to perform seamless application handover, the Mobile Node has to be equipped with at least two communication interfaces. This is not far-fetched taking in consideration that nowadays smart phones have both an Wifi and 3G interface. All test scenarios discussed in this thesis suppose a Mobile Node with two Wifi interfaces. Additionally, UAS triggers handover in all test cases. The measured handover time starts with the handover trigger and ends when media flows on the new interface.

3.1 Analysis of SIP Handover Scenarios

3.1.1 Analysis of reINVITE Scenario

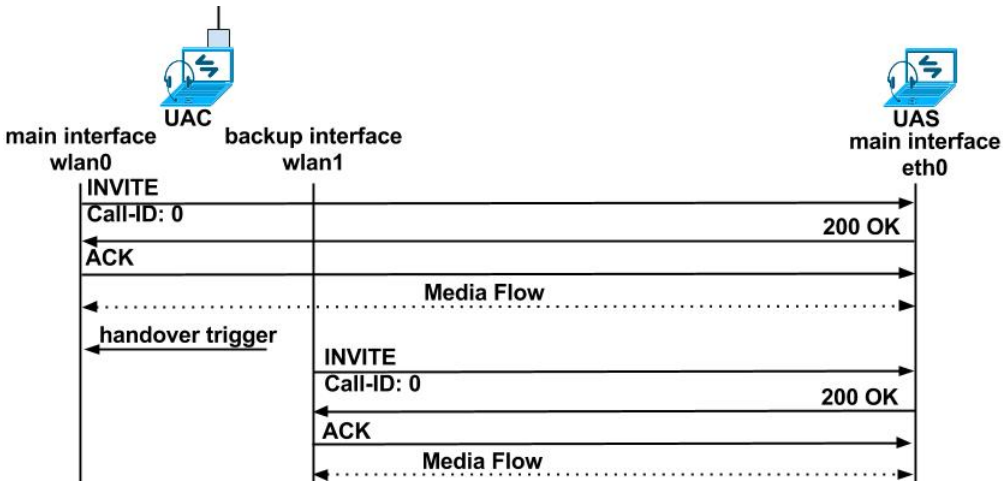

Figure 3.1.1: reINVITE Handover Scenario

This section presents the basic handover scheme using SIP protocol, based on the SIP INVITE message. Basically, the idea is to send a SIP INVITE request with new "Contact:" pointing to lowland address when the signal on wlan0 is decreasing. It is a simple, yet fast SIP handover technique, with built-in SIP protocol support.

Figure 3.1.1 pictures the basic SIP handover message flow. When the UAC moves and wlan0 received signal strength decreases, the handover is triggered. This lead to sending reINVITE to move the existing session on wlan1.

3.1.2 Analisys of reINVITE Proactive Scenario

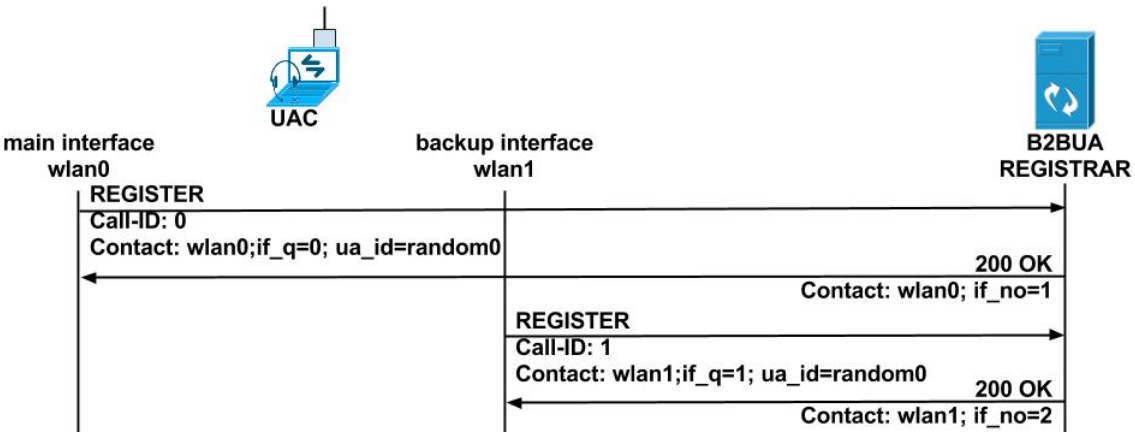

Figure 3.1.2: Proactive Interface Registration

This section adds an improvement to the previous 3.1.1 handover scheme, by adding a proactive feature[3]. While stating the idea, the authors focus especially on reducing the delay and packet

loss during handover. Also they keep in mind the possible event of losing sessions due to sudden link drop and also the ease of deployment.

Upon network detection, all UAC interfaces try to connect and get network information like IP address, and default gateway. UAC REGISTERs its interfaces with different priorities sent in “if_q” parameter, as shown in Figure 3.1.2. One main interface is maintained, based on the "if_q" priority, while all the others are considered backup interfaces. All the REGISTERs sent for different interfaces, from the same UAC to the SIP server, have to contain the same “ua_id”.

Subsequently, using SIP signaling on all UAC interfaces, RTP session is established through the main interface, and placed on hold for the backup interfaces. All the INVITEs are sent with the same SIP session identification parameters. When handover triggers, a new INVITE is sent on the backup interfaces which informs the B2BUA to replicate packets on both interfaces(main and backup). The B2BUA is essential in order to provide zero packet loss during handover. UAC must filter the received duplicated packets. After BYE is sent on the old main interface and handover is completed, the priority of the interfaces adjust accordingly and B2BUA sends duplicated packets no more. Figure 3.1.3 pictures this whole process.

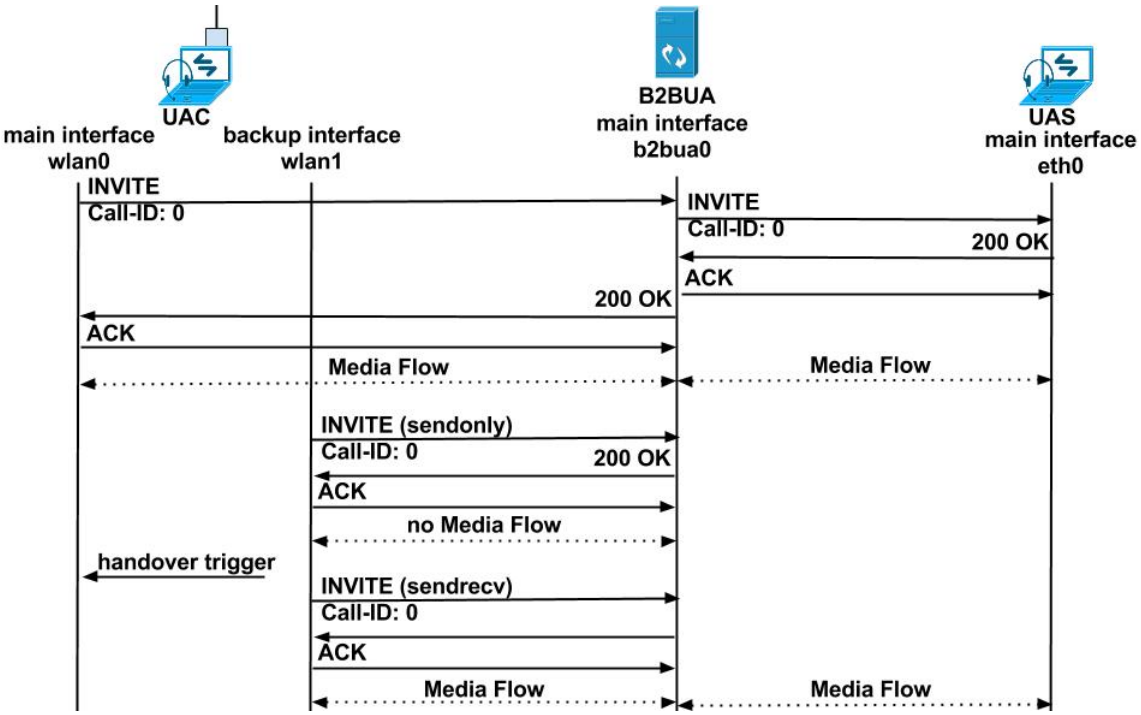

Figure 3.1.3: reINVITE Proactive Handover Scenario

One drawback is that for some networks, like UMTS, signaling might take longer than sending media. When UAS receives the INVITE from the backup interface it sends 200 OK back to the UAC and is able send media immediately. UAC needs to process the 200 OK before actually establishing media stream. Thus, the first media packets could be lost. On the other hand, this is only relevant when the link on the main interface breaks suddenly.

Other drawback of this approach is the B2BUA single point of failure for both signaling and media paths. To solve this, the authors proposed extension ?? to the proactive approach.

3.1.3 Analisys of reINVITE Proactive Scenario with Extension

This section presents an extension[2] to the previous proactive approach by using a new SIP "Handover:" header field. The new header identifies a dialog and looks like the following: "Handover: call-id; to-tag=X; from-tag=Y". The idea is that media can be established directly between the UAS and UAC as long as the former understands "Handover:" values; thus, the media bypasses B2BUA. UAS responds with “420 Bad Extension” if it does not support proactive handover. If that is the case, media will be still routed through B2BUA.

Figure 3.1.4. As you can see, when handover is triggered, UAC sends a new dialogue INVITE which contains the initial diloag identification parameters in the "Handover:" header. If UAS supports this header, it will establish a second media session with UAC's second interface. UAS will duplicate RTP packets until the first media session is destroyed. Thus UAS still need a duplicated packet filtering.

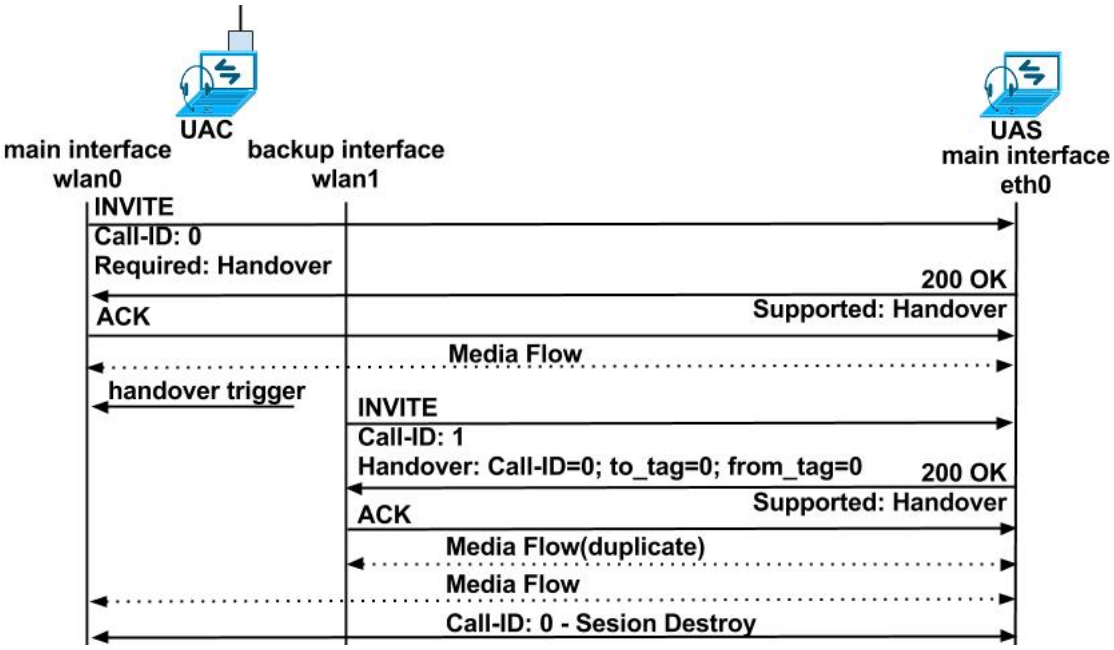

Figure 3.1.4: reINVITE Proactive Extension Handover Scenario

The advantage of this extension is that allows media to flow directly between endpoints, without a B2BUA need.

The drawback of this extension is that not all endpoints supports the newly proposed SIP header, thus falling back to the previous 3.1.2 drawbacks.

3.1.4 Analisys of REFER Scenario

Another SIP handover scenario described in [15] is based on SIP REFER method. This is another example of seamless soft handover scenario because the end user does not notice the handover occurrence.

As shown in Figure 3.1.5, after the initial call is set up on the UAC's first interface, media is flowing between UAC and UAS. When the link is degrading and the handover is triggered, UAC sends a REFER method through the first interface with a "Refer-to:" header pointing to its second interface. Next, UAS tries to establish a new session through the second interface, given previous "Refer-to:" information. After it finishes setting up a new session with UAC's second interface, it destroys the former session.

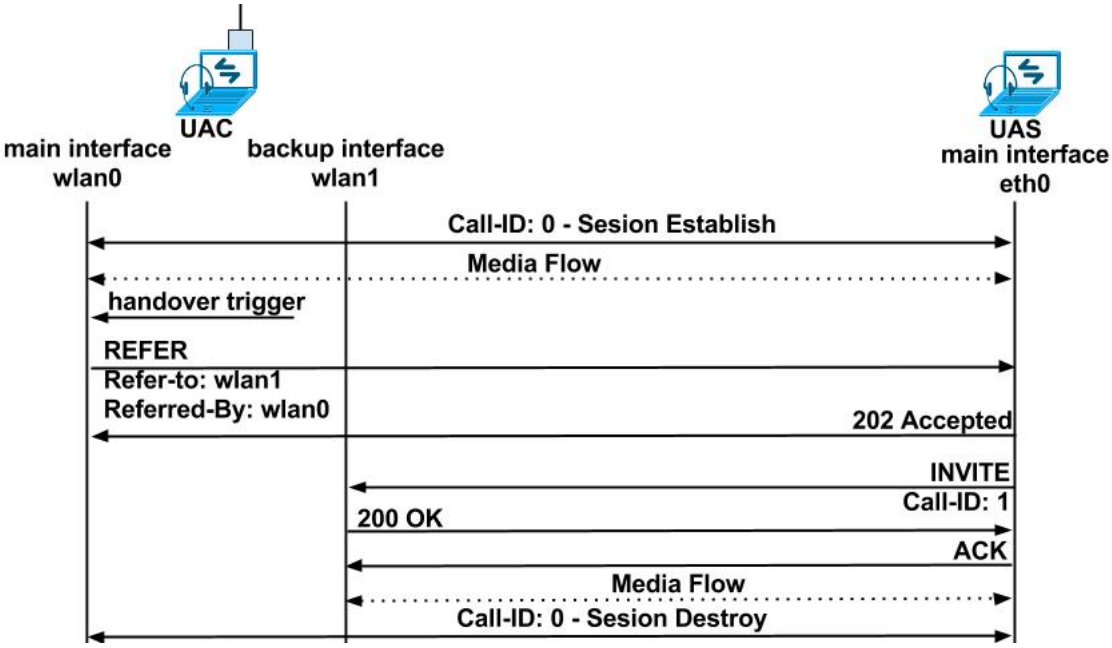

Figure 3.1.5: REFER Handover Scenario

The advantage of this approach is that REFER method (RFC 3515) is already included in the SIP stack so the endpoints need no special upgrade to understand the flow. Another asset is that UAC can send REFER method directly to its peer UAS, so there is no bottleneck in the SIP flow.

One of the drawback of this approach is that a new session has to be established using additional REFER transactions; the complete RFC scenario also includes some NOTIFY transactions. This leads to a higher number of exchanged SIP messages. Another drawback is that UAC needs to be equipped with RTP packet filter in order to remove duplicates that come until the initial session is destroyed.

3.2 Case study of SIP Handover Scenarios

3.2.1 Tools

This section describes the tools used for testing the above examined scenarios. For simulations, we use SIP packet generators rather than SIP soft phones. Thus, we can target the time needed for handing over the SIP session without counting soft phones delay. Moreover, we can script and replay the desired test case.

Among the SIP traffic generators tried are SIPp, sipsak and SIPUnit. We choose SIPp for testing because of its maturity, versatility, and ease of usage. In addition, we use tcpdump and editpcap to capture the SIP messages in a file and eliminate duplicates in order to have a cleaner flow. The main reason of choosing tcpdump is because it is a CLI packet filter and has a lot of documentation and examples. Editpcap comes with wireshark suite and has also CLI syntax and has some straightforward parameters.

In the end, we use callflow to generate images with the actual SIP messages, for a better understanding of the evaluated schemes. Also, it helped in SIP message debugging. A brief description of each used tool will follow:

- **SIPp[8]** is an Open Source SIP packet generator, maintained by HP. One can configure his own test cases in an .xml file. Thus, very simple to very complex call flows can be easily specified to cope with the specific experiment needs. It dynamically displays statistics about running tests like call rate, round trip delay, and message statistics. Also, it offers a simple curses interface for dynamically increasing or decreasing call rate or specifying number of calls per second etc. Compared to other tools like sipsak or SIPTools, SIPp can better simulate a real life use cases using third party call control (3pcc) feature and being able to send RTP traffic, through pcap re-play.

  Of course, SIPp has its limitation of how many simultaneous messages can send, but this depends also on the hardware used. To simulate a stress test we used SIPp with 100 calls per second. We have mostly used SIPp in 3pcc mode with pcap re-play to simulate the below case studies. All the scenarios we have tried can be found in [4].

- **Tcpdump** is the well known CLI packet filter. Even if it has a lot of filtering options, tutorials and examples can be easily found. We have used tcpdump to simply capture all the traffic and to save it in a .pcap file. The resulting file is used by edipcap and callflow for further processing.

- **Editpcap** is a CLI wireshark suite tool use for editing the resulted .pcap file. It is simple to use because it has a straightforward syntax. It has a lot of option and allows for very complex .pcap file editing. We have used it in order to eliminate the duplicated SIP frames generated sometimes by SIPp when the expected timers were reached.

- **Callflow** is a CLI tool used to generate a picture from a given traffic trace. It makes possible to replace the IPs of the endpoints with their names. It is also possible to select the packets that will be plotted based on a filter, which is very useful when plotting RTP media. In order to generate the .png file, it uses a image processing software, namely inkscape.

  We have used callflow for making SIP and RTP diagrams given a .pcap trace. The .pcap file was generated using tcpdump tool. Thus, the SIP handover techniques can be easily understood.

3.2.2 Setup

This section depicts in Figure 3.2.1 the setup we use for testing the previous analyzed scenarios. The SIPp tests are run on an intel i7 dual-core, 2.5 Ghz UAC running SIPp and an UAS running SIPp on a Local Machine, Local Area Network and Internet. The UAC is equipped with two wifi interfaces, namely an integrated Broadcom card(Tx Power: 10dbm) acting as wlan0 and an external Realtek card(Tx Power: 25 dbm) acting as wlan1. The UAS is equipped with a wired 100Mb/s Realtek interface card acting as eth0.

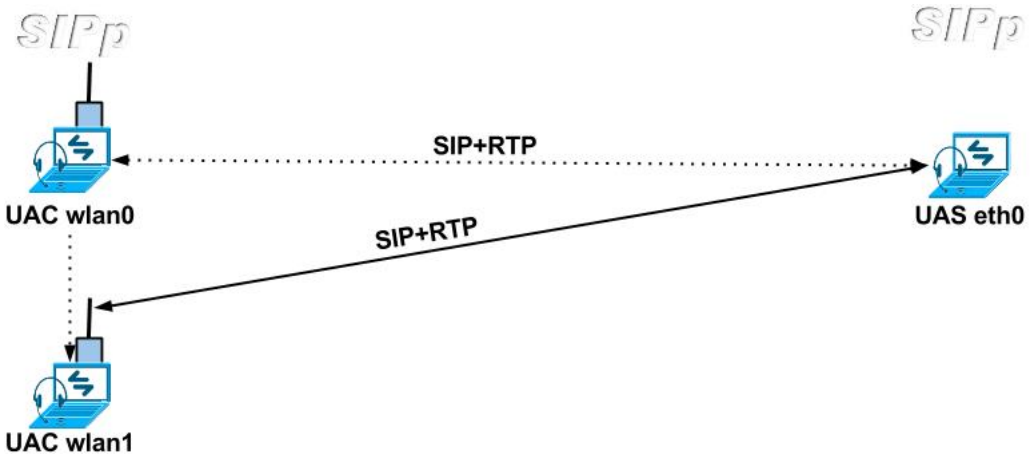

Figure 3.2.1: Two-Interface Setup

3.2.3 Simulations

This section pictures the callflow of one call from each of the previously investigated SIP handover schemes.In order to simulate the above considered scenarios, we use third party call control(3PCC) SIPp feature. The feature allows UAS to have separate, yet inter-dependent, SIP logic for communicating with UAC wlan0 and wlan1. This is very helpful when synchronizing the UAS SIP messages in accordance to UAC wlan0/wlan1 interfaces. Have a look on Figure 3.2.2 for a better understanding of the SIPp 3PCC aspect.

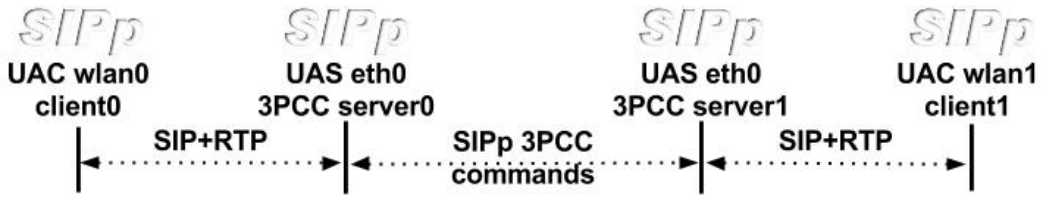

Figure 3.2.2: Two-Interface SIPp Flow

The first simulation, presented in Figure 3.2.3, is based on the reINVITE approach. As seen below, the session is established via UAC wlan0 interface. Upon handover trigger, the **same** session gets transfered on the wlan1 interface. The handover time is the time needed for this transfer. Ultimately, the session is terminated on wlan1.

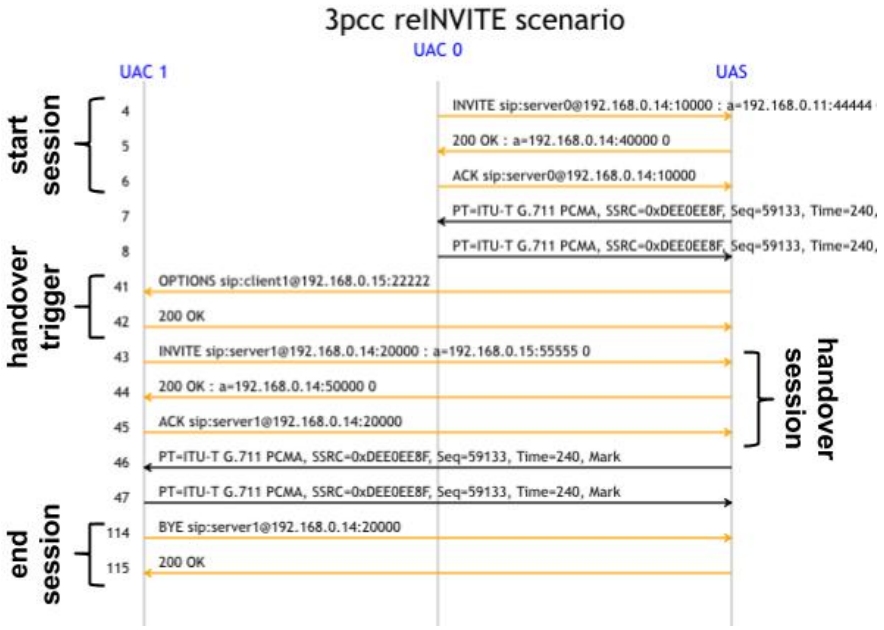

Figure 3.2.3: reINVITE Scenario Flow

The second simulation, presented in Figure 3.2.4, is based on a proactive improvement of the previous reINVITE approach. As pictured below, the session is established via UAC wlan0 interface. Afterwards, the **same** session is established, pro-actively, on interface wlan1, without allowing media through it.

Upon handover trigger, the pro-actively established session gets activated on the wlan1 interface, thus allowing media through it. The handover time is the time needed for this proactive session activation. The biggest advantage with this scheme is that the proactive switch requires only 1 message to make the transfer and allow RTP on wlan1. Finally, the session ends on wlan1.

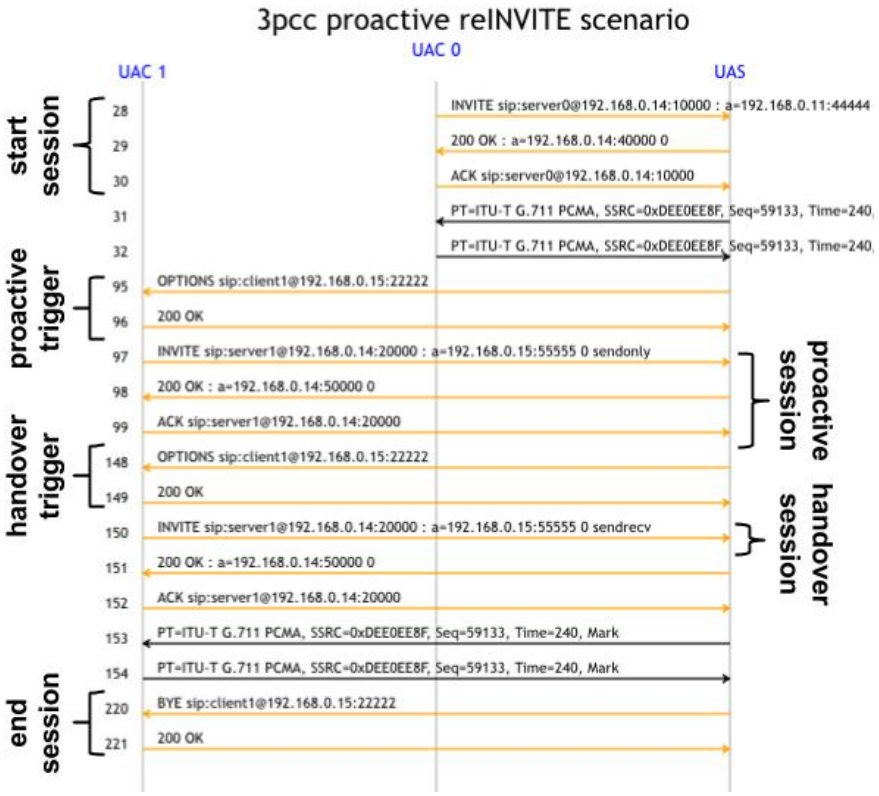

Figure 3.2.4: reINVITE Proactive Scenario Flow

The third simulation, presented in Figure 3.2.5, is based on the reINVITE approach. The difference is that upon handover, a new session is created on wlan1. The improvement of this method is not about of handover time but about detecting if the endpoints implement and support proactive handover by adding the "Handover:" header extension. The handover time is the time needed for creating the new session on wlan1. The old session on wlan0 is destroyed. Ultimately, the session ends on wlan1.

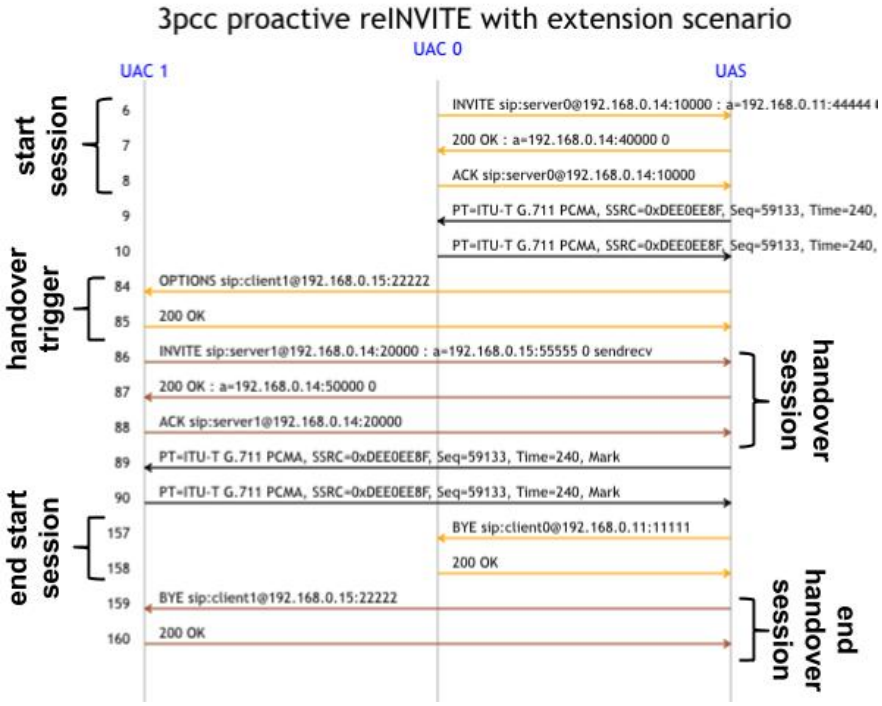

Figure 3.2.5: reINVITE proactive Extension Scenario Flow

The fourth simulation, presented in Figure 3.2.6, is based on the REFER approach. The initial session is established via UAC wlan0 interface. Upon handover trigger, a REFER message is sent to the UAS, referring wlan1. Next UAS establishes a new session with wlan1 and destroys the old session with wlan0. The handover time is the time needed for establishing this new session with wlan1. The session on wlan0 is destroyed and the flow continues on wlan1. In the end, the session is terminated on wlan1.

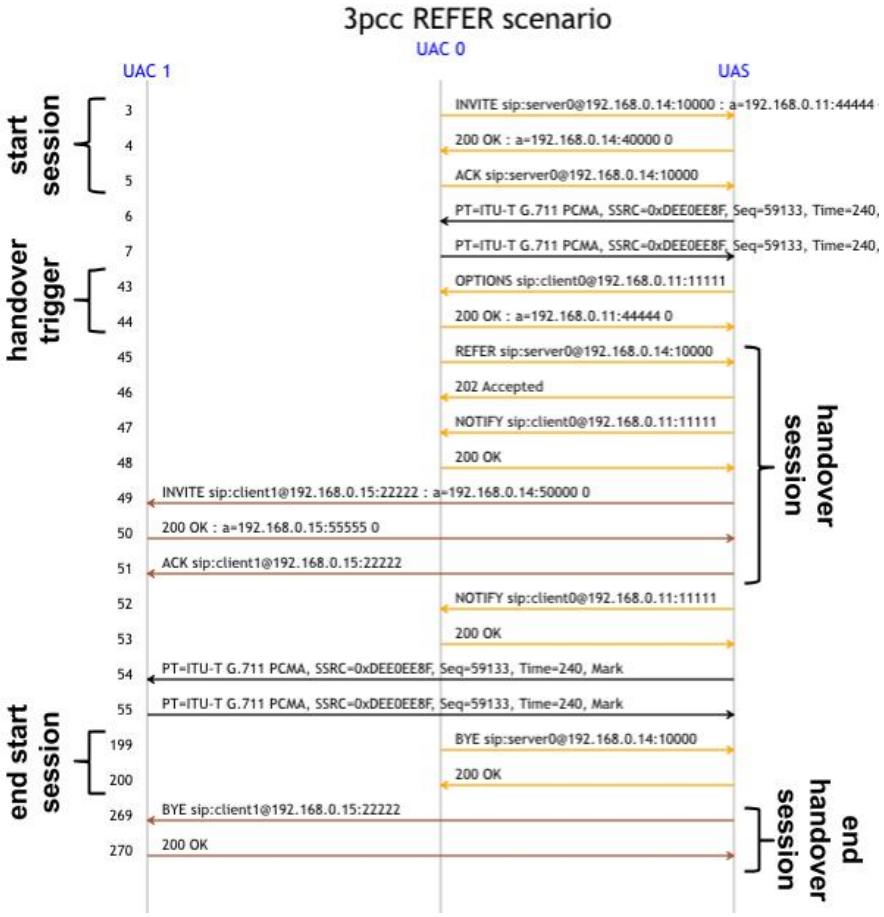

Figure 3.2.6: REFER Scenario Flow

3.2.4 Results

This section compares and discusses SIP handover duration for all the previous scenarios. All the measurements focus on the SIP handover time, namely the time needed until media can flow on the second interface. Thus, the measurements are not influenced by handover detection algorithms, by the network latency or by IP address and IP route setting. The measured time basically tells how much time the application needs for re-establishing the sessions, after lower level handover have happened.

The handover time in the below table is calculated as an arithmetic mean of 1000 calls, 100 calls per second call rate. The unit of measurement is millisecond. The results are present in the below Table 3.1. There were 12 hops between UAC and UAS when we tested over Internet.

Table 3.1: Case Study Simulation Results

Scenario	Local VM (ms)	Local Area Network (ms)	Internet (ms)
reINVITE	0.42	2.1	13.7
reINVITE Proactive	0.17	0.64	5.2
reINVITE Proactive EXxtension	0.41	2.2	13.2
REFER	0.98	4.8	31.4

For a better visual comparison, have a look on Figure 3.2.7. As expected from the above analysis, the fastest handover method is the proactive reINVITE followed by the proactive reINVITE extension and simple reINVITE. The slowest method is the one based on REFER method, beacuse more SIP messages need to be exchanged until media can flow on the second interface.

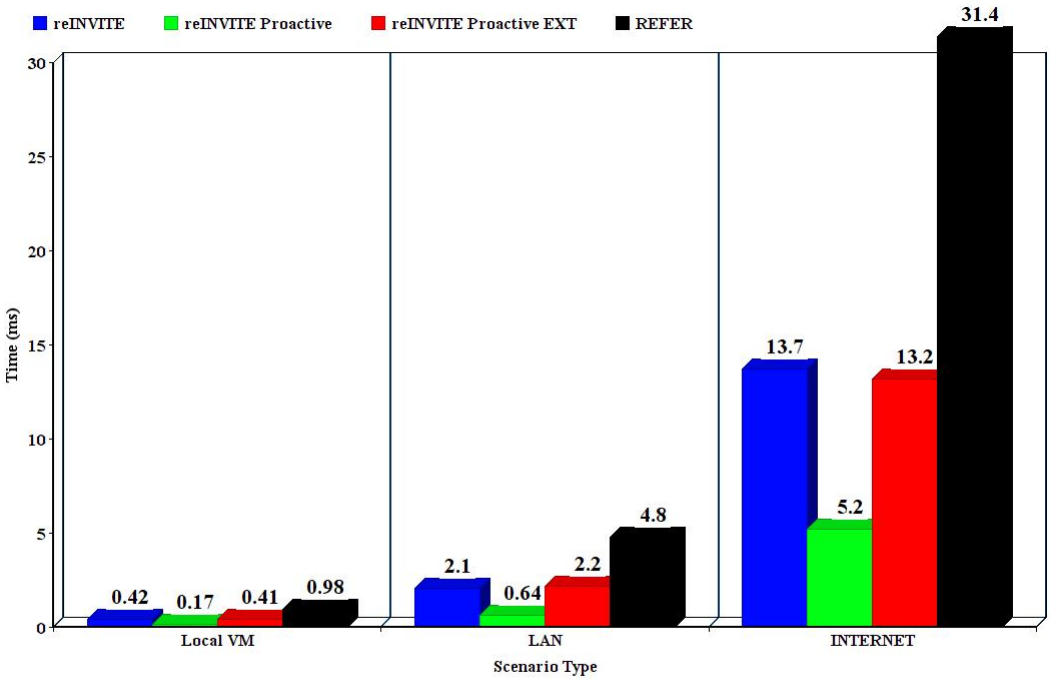

Figure 3.2.7: Two-Interface SIPp Flow

Chapter 4

Proposed SIP Geolocation Handover Trigger

This chapter presents the proposed alternative of SIP handover trigger based on UAC Geolocation. It uses Kamailio as a SIP server to take the handover decision and inform the UAC about this. For this concept, the Mobile Node has to be equipped with at least two communication interfaces. This is a decent assumption, considering that nowadays smart phones have both an Wifi and 3G interface.

It begins by presenting the trigger scenario and the entities present. This includes also a summary about Kamailio server. Next it discusses the idea of geolocation trigger together with how we achieved it. In the end the tests setup is discussed together with the simulation flow of the trigger approach.

SIP geolocation is a relatively recent feature, introduced in RFC 6442. It introduces new headers that allow end user's device to add its current location in the SIP requests and inform the server. Figure 4.0.8 shows the high level architecture used for this handover trigger proposal. The UAC will send the geolocation data in the body of the SIP message. Kamailio extracts the geolocation data, maintains a history of location points and calls a python script to compute the distance from the associated AP. Based on this value, Kamailio decides to trigger the handover for the current dialog or not.

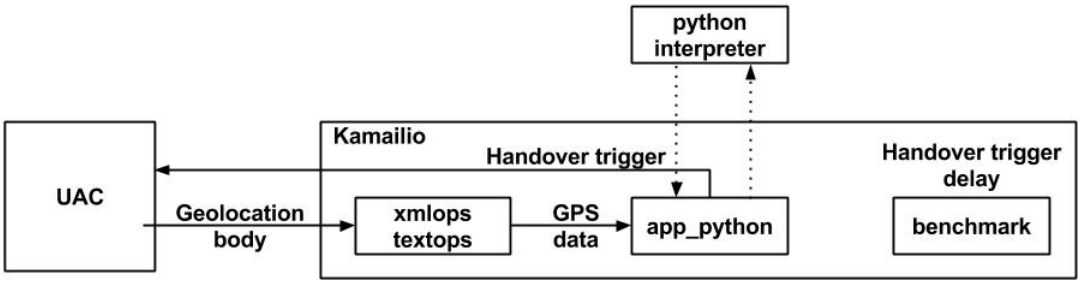

Figure 4.0.8: Handover Trigger Architecture

4.1 Analysis

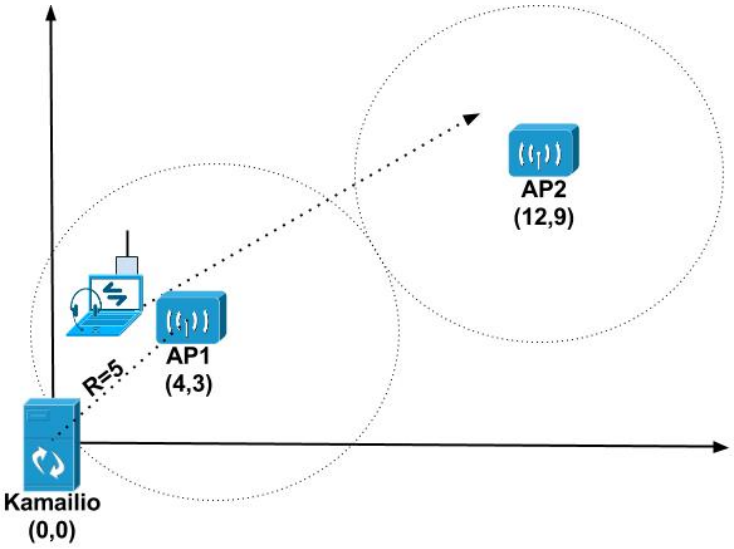

Figure 4.1.1: Geolocation Simulation Idea

This section presents the concept of centralized handover trigger and describes how we implemented it using Kamailio server.

The premises of this approach is that the SIP server has an overview of the network it serves, as pictured in figure Figure 4.1.1. More precisely, it knows the approximate APs ray of action and the position of APs distributed over the network coverage area. In our test setup we have used two APs, whose position definition is maintained in a python script.

In order to implement and test SIP location-based user access, we use **Kamailio**[5], an Open Source SIP Server released under GPL. Kamailio can be used to build large platforms for VoIP and real-time communications: presence, WebRTC, Instant Messaging and other applications. Kamailio is easily extensible because of its modular design. Different modules can be loaded into Kamailio, depending on the functionality one needs. It is written in C programming language and has a great, supporting developers and users community. Kamailio is able to handle up to 200 concurrent calls per second.

As shown in Figure 4.0.8, we are using **textops** and **xmlops** Kamailio modules to extract X-axis and Y-axis geolocation data from the received OPTIONS body. We have contributed with a fix commit to the parsing of Kamailio "multipart/mixed" body parsing in commit 3895cd2e08c26594a3cff866caea2b431eb71d1e. We are using **app_python** Kamailio module and a python script to process the received geolocation input. We are using **benchmark** Kamailio module to eventually measure the per-handover-delay introduced by the handover feature. The Kamailio geolocation code 4.1 is pasted below.

The python script computes the distance between the UAC wlan0 and AP0 using basic euclidean distance. When UAC reaches out of the AP0 area, Kamailio sends him an OPTIONS which tells UAC to connect to AP1 and handover SIP application session on wlan1. To test this new handover trigger scheme, we have used the basic reINVITE SIP handover scheme.

```
if (is_present_hf("Geolocation") &&
    is_present_hf("Geolocation-Routing") &&
    $hdr(Geolocation-Routing)=~"yes" &&
    has_body("multipart/mixed"))
{
        # get body data
        get_body_part("application/pidf+xml", "$var(geobody)");

        # get geolocation data
        $xml(posxml=>doc)=$var(geobody);
        $var(posX)=$xml(posxml=>xpath:/pos:position/pos:x/text());
        $var(posY)=$xml(posxml=>xpath:/pos:position/pos:y/text());
        $var(pos)=$(var(posX){s.trim}) + " " + $(var(posY){s.trim});

        # get geolocation data history
        if (is_method("OPTIONS")) {
                $dlg_var(dlg_pos)=$dlg_var(dlg_pos) + " ";
        }
        $dlg_var(dlg_pos)=$dlg_var(dlg_pos) + $var(pos);

        # call python script with geolocation data history
        python_exec("my_python_function", $dlg_var(dlg_pos));
}
```

Listing 4.1: Kamailio Geolocation script code

4.2 Setup

This section depicts in Figure 4.2.1 the setup we use for testing the previous analyzed scenario. The SIPp tests run on an intel i7 dual-core, 2.5 Ghz UAC running SIPp and an UAS running SIPp on a Local Machine, Local Area Network and Internet. The UAC is equipped with two wifi interfaces, namely an integrated Broadcom card(Tx Power: 10dbm) acting as wlan0 and an external Realtek card(Tx Power: 25 dbm) acting as wlan1. The UAS is equipped with a wired 100Mb/s Realtek interface card acting as eth0.

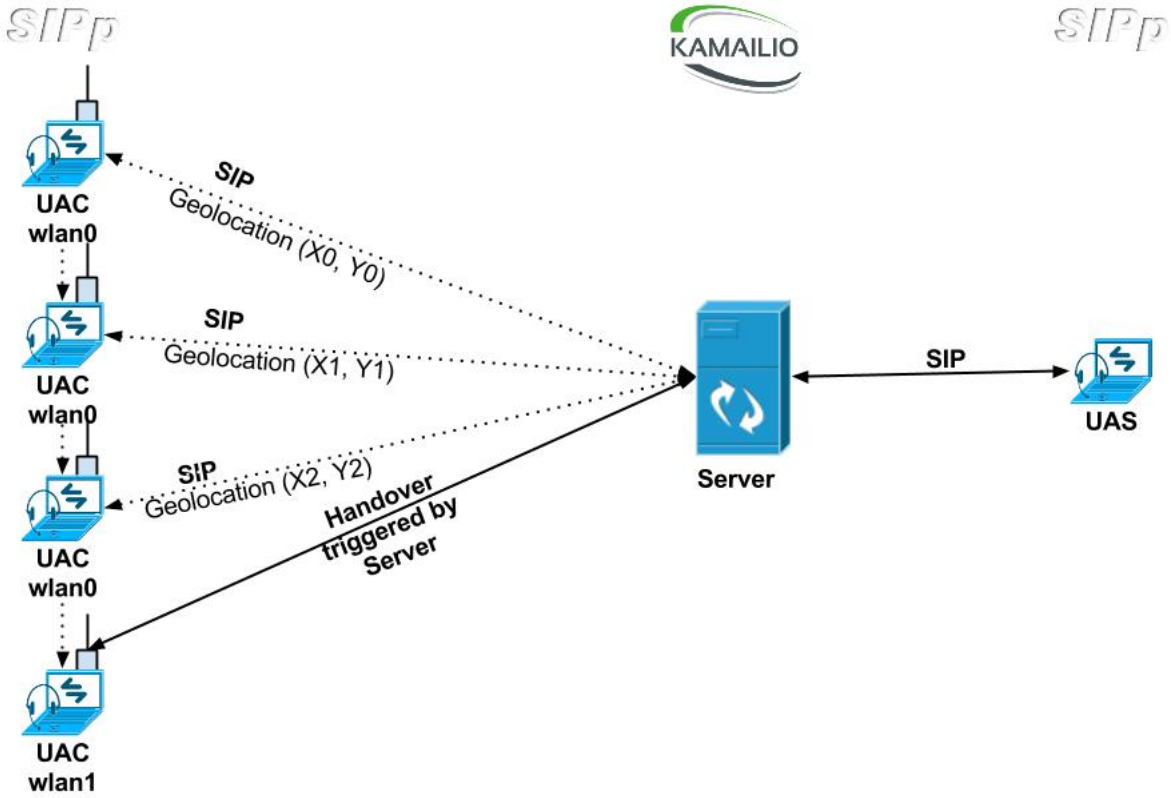

Figure 4.2.1: Geolocation Simulation Setup

We simulate the UAC movement by sending OPTIONS messages to Kamailio, with constantly increasing geolocation content. At a certain point Kamailio takes the handover decision and triggers UAC handover by sending back an OPTIONS to UAC.

4.3 Simulation

This section presents the simulation flow resulted upon SIP server triggering handover.

In order to simulate the above considered scenario, we use third party call control(3PCC) SIPp feature. The feature allows UAS to have separate, yet inter-dependent, SIP logic for communicating with UAC wlan0 and wlan1. This is very helpful when synchronizing the UAS SIP messages in accordance to UAC wlan0/wlan1 interfaces. Have a look on Figure 4.3.1 for a better understanding of the SIPp 3PCC feature.

As you can see, UAS sends 3PCC commands between his server0 and and server1 side in order to synchronize communication with UAC wlan0 and wlan1. UAC sends all the SIP messages to Kamailio which routes them to UAS. UAC sends RTP directly to UAS, after the session has been established.

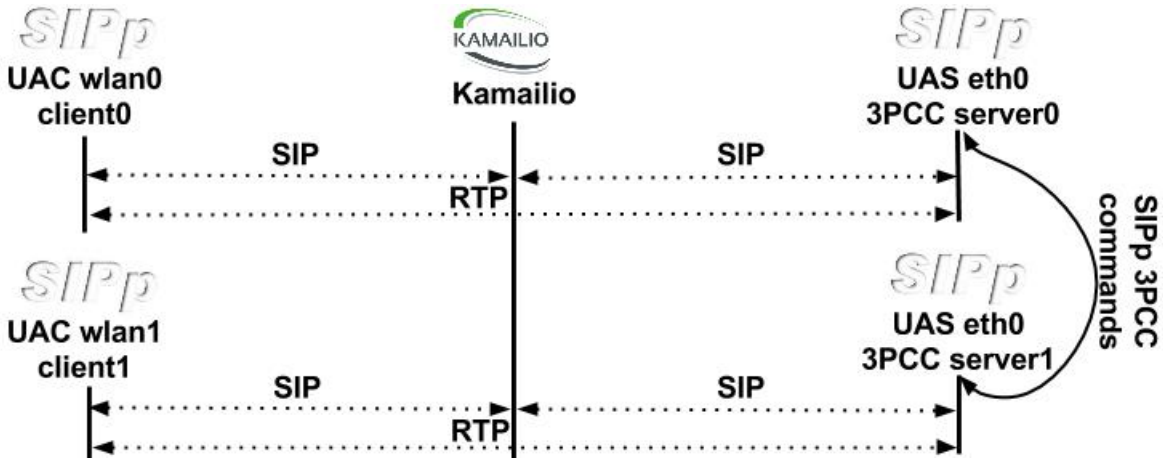

Figure 4.3.1: SIPp Flow

Figure 4.3.2 shows the resulting simulation flow of the proposed solution, using the previously described setup.

First, the session is initially established on UAC wlan0; media flows over UAC wlan0. All SIP signaling is handled by Kamailio. All RTP packets are send on an end-to-end basis.

Next, UAC sends (X,Y) geolocation data in the "Geolocation:" header of the 20, 26 and 32 SIP OPTIONS requests. After doing computations in the running python script, Kamailio decides that UAC has reached out of the initial AP1 and triggers the handover using itself an OPTIONS message.

Last, upon receiving the handover OPTIONS trigger, UAC ports the initial session on wlan1 using one of the SIP handover methods - here the most basic one, based on reINVITE; now media flows over UAC wlan1.

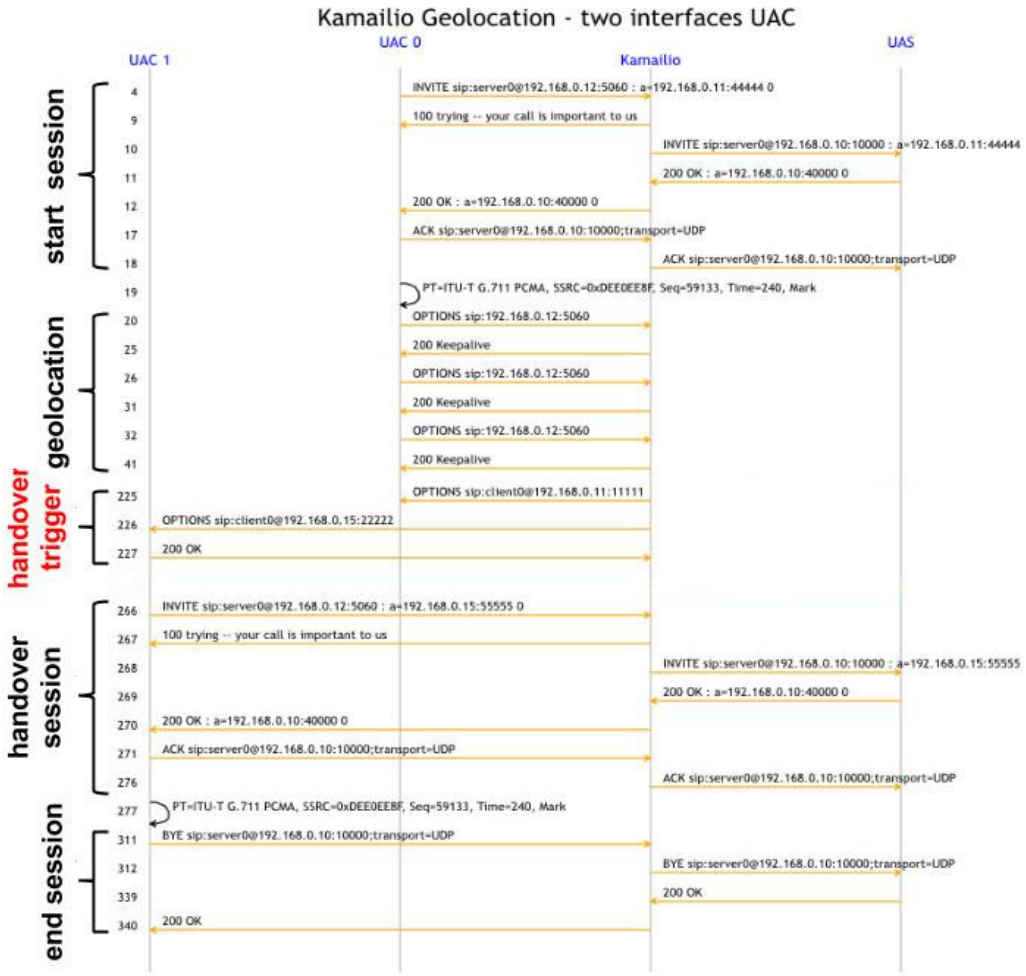

Figure 4.3.2: Geolocation Simulation Flow

4.4 Results

This section presents the impact that this new handover trigger feature has on Kamailio SIP server. To test this we used a virtual machine with 1 CPU, 1 GB of RAM, running Kamailio with 1 child. We used **benchmark** module, ran SIPp scenarios with different handover trigger rate and measured the average handover trigger delay.

In all the tests we used a total number of 1000 triggers. We started from a rate of 1 trigger per second then we gradually increased the rate to 10, 50, 100, 150, 200 triggers per second. The results are shown in the below Table 4.1.

Table 4.1: Handover Trigger Delay Results

Trigger rate (triggers/sec)	Average delay per handover trigger (ms)
1	1.28
10	1.25
50	1.21
100	1.19
150	1.13
200	1.04

Exceeding our expectations, we discovered that with the actual increase in trigger rate, the measured average handover time drops. Asking the community about this behavior, we found out that this may be possible because on higher rates the memory pages get hit more often. Thus, swapping is not happening so often due to the caching done by OS or python interpreter.

We can conclude that the average delay added to the server is around 1.2 milliseconds per handover trigger. Given the fact that in order to not observe the RTP changes, SIP handover should last less than 150 milliseconds, we believe that the delay introduced by the centralized handover trigger feature does not introduce noticeable overhead.

Chapter 5

Conclusions and Future Work

This chapter summarizes the work done in this thesis.

First we begin with the analysis of some current lower layer handover solution. We present the current handover solutions on Layer 1 (ARUBA networks), Layer 2 (DNDP), Layer 3 (MIP), Layer 4 (MPTCP) and Layer 5 (SLM). Next we focus on Layer 7, specifically on SIP protocol.

In the first part of our SIP work we studied, simulated and compared some well known SIP handover schemes. We have come to the conclusion that the fastest scheme is the proactive reINVITE followed by extended proactive reINVITE and simple reINVITE. The slowest method is the one based on REFER method, with a handover time of. This results are shown in Figure 3.2.7.

In the second part of our SIP work we proposed a centralized SIP handover trigger approach using Kamailio SIP server. The server has an overview of the network, more precisely the Access Point distribution. The SIP server monitors the User Agent's position based on the received OPTIONS request, containing Geolocation data. The SIP server triggers handover by sending an OPTIONS message to the User Agent, when the latter reaches out the current Access Point coverage area. The SIP server does all the processing related to User Agent's position in a python script.

We measured the impact of this feature on Kamailio. On average, this feature introduces around 1.2 milliseconds delay per handover trigger. We believe this is an acceptable delay, given the fact that the total handover should last less than 150 milliseconds in order for RTP changes not to be perceived. The benchmark results are listed in Table 4.1.

In the future SIP work we plan to try this handover trigger concept using real User Agents and see how it is behaving. We may encounter issues on the User Agent getting the accurate position relative to its last position due to GPS devices. We may also encounter issues when sending handover OPTIONS because of the network latency.

In the future SIP work we also have the idea to allow or deny an user to make SIP calls based on their location. The security issues of this idea should be considered, given the fact that a user can modify his User Agent to send fake Geolocation to the SIP server.

Bibliography

[1] Ieee trial-use recommended practice for multi-vendor access point interoperability via an inter-access point protocol across distribution systems supporting ieee 802.11 operation. *IEEE Std 802.11F-2003*, pages 0_1–67, 2003.

[2] E. S. Boysen and J. Flathagen. Using sip for seamless handover in heterogeneous networks. In *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2011 3rd International Congress on*, pages 1–8, Oct 2011.

[3] E. S. Boysen, H. E. Kjuus, and T. Maseng. Proactive handover in heterogeneous networks using sips. In *Networking, 2008. ICN 2008. Seventh International Conference on*, pages 719–724, April 2008.

[4] Mititelu Stefan Cristian. research. GitHub, 2015.

[5] Daniel Constantin Mierla et al. Kamailio. GitHub, September 2001.

[6] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure. TCP Extensions for Multipath Operation with Multiple Addresses. RFC 6824, January 2013.

[7] B. Landfeldt, T. Larsson, Y. Ismailov, and A. Seneviratne. Slm, a framework for session layer mobility management. In *Computer Communications and Networks, 1999. Proceedings. Eight International Conference on*, pages 452–456, 1999.

[8] lemenkov, vodik, and wdoekers. SIPp. GitHub, 2006.

[9] M. Liebsch and al. Candidate access router discovery. Internet Draft, September 2004.

[10] A. Mishra, M. Shin, and W. A. Arbaush. Context caching using neighbor graphs for fast handoffs in a wireless network. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 1, page 361, March 2004.

[11] Abdul Nasir and Mah-Rukh Mah-Rukh. Internet mobility using sip and mip. In *Proceedings of the Third International Conference on Information Technology: New Generations*, ITNG '06, pages 334–339, Washington, DC, USA, 2006. IEEE Computer Society.

[12] C. E. Perkins. Mobile ip. *IEEE Communications Magazine*, 35(5):84–99, May 1997.

[13] Maximilian Riegel and Michael Tuxen. Mobile SCTP. Internet Draft draft-riegel-tuexen-mobile-sctp-09, IETF, Individual Submission, November 2007.

[14] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. Sip: Session initiation protocol, 2002.

[15] H. Schulzrinne and E. Wedlund. Application-layer mobility using sip. In *Service Portability and Virtual Customer Environments, 2000 IEEE*, pages 29–36, 2000.

[16] Peter Thornycroft. ARUBA Single Channel Model. White paper, ARUBA networks, 2008.