ENHANCED HOLE PUNCHING FOR RSSI LOCATION TRACKING IN HOSPITALS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

DENİZ PEÇEL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF MEDICAL INFORMATICS

MARCH 2008

Approval of the Graduate School of Informatics

Prof. Dr. Nazife Baykal
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Assoc.Prof. Dr. Erkan Mumcuoğlu
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science of Medical Informatics.

Assoc.Prof. Dr. Murat Erten
Co-Supervisor

Prof. Dr. Nazife Baykal
Supervisor

Examining Committee Members

Assoc.Prof.Dr.Yasemin Yardımcı (METU, II)

Prof. Dr. Nazife Baykal (METU, II)

Assoc. Prof. Dr. Murat Erten (TOBB, CENG)

Dr.Erhan Eren (METU, II)

Dr.Alptekin Temizel (METU, II)

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this wok.

Name, Last name: Deniz PEÇEL

Signature : __________________

ABSTRACT

ENHANCED HOLE PUNCHING FOR RSSI LOCATION TRACKING IN HOSPITALS

Peçel Deniz
M.S., Department of Medical Informatics
Supervisor: Prof. Dr. Nazife Baykal
Co-Supervisor: Assoc.Prof. Dr. Murat Erten

March 2008, 40 pages

With the enhancement of the Radio signal communication systems, Wi-Fi technology become a “de facto” standard used in Campus areas such as hospitals and universities. Besides being used as a data communication method, Received Signal Strength Indicator (RSSI) is also used as a location tracking method. There are lots of studies enhancing the RSSI based location tracking.

In this thesis we tried to generate a test environment as close to a real Wi-Fi network scenario as possible. Our aim is to implement a simple moving client among different wireless local area networks, which is tracked across the internet by a stationary client. We also assumed that there is a Network Address Translation (NAT) at both LAN internet edges.

For continuous data communication, hole punching method is implemented and obstacles with a mobile client are observed. An enhancement to hole punching method is implemented to reduce the loss incurred during wireless network handover. The proposed method reduced the handover duration.

Keywords: Hole punching, Network address translation, RSSI, Location tracking, Mobility

ÖZ

HASTANELERDE RSSI İZ SÜRME TEKNİĞİ İLE KULLANILAN GEÇİT OLUŞTURMA TEKNİĞİ ÜZERİNE İYİLEŞTİRME

Peçel Deniz
Yüksek Lisans, Tıp Bilişimi
Tez Yöneticisi: Prof Dr. Nazife Baykal
Ortak Tez Yöneticisi: Doç. Dr. Murat Erten

Mart 2008, 40 sayfa

Radyo sinyali iletişimi sistemlerindeki gelişmelerle, Wi-Fi teknolojisi günümüzde de facto standart olarak Hastane, üniversite gibi Kampus ağlarda yaygın olarak kullanılmaktadır. Veri iletişimi yöntemi olarak kullanılmanın yanında, alınan sinyal gücü belirteçleri kullanılarak konum belirleme ve takip işlemleri yapılabilmektedir. Bu alanda çok sayıda çalışma hali hazırda sürmektedir.

Bu tez çalışmasında, gerçeğe en yakın bir Wi-Fi test düzeneği oluşturulmaya çalışılmıştır.

Amacımız, farklı yerel kablosuz ağlar arasında hareket eden bir istemci geliştirip, basit bir yöntem ile internet üzerinden uzaktaki sabit bir diğer istemci ile takip edilmesidir. Bu işlemi günümüz şartlarına daha uygun hale getirmek için daha da ileri gidip, her istemcinin NAT arkasında çalışması sağlanmıştır.

Veri transferinin sürekliliği için, geçit oluşturma tekniği yazılımı yapılmış, hareketli bir istemcide karşılaşılan sorunlar gözlemlenmiştir. Bu çalışmada sürenin kısaltılması için bir yöntem önerilmiştir. Önerilen çözümüm algoritması açıklanmış, kurulan test düzeneği tanıtılmıştır. Önerilen sistemin bağlantı kurulum süresini kısaltması beklenmektedir.

Anahtar Kelimeler: Geçit oluşturma, Ağ adresi çevirimi, RSSI, Konum takibi,
Hareketli istemciler

To My Family

ACKNOWLEDGMENTS

I express sincere appreciation to Assoc.Prof. Dr. Murat Erten for his guidance and insight throughout the research. Also I wish to express special thanks to Prof. Dr. Nazıfe Baykal.

Most of the equipments used in this thesis is provided by the TOBB ETU network laboratory and Murat Erten.

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

With the enhancement of Wi-Fi technology, there exist several methods devised for location estimation and enabling continuous communication in case of traversal between networks. There are many usage areas of this technology, such as asset tracking, patient or medical employee tracking and location, remote controlled wheelchairs, remote patient care, telemedicine, indoor communication modules, wireless cameras etc. Everyday new areas and equipment are added to the list. It is inevitable that telecommunication technologies are one of the most important technologies today and in the future for healthcare. Any technology improvement or new method in telecommunication can lead to a better practice in healthcare.

In this thesis, first we proposed network environment that has the highest probability to appear in a real hospital campus and identified the problems that may occur in such a design. We proposed a method to minimize the affects of the real life scenario.

According to this scenario, hole punching registration period of mobile clients moving across Wi-Fi networks when network address translation takes place, observed as a problem. Our method is used to reduce the handoff period when clients move between Wi-Fi networks. According to hole punching algorithm, each client registers to a server as they first connect to the LAN.

Afterwards, server updates each client with other peer information so that each device knows peers IP addresses and can start communication whenever needed. However, whenever an access point registration change is needed, this process starts from the beginning.

In the implemented method, clients can directly initiate the connection rather than all hole punching registration process. Using this method can save lot of time.

1.1 OBJECTIVES OF THE THESIS

The objective of this thesis is to implement the hole-punching method using a “tag-like” single board computer and evaluate the data communication skills in a multi WLAN environment where NAT exists. The following tasks will be realized.

- Design software for single board computer.
- Design a software for the server, used in the registration process
- Design a software for client side, which we will use evaluating the performance.

We evaluated the data communication performance and enhanced the hole-punching technique that might be used in a hospital environment where several WLANs exist.

1.2 THESIS OUTLINE

Chapter 2 gives the background information in detail. We collected several sample Wi-Fi network deployments, their significance. Literature surveys of the recent academic papers are also reviewed in this chapter. Furthermore, we also described the scenario we have created in some detail. The Network Address Translation method, hole punching methodology and how it is used to connect a remote end is described again in Chapter 2. Also the problems that might be encountered during mobility and the proposed method to solve the problem are described in this chapter. Chapter 3 illustrates the test setup and the equipments used in detail. How we took the measurements is discussed in Chapter 4. And finally in Chapter 5, the conclusion, the results we obtained and possible future works are discussed.

CHAPTER 2

BACKGROUND

According to Wi-Fi consultant Jim Geier[16], hospitals have a necessity for Wi-Fi networks. Lots of hospitals are deploying these networks to improve the efficiency, most of the time where there is a patient-traffic jam occurs such as, nursing stations, emergency rooms, doctor offices and patient waiting areas.

There is a common trend that newer doctors and also the nurses are demanding mobile applications. Using traditional paper-based methods are slow, waste of money when you look to the return of investment of Wi-Fi networks. There are lots of researches which predict the hospital market for Wi-Fi deployments is growing with a compound rate of 52% every year [16].

There are four issues to consider in a hospital LAN. Need for Privacy, Spotty Coverage, Roaming and Denial of Service.

Hospitals are, by nature, highly mobile environments. Most of the time users need continuous access to Wi-Fi applications while roaming between patients' rooms, clinics or offices. Typical Wi-Fi architecture should provide a lower level of roaming providing users' ability to move between different access points. Provided that the access points uses different subnets, users will have problems to have continues access to TCP-IP applications.

One very common solution is that to set all access points to use the same subnet assuming the hospital has a quite large subnet. Another possible solution is to deploy MobileIP, which is most of the time vendor specific.

MobileIP is not a good solution for hospitals. There are lots of reasons to choose to implement multiple subnets to a common Wi-Fi network; network management becomes easier, you can easily deploy cheep location based services, the broadcast domain decreases drastically.

For example, a hospital most probably compel to give specific level of services based on the users location, floor, building. By using different subnets thru the Wi-Fi networks, the location information could be easily derived and the services delivered to the patients or care people based on their location. Such a system can deliver maps; guide the doctors to the patients or equipments. You can imagine lots of side usages, such as advertising.

Looking to the picture above Hospitals can match the exact definition. Adam Stone and Julie Ask, analysts from Jupiter Research and Wi-Fiplanet.com explain a real deployment [17]. Network administrator David Gilmour says that wireless is very good way to increase productivity. They used the technology for on bed checks of the patients, enabling access of the information in doctors' offices as well. There is an interesting explanation as; "there was a learning curve for some of the doctors, training them to log back on if they get dropped out of their applications"... So when we are talking about TCP/IP applications there always are timeout period for closing the connections. With our method this can be seen as a scenario as well.

Techworld.com is an online review site which announces annual awards on different projects chosen across the Europe [18]. A project which placed Wi-Fi to the heart of a new hospital IT system was the winner of the Public Sector Project of the Year award in 2006. This hospital was the Royal Hospital.

Royal Hospital, in most common words, determined a demand for secure and robust Wi-Fi solution for its material management system. One of the other primary goals was also replace the paper-based systems that have lots of errors and misinterpretation of handwriting.

They also integrate the Ekahau Wi-Fi technology in this network, so that they are able to locate people and equipments in the real-time. There was an emergency alert button for patients and staff that send messages to the map tools. This feature allows patients to roam around the hospital. Another very impressive benefit was prescription solution on which doctors pass the pharmacy info in the real time by tablet pcs. This significantly decreased the administration time from three hours to one hour.

Doctors and nurses who use the PDA's have mobility and have access to the applications and information from anywhere where allowed within the hospital network. There was a VoIP system that is integrated to the hospitals already deployed telephony system, allowing very efficient communication with all staff.

"Since the wireless network has been installed, we have been able to make considerable cost savings through improved efficiency in the material ordering system; and the design of the wireless network itself means that it has been easy to manage," said Christy Donnelly, network manager for Royal Hospitals.[18]

One of the primary reason for WLANs in hospitals is to ensure that doctors, nurses and other important staffs gets access to all the important information on the move, they should be able to access the important applications as they roam through the patient rooms, clinics, and offices. This being the case the administrators have to ensure that the Wi-Fi coverage is adequate through out the hospital. The problem gets compounded by the fact that both RF and the environment in the hospitals are very dynamic – RF coverage pattern seen today may not be the same tomorrow. Emergency Rooms, highly mobile, fast environment, nobody has tolerance for any time loss.

Received Signal Strength Indication (RSSI) is a measurement of the power present in a received radio signal. RSSI is generic radio receiver technology metric, which usually is invisible to the user of device containing the receiver, but is directly known to users of wireless networking of IEEE 802.11 protocol family. RSSI based location estimation methods are popular nowadays and started to be deployed in industries such as Healthcare, Military, Manufacturing, Mining, Gas and Oil. There are couples of companies providing solutions such as Ekahau Inc. [1], Wherenet, AeroScout, PanGo Networks and Newbury Networks. It is consist of Wi-Fi tags, Positioning Engines and readily deployed Access Points. Basic operation can be explained as following: RSSI values received by Wi-Fi tags from at least three different AP send to Positioning engines by IP networking. Positioning engines calculate the location by Kalman filters [2]. On the other hand this technology is tailored for positioning.

Figure 2.1 Extronics Wi-Fi tag

When technologies such as Wi-Fi, internet, DSL, Firewall, Security Policies, Mobility etc come together in a campus like environment such as universities, hospitals, network architects builds networks according to well known design methodologies. Today almost all campus environments have a WLAN deployed but not in a common way. Most of them lack mobility and support for a peer-to-peer communication because of security policies port blocking. Unfortunately there is no certification and testing used widely, for deployed WLANs evaluating P2P, security, VoIP, Mobility support capabilities. On the other hand, each of these concepts has a different set of requirements. In this thesis our main focus is the problems of mobile stations in a NAT environment and their continuous Peer-to-Peer communication. There are several methods developed for overcoming NAT problem but Hole-punching [3] is the most widely used technique by peer-to-peer software developers. It is an easy and clear method which does not require any new implementation or requirement in NAT devices, firewalls or any other network equipments in the path of communication.

We selected this method since it is the widely used one and thought of several scenarios where we can have a practical problem. We selected a scenario which would be the one of most possible to see in everyday usage and tackling. Our main focus will be data communication.

Several academic researches exist on RSSI location tracking and handover; some of them are presented below,

A recent research done by Network Centric Applied Research Team from Ryerson University, Canada, introduces techniques for modifying and using powered wheelchairs as mobile platforms enabling communication and remote control. The wheelchair runs a PC104 based embedded server allowing both PC and Pocket PC clients to connect in either infrastructure or ad-hoc mode. The clients receive audio, video and other sensory feedback from the wheelchair and can send control data for maneuvering the wheelchair [4]. However in this paper, the possibility of a Wi-Fi network change has not been evaluated. This case can be a very good example for our research since it uses Video and Voice communication in addition to location estimation. The basic setup is shown in figure 2.2.

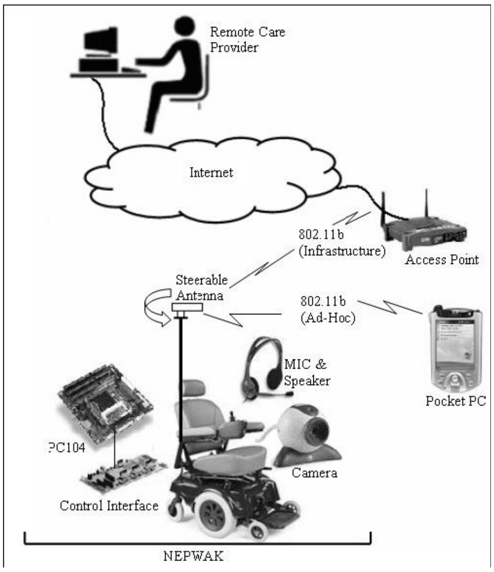

Figure 2.2 Conceptual System

There are many RSSI calibration methods and one approach is the one used by the research continues on Digital and Computer Systems Laboratory of Tampere University of Technology, Finland. The Received Signal Strength Indicator (RSSI) values and Extended Kalman Filter (EKF) are described and enhancement methods are used on a standard Wireless Local Area Network (WLAN) by software computing, thus omitting the need for hardware component additions. The paper is an addition to the enhancement to the reviews of the effects of an office environment to the RSSI- values and present information concerning the effects of base station placement. [5] Since this research is done in pure software environment with focus on location estimation it lacks the real network requirements and obstacles.

Another research which is published in IEEE, deals with Performance of Handoff Algorithm Based on Distance and RSSI Measurements [6]. In this study the performance of a proposed handoff algorithm which is based on both the distance of a mobile station to neighboring base stations and the relative signal strength measurements is evaluated. The algorithm performs handoff when the measured distance from the serving base station exceeds that from the candidate base station by a given threshold and if the measured signal strength of the adjacent base station exceeds that of the serving base station by a given hysteresis level. The average handoff delay and average number of handoffs are used as criteria for performance. The results obtained in this paper lead to the conclusion that the proposed algorithm is relatively insensitive to hysteresis level and threshold distance settings.

When we look to the literature for the studies done supporting and enhancing mobility, one of the most exciting works is done by PCN&CAD center in Beijing University China. An IP layer soft-handover approach for all IP wireless networks is proposed and evaluated [7]. The initial analysis result indicates that the IP layer soft handover scheme can provide better performance than hard handover in overlapped coverage by heterogeneous networks. This is achieved by simultaneous double connection in the edges of to wireless networks. Besides lots of advantages there are also disadvantages which lead us to think in our way. Multi linking will cost more air and network resources, and all copy sending will need complex control algorithms and hardware resources (devices will need at least two Wi-Fi cards). Also in a case where our device is in going from state standby to active, this work has no significance.

There are also other hybrid studies as vertical handover methods [8], Mobile IP solutions [9]. Some other works are also indicated in other chapters according to their relevance to the chapter subject.

2.1 A REAL LIFE SCENARIO

Being able to control Wi-Fi enabled devices over the internet has enabled making laboratory tests, control of robots and even surgeries remotely [10]. Besides these wide opportunities there are some problems. Delays, changes in delays (a.k.a jitter), data transfer errors can be listed as some of them [11]. One of these problems is that the remote device to have a private IP address and consequently the natural need of address translation techniques for a remote wide area connection.

This requirement appears in every network infrastructure, used in almost every hospital, campus, business office and home in either cable or wireless way. Private IP addresses usage is developed to solve the problem of restricted number of public IP addresses. Even when Ipv6 started to be used as a standard protocol worldwide, many of network administrators will choose to continue using private IP addresses because of security reasons.

The devices connected to the Local Area Network share one or more public IP addresses using port numbers. In case there exists more then one wireless access point, each access point can assign its' own private IP address.

These devices can communicate with a server, whose IP address is public, without a problem, only if they initiate the connection. It is not possible to establish a connection in the reverse way without using some methods. These methods must be used bye the clients that reside in private IP address domain. With this method, a hole is established through the local area network border and direct communication from outside that is the public area becomes possible. For this purpose a registration server is used. This method is called Hole Punching [12] and explained in the following sections.

A device using hole punching method to connect a public IP address from a local area network needs to re-initiate the connection if it changes the registered wireless access point. This means a new hole must be established and will take some extra time. If the loss of connection is long enough, we may encounter some undesirable situations. We may loss the control of a robot or we may even loss the position of any tagged device.

There can be other cases which any device may need to re-initiate the connection, such as getting in to stand-by state to conserve battery. In such a case, the control server will need to establish a hole again to activate the device remotely.

In this thesis, there is an enhancement offered to this re-establishing of the hole to minimize the delay. In next part the proposed method is described.

2.2 NETWORK ADDRESS TRANSLATION

As we describe in the previous section, converting private IP addresses to public IP address: port number pair is called Network Address Translation (NAT). The routers maintain the connection between local area network and the outside world and do the network address translation. In the remaining part of this thesis this device will be called NAT server. The description of this process is detailed in the following section.

2.3 CONNECTION USING HOLE PUNCHING METHOD

Hole punching is a method used for establishing connection when the first request comes from the public domain in symmetric NAT environments. In this approach both of the devices might be behind two different NAT servers and LAN, or one may reside in public internet while the other might be behind NAT server. In both cases a registration server is used. The algorithms are summarized below [13]:

When one of the devices resides in public domain:

1. After connecting to network, computers announce their private and public ip addresses to the registration server. (Figure 2.3 a)
2. The device that asks for a connection sends an invitation to remote device via registration server.
3. The registration server sends each others public IP addresses to other side. (Figure 2.3 b)
4. Devices initiate the connection according to the information gathered from the registration server. (Figure 2.3 c)
5. Mutually connection is established.

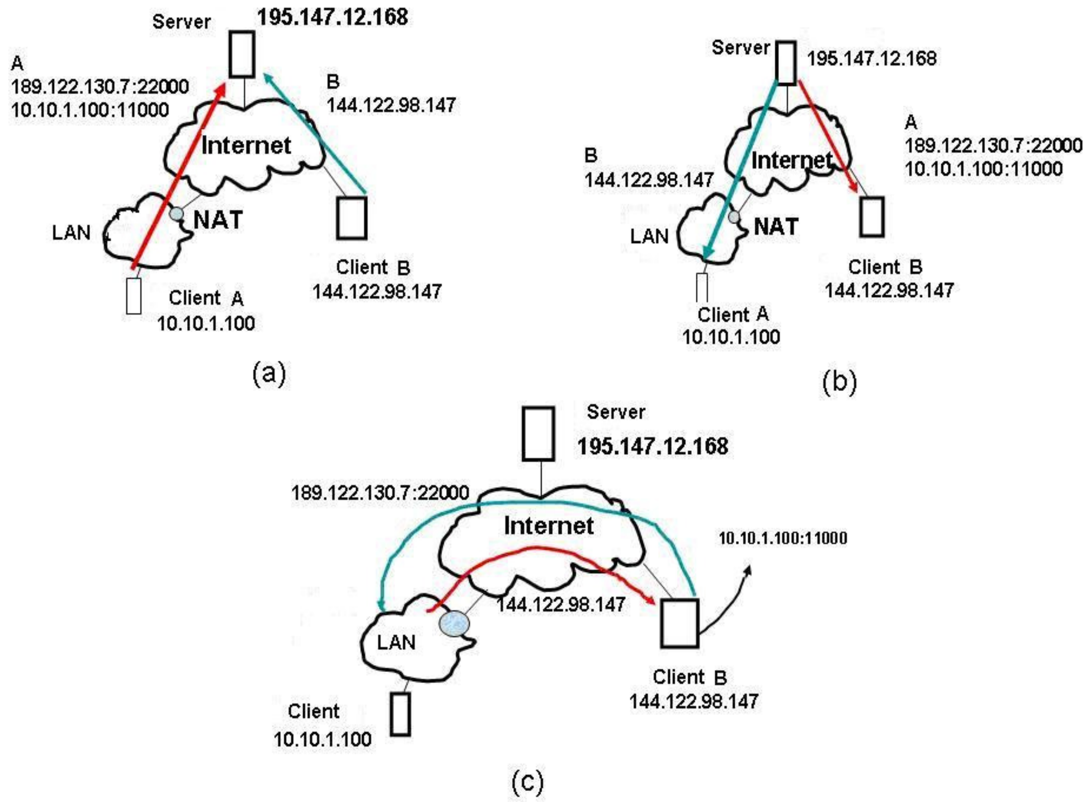

Figure 2.3 (a), (b), (c) Hole punching case when one client is in public domain

After the hole is established, registration server is out of service and devices communicate directly. During the hole establishment, any trial for connection from B to A will fail since A is behind NAT. On the other hand, since A is trying to establish a connection to B, even though A is behind the NAT server, a hole will be punched and a connection will be established between A and B. This approach is called "connection reversal". When B needs to communicate with A, registration server starts the connection from A to B and the problem is solved.

When both clients are behind the NAT servers the situation is even more complicated. (Figure 2.4) Clients register to the registration server in a similar way and both devices is informed about each others private and public IP addresses. As a result of this, both devices initiate a connection to each other and during these requests a hole is established on both sides. During this operation both clients will initiate connection to both IP addresses of the remote peers. If there exists same private IP address in the same LAN, there can be undesirable connection attempts. Using an authentication method could be useful to prevent this situation.

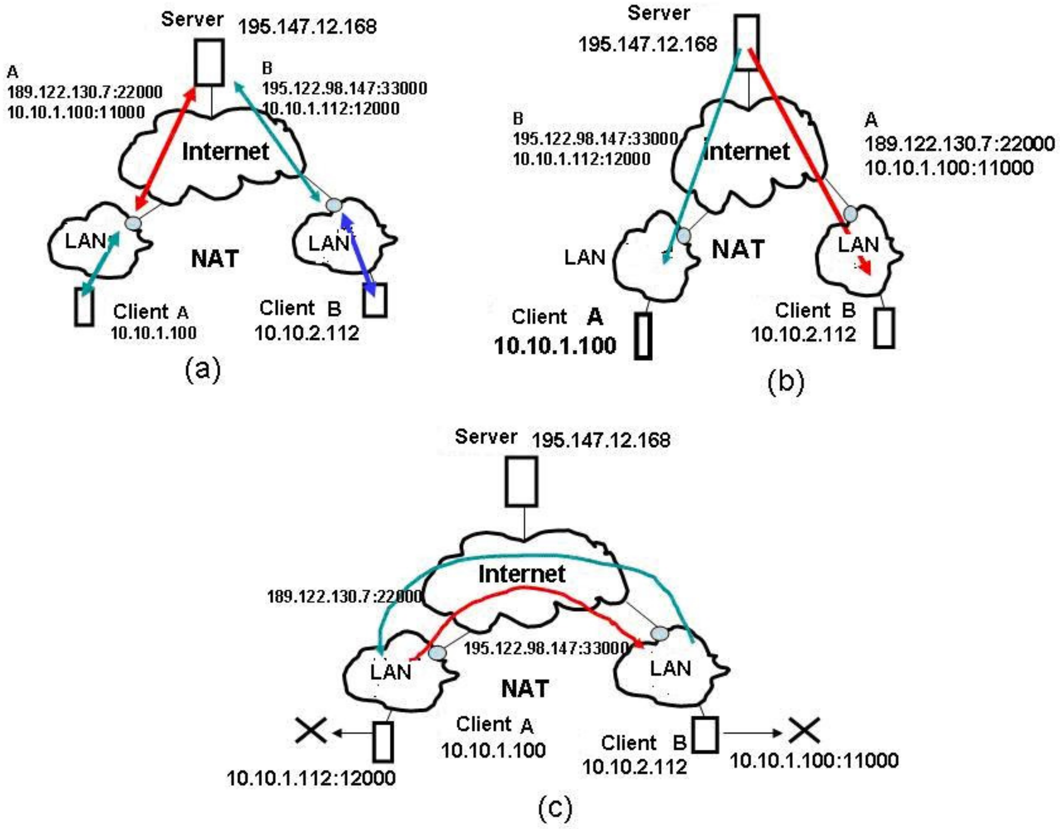

Figure 2.4 Hole punching case when both clients is in private domain

The process of registration and initiating the connections are presented in the figure 2.5.

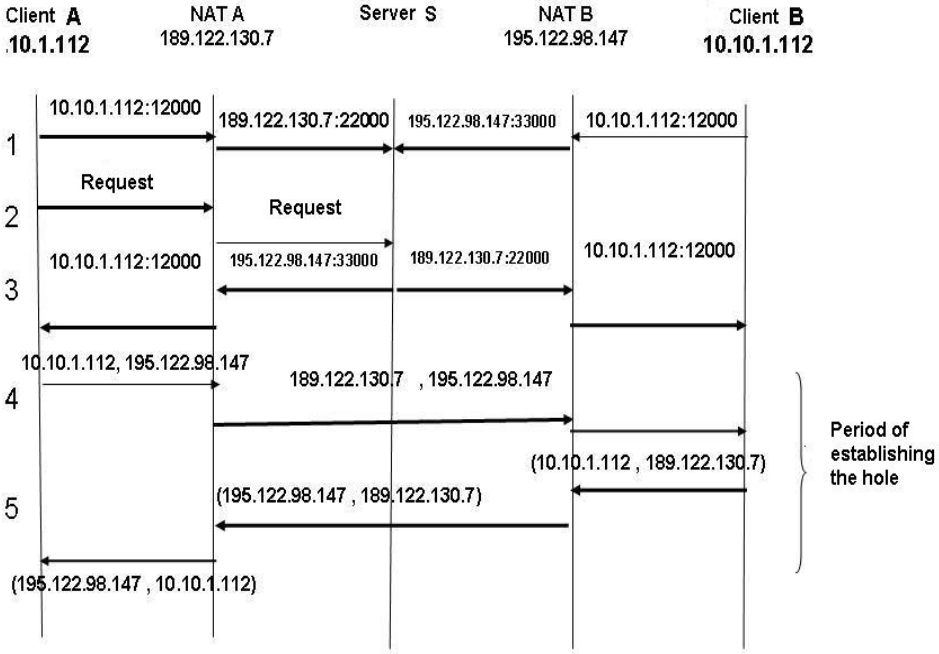

Figure 2.5 The periods of clients' registration to server and establishing the hole

2.4 THE PERSISTANCE OF CONNECTION IN MOBILITY CASE

Considering one of the devices is mobile and have a possibility to change the wireless access point, the entire hole punching process needs to be initiated from scratch. Since the client may acquire new private and/or public IP address some special techniques might be required.

Mobile IP is one of the proposed methods [14]. In this method every device has a Home Agent (HA) to which it is always connected. The Mobile Terminal (MT) registers to the Foreign Agent (FA) whose coverage area is used by MT, receives the IP address from FA and acknowledges HA. A connection from a outside device (B) to the MT is established over HA. HA forwards packets to FA and FA forwards to MT. On the other hand, MT sends its' packets to the B directly. Since the communication triangle B-HA-MT is a waste of resource, there are also some techniques offered that will prevent this communication.

When the methods as Mobile IP and hole punching used together it consumes a lot of time. Depending to the type of applications this delay might be out of tolerance. For the sake of establishing mobility and to decrease the required time the following method is proposed.

2.5 PROPOSED METHOD

This method is derived from a method used for minimizing the interruption in VoIP traffic [15]. According to this, each client registers to the server as they first connect to the LANs. Afterwards, each device knows peers IP addresses and can start communication whenever needed (Figure 2.6).

As a next step, the mobile client is assumed to move from NAT A to NAT C and acquire new IP address from NAT C. In this new state the mobile client is called client A'. It has been shown in figure 2.6 in step 6 below. At this moment, since the A' knows the IP address of B, A' can directly initiate the connection rather than all registration process. Using this method can save lot of time. In the next chapter we shall explain the process and the test setup we propose in detail.

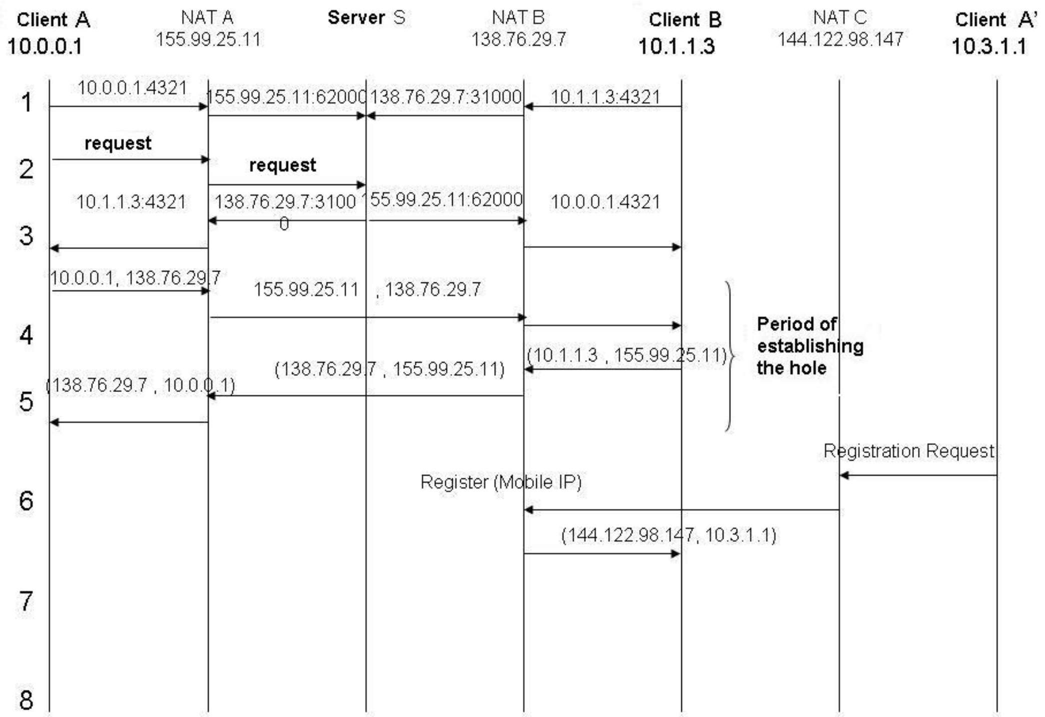

Figure 2.6 The protocol for the Proposed Method

CHAPTER 3

TEST SETUP AND MEASUREMENTS

To test the method described in previous chapter, we setup a test bed as shown in Figure 3.1.

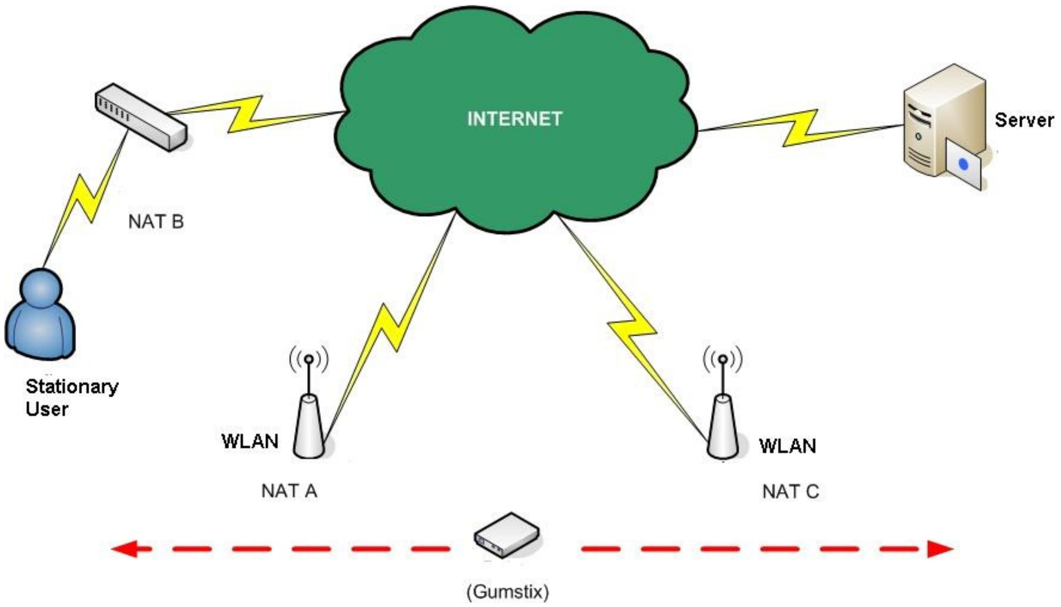

Figure 3.1 Test Setup

To conduct the tests,

- We used a single board computer called “gumstix” as a mobile client.
- Client software and shell script to carry out the tests is prepared.

- A PC to be used as the registration server is installed
- Server side software is developed.
- Another client which is an ordinary PC is used as a stationary client.
- Two wireless access points are installed.

The details of these components are given below:

3.1 GUMSTIX AS A MOBILE CLIENT

Gumstix is chosen as a mobile client because of its small size. Its capabilities are equal to that of a real single board computer which may be used as a Wi-Fi tag or a control plane computer for any remote operation. Its size is 20mm x 80 mm x 8 mm. It has a 400 MHz CPU, 64 Mbytes of RAM and 4 MB of flash storage. (Figure 3.2)

The device has two expansion slots. In this project we used two different expansion cards *cfstix* for wireless communication and *breakout-gs* for serial, parallel ports that can be used for interoperation with many different devices (Figure 3.3 and 3.4).

Figure 3.2 Gumstix Single Board Computer

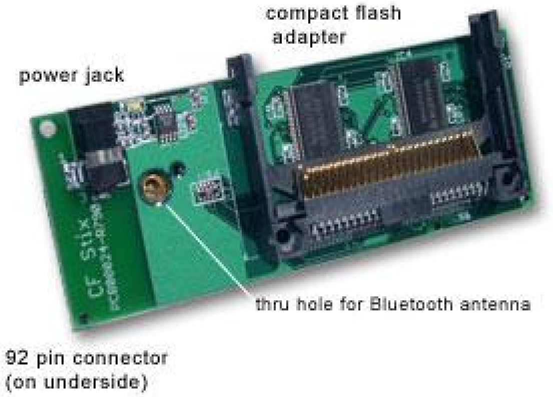

Figure 3.3 cfstix expansion card

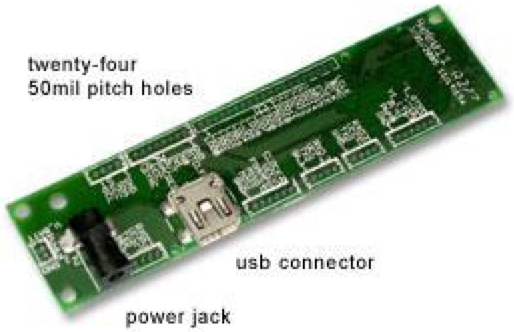

Figure 3.4 breakout-gs expansion card

The operating system of gumstix is tiny version of Linux. There are 200 UNIX commands which are handy to use in this work.

We developed a C code to enable communication with the registration server. Every time gumstix changes its' IP address, it announces its new address to this server. The internet connection of the device is 256/64 Kbits/sec ADSL connection.

3.2 REGISTRATION SERVER

The server is connected to the internet with a public IP address. Each time both clients want to initiate a connection behind their NAT servers, they register to this server and obtain other peers IP addresses. After that communication requests are sent by both ends and using hole-punching method communication is established.

In this test, we used a server with 366 MHz Celeron CPU, 128Mbytes of RAM and 8Gbyte hard disk. The operating system is CentOS 4.2 and has an internet connection speed of 2Mbits/sec.

3.3 STATIONARY CLIENT AND WIRELESS NETWORK

An ordinary Laptop is used as the stationary client which is also behind a NAT server.

In regular wireless networks, IP address is given using DHCP to the clients by the wireless access points. On the other hand, we can change neither DHCP itself nor the wireless access point. In this setup we used 2 Linksys WAP54G as wireless access point. We just used them as a wireless hub and behind them we used two ordinary PC's running SuSE 9.2 Linux as operating system. These computers behave like our NAT servers and uses 'iptables' software to perform NAT operations. We chose this method since we can hack 'iptables' software as it uses Linux kernel command set. As a future work we may implement a new NAT server to run authentication as well.

Since the device IP address will change after moving to next WLAN, the new IP address will be announced to the registration server.

3.4 PROPOSED METHOD

As we described the theory in Section 2.5, our method is different from the ordinary hole punching process because the client is mobile. The main difference and the main advantage are depicted using the Figures 3.5 and 3.6.

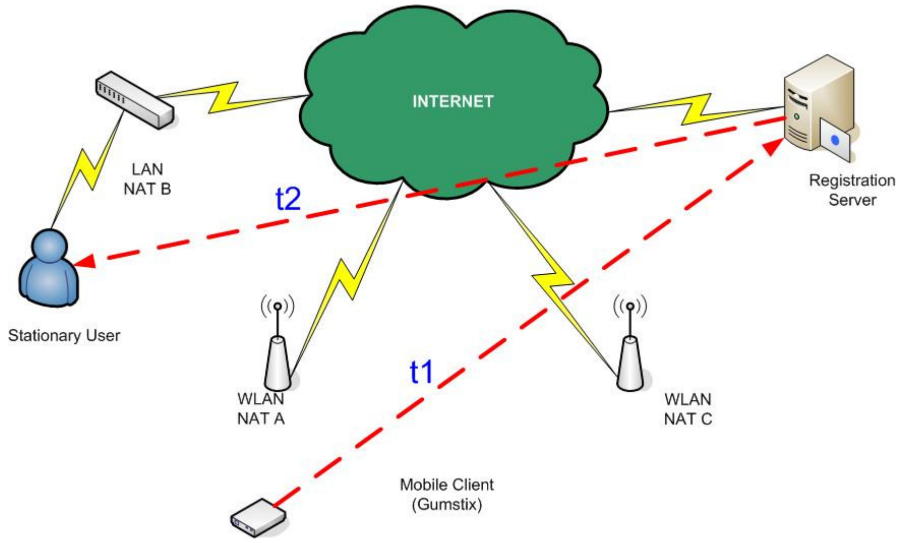

Figure 3.5 Present Situation

With the present situation when the mobile client changes access point, since it will change its public and private IP addresses, the hole punching process needs to start from the beginning. Since t2 and t1 are both wide area connections, total round trip time can be hundreds of milliseconds. The figure below will help us to compare the two methods.

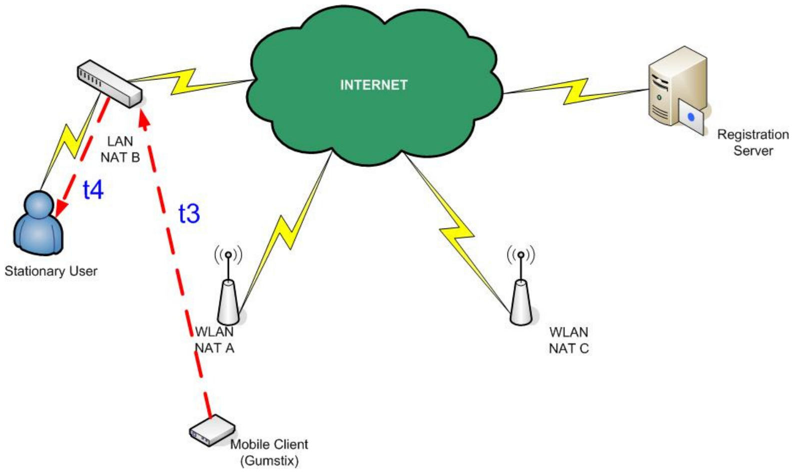

Figure 3.6 Proposed Method

In the proposed method, as the mobile client changes the access point it is registered with, it does not re-initiate the entire hole punching process, instead it just continues with the IP flow, since it already knows the IP address and port number details of the stationary client. In this case, the times required in the proposed method are t3 and t4, where t3 is a wide area connection and t4 is a local area connection. Theoretically, the gain will be at least one wide are connection delay which can be hundreds of milliseconds in lossy network environments. The comments on our implementation can be also found in Chapter 4 and Section 3.5 below.

3.5 MEASUREMENTS

Using the defined test setup, we took some measurements. First, we tested both gumstix and wireless network using UNIX commands. Using “iwlist” command existing wireless networks are scanned, the results are processed using word wrapper commands like ‘awk’, ‘sed’ and Extended Service Set Identification (ESSID) and signal strengths are extracted. Using If-else choosers’ first wireless network is chosen and registered. This is state one.

The shell script continuously scans wireless networks using ‘iwlist’ command and stores the information for each loop in a temporary file. If in any given state, the received signal strength indicator (RSSI) is higher than the existing state, gumstix registers to the new ESSID and changes its IP address. We define a threshold value for this operation. No change will be observed if the difference of the RSSI is below threshold value.

Meanwhile, a shell script and a C code is developed to continuously monitor ESSID, RSSI and IP address information. C code is used to establish UDP connection to server using port 9034 and to implement modified hole-punching method. Whenever the gumstix changes access point, it sends its IP address and ESSID to the server. The stationary client can continuously watch any change in gumstix. The algorithms used are shown below [9]:

1. Listen to existing WLANs.
2. Extract RSSI and ESSID
3. Find stronger signal and compare it to the existing state variable.
4. If it is different compared it to the existing state variable, switch to the new WLAN, change the state variable, record it to changed.log, and return to 1.
5. If the state variable is same just return to 1.

State observer:

6. Read the state variable from *changed.log* file, compares it with the previous value.
7. If it is different, connect to registration server using 9034 port and announce new IP address and new ESSID. Return to 6.
8. If it is same, return to 6.

During the first observations, the period of time to send the information of ESSID and changing it takes couple of seconds. The reason for this was that choosing of the access point and sending this information was written as a single code. As a result, it is understood that when these two jobs is queued, any delay in one of them affects the other part, increasing the handoff time drastically. After splitting these two jobs into two different codes and processes, we observed that the time needed to catch and change the WLAN was reduced to about 600 ms.

If TCP connection is used, to acknowledge the stationary observation client, took up between 900 to 1500 msec. To cut this time we used UDP. The most important part of the delay is actually generated by the Wi-Fi CF cards registering to new WLAN which we can not change. We measured this time as 300 ms.

The delays measured initially, 900 to 1500 msec., are actually caused by the nature of the shell script. In the script, there is an attempt to connect to the registration server before the registration to the new access point is completed. However, since the access point registration process is not complete and the next line which is used for registering to hole-punching server is executed, the process fails and script tries to scan the Wi-Fi networks from the beginning. This causes a huge delay and instability. This delay comes mainly from the Wi-Fi scanning which is measured to be more than 300ms. We, therefore, split the hole-punching registration server connection script into “state observer” and the “signal chooser” scripts, so we can continuously try connecting to the hole punching registration server, after we see that *changed.log* file is modified to show that a stronger Wi-Fi connection exists. This was a problem of our design, and not the algorithm itself.

In this part of the work, a C code is developed to enable the registration of the clients. Using this code, gumstix sends a register message to the registration server. After registration server receives this message, it forwards port and IP address information to the stationary client. In the first connection step, the reverse is also performed, i.e. from the stationary client to the registration server, from the registration server to gumstix. At this stage, both devices can connect each other. When gumstix changes its WLAN, its private and public IP addresses are also changed. Since it knows stationary clients IP address, using the existing hole, acknowledges the stationary client directly about its new IP addresses and can maintain its existing connection. However this method works only in environments using Full Cone NAT. It does not work in symmetric NAT and Restricted Cone NAT.

We also faced with many other problems when configuring and using gumstix. It is a very handy tool, but when we tried to initialize the device some of the processes did not start as expected. Some of the shell scripts are given in the Appendix A. These codes are called initiator; which initiates the gumstix, so that device starts operating without any external action, “S95chooser” is a UNIX start-up script which is the main whole, not separated, shell script and “killer” is an other type of shell script used several times.

3.6 BREAKDOWN OF THE TIMES

Some scripts are written to calculate the time needed to run some of the linux native commands.

```
#!/bin/sh
cat /proc/uptime >> times12
iwlist wlan0 scanning >> tmp1.txt
cat /proc/uptime >> times12
```

Figure 3.7 Calculation of the time required to run ‘iwlist’

Script in figure 3.7 uses the system clock, first records the initial time, than runs the ‘iwlist’ command in the scanning mode, which is run in every cycle in the original code, and just after that records the time again.

```
# more times12
1485.06 1385.87
1485.35 1386.27
```

This process requires around 300 msec as seen above. It has been tried 10 times and the first part of the values represent the time in seconds. The values saved in a file named times12. Above the file container of times12 is shown as an example.

The same script written for DHCP and WLAN selection as in the figure 3.8:

```
#!/bin/sh
cat /proc/uptime >> times12
iwconfig wlan0 essid wireless1
uDHCPc
cat /proc/uptime >> times12
```

Figure 3.8 Calculation of the time required for DHCP and WLAN selection

```
# more times12
10945.38 10385.24
10945.54 10385.27
```

This process takes around 150 msec. As a total 450 msec is required at least for linux native commands. The entire process is repeated for 20 times and the times can be found below:

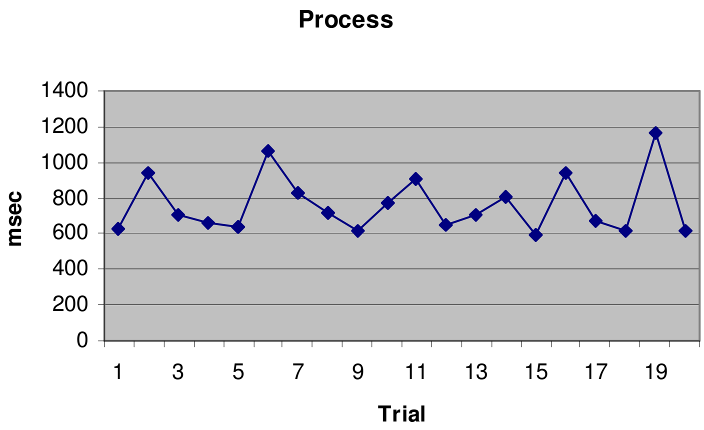

Figure 3.9 Total Process Duration

Table 3.1 The results obtained for the 20 full reconnection trials

Trial #	**Connection time (msec)**
1	630
2	940
3	710
4	660
5	640
6	1060
7	830
8	720
9	620
10	770
11	910
12	650
13	710
14	810
15	590
16	940
17	670
18	620
19	1160
20	620
Mean:	**763**

The sample code used for calculating these durations can be found in Appendix A. The breakdown of these values into process, Wi-Fi registration, scanning and DHCP registration times are shown in Table 3.2 below;

Table 3.2 Breakdown of times with the total process vs other processes

Trial	ConnectionTime (msec)	Scanning Time (msec)	AP Registration (msec)	Hole Punching (msec)
1	630	290	150	190
2	940	290	150	500
3	710	290	150	270
4	660	290	150	220
5	640	290	150	200
6	1060	290	150	620
7	830	290	150	390
8	720	290	150	280
9	620	290	150	180
10	770	290	150	330
11	910	290	150	470
12	650	290	150	210
13	710	290	150	270
14	810	290	150	370
15	590	290	150	150
16	940	290	150	500
17	670	290	150	230
18	620	290	150	180
19	1160	290	150	720
20	620	290	150	180
	Mean: 763			**Mean:323**

The graphical demonstration is also given in Figure 3.10;

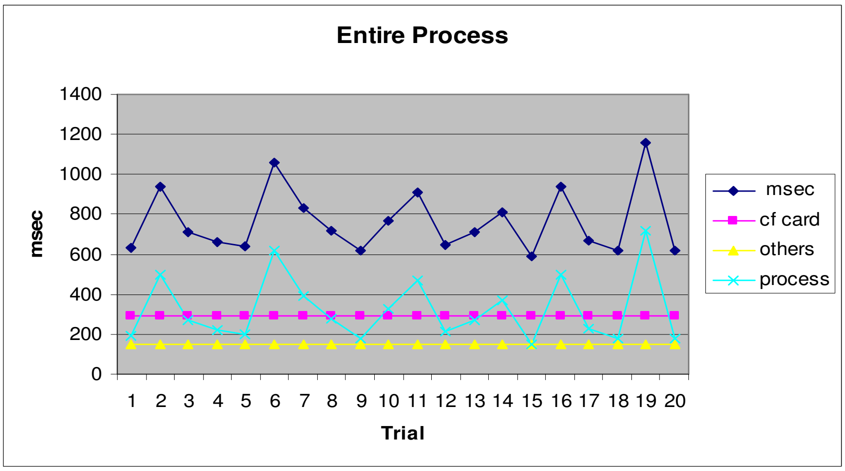

Figure 3.10 Total Process Duration and Breakdowns

We need to highlight some points about these calculations. Total process measurements are made independent of the measurements to evaluate access point registration and scanning times. We first made the 20 trials in Figure 3.10. After that, we measured the DHCP registration, Wi-Fi scanning and Wi-Fi registration processes independently. We do not have a chance to separate them while we are measuring the total connection time as our setup does not allow it. As a result we measured them separately and they appear as constant values in the graphics and tables. The “cf card” plot in Figure 3.10 shows the Wi-Fi scanning process corresponding to the code shown in Figure 3.7. The other plot shown in Figure 3.10 consists of the Wi-Fi registration to the access point and the DHCP IP measurements obtained using the code in Figure 3.8. So the remaining part, which is shown as “process” in the figure 3.10 is the time consumed by our algorithm.

Also we have to note that we could not measure which component causes the fluctuations; since we could not split the times consumed by each component while we measure the entire process. However, it can be said that as the access points lose connections and they do not respond uniformly due to this breakage in the connection, these fluctuations are observed.

CHAPTER 4

DISCUSSION AND CONCLUSION

In this thesis, we proposed a method to enhance a data communication between mobile and stationary devices, which are behind NAT and connected over internet. The existing RSSI location tracking systems focused only on location estimation algorithms, while in near future there will be a huge demand on continuous data communication between end points.

Our design purpose was to implement a tag-like computer, which has energy, size and price constraints. Other methods such as MobileIP, registration to multiple access points simultaneously using multiple network cards is not applicable because of these constraints.

There many other areas this method can be useful; some of them are discussed below:

4.1 POSSIBLE BENEFIT AREAS

Design in this thesis is focused on roaming, Network Address Translation, location tracking, wireless communications since while there are lots of usage areas of these technologies in Health Care. Location tracking is taken as an example in this thesis since it is the most recent wide deployment plans of the hospitals worldwide.

In the introduction part lots of examples given, including public technology awards which are given to the wireless network and technologies deployments in the hospitals, showing that how challenging a Wi-Fi deployment in a Hospital Campus can be. Basically, each time a new technology arises it is associated with the current deployment. Thinking in a long term, we can say that everything in a hospital communication infrastructure will eventually converge to IP and Wireless networks. When technology penetration in Health care has the highest ratio among all other sectors, Health care will be the driver of the new technology innovations.

We stated lot of example usage areas of this technology such as; RSSI location tracking systems for highly mobile environments where application timeouts can be a big problem, remote patient control systems, where patient is monitored continuously, a wheel chair, telemedicine.

We can setup a basic scenario such as in emergency rooms or ambulance, where everybody is highly mobile; nobody has a tolerance for any single time loose. As the patient is carried with Ambulance he/she will be tagged and all measurements and symptoms will be embedded in this tag. It can be also thought as a patient's initial file. This information will be shared with the doctors in the hospital and necessary early treatments can be done remotely even the patient is still on the way. System can arrange needed equipments and check their availability. Doctors can consult each other using latest communication techniques, voice and video conference. It is easy to enhance and detail such a scenario, on the hand it is out of scope of this thesis.

4.2 DISCUSSION OF THE RESULTS

In our implementation we tried to design a test environment as real as we can and experienced the real world problems in mobile client as continues communication. Medical applications, RSSI real time tracking systems, real time communication systems such as voice and video, especially in highly mobile environment as Emergency rooms, are just some places that a solution is obviously handy. Especially in a lossy environment this enhancement can be handy, since any one way communication delay can be up to hundreds of milliseconds.

Our method is an hybrid solution which uses hole punching as an infrastructure and enhances it. This method can be thought as a similar approach as mobile IP approach. However there are lots of reasons to choose this method rather than mobile IP. Network management becomes easier, you can easily deploy cheep location based services, the broadcast domain decreases drastically are just some reasons we identified.

Using these methods we calculated 600 ms of new connection period. On the other hand, it must be noted that 300 ms of this period is caused by the registration of the CF wireless card to the access point, which can not be changed without hacking the device driver. There are also other delays which are not directly related with our implementation such as DHCP and linux native commands caused delays. These commands also add latency measured around 150 ms in each cycle of the shell script. Using linux command is handy and adds scalability since these commands can do their job in high dense environments; they can be manipulated and adds modularity. We believe these times are acceptable for general discussed usage methods in the introduction part. Also we observe some fluctuations in our tests, caused by access point response times can increase sometimes without a reason. This can be because of the firmware of the access points and should be checked with the vendor itself.

One of the most important drawbacks of our implementation is that, it does not work in symmetric NAT and Restricted Cone NAT. It works perfectly in Full Cone NAT. This is a general problem also even biggest internet companies don’t give guarantee to their applications such as Yahoo Messenger/Voice services is not guaranteed to work in every kind of NAT environment. There is still lack of standardization in the deployments and the devices themselves such as there is no industry common certification or validation of Real time communication interoperability. A very brief explanation of the NAT methods can be found in Appendix B.

4.3 FUTURE WORK

As future work we are also working to edit NAT state table algorithm used by ‘iptables’ feature set in Linux kernel. Using this powerful tool you can even directly start any communication whether the devices are behind the NAT or not using a peer to peer authentication. Lot of coding and research is needed for such an implementation but this may attract lot of PhD studies as well.

A person can also choose other methods, such as developing the NAT state table from the scratch, but this will be just discovering the world from the beginning. We made feasibility research on Linux and found it quite appropriate for such an enhancement. ‘Netfilter’ framework enables packet filtering, network address and port translation (NA[P]T) and other packet manipulation in quite stable, easily managed, robust and widely deployed manner. There is a very big online community supporting all kind of enhancements or feature add-ons which are all freely available via website.

So the next step to this research should be the authentication of stateful bindings so that such a handover method can be universally acceptable and stable to all possible attacks.

REFERENCES

[1]Ekahau Inc., Hospital Models, Retrieved January 15, 2008 from http://www.ekahau.com/?id=1010

[2] Latvala, J., J. Syrjärinne, S. Niemi & J. Niittylahti (1999) Patient Tracking in a Hospital Environment Using Extended Kalman-filtering. *Proc. IEEE Middle East Workshop on Networking, Beirut, Lebanon.*

[3] B. Ford, P. Srisuresh, and D. Kegel, "Peer-to-peer communication across network address translators," *Proceedings of the 2005 USENIX Annual Technical Conference.*

[4] A. Arora and A. Ferworn, "Pocket PC beacons: Wi-Fi based human tracking and following," *Proceedings of the 2005 ACM symposium on Applied computing,* pp. 970-974, 2005.

[5] M. Helen, J. Latvala, H. Ikonen, and J. Niittylahti, "Using calibration in RSSI-based location tracking system," *Proc. of the 5th World Multiconference on Circuits, Systems, Communications & Computers (CSCC20001),* 2001.

[6] K. I. Itoh, S. Watanabe, J. S. Shih, and T. Sato, "Performance of handoff algorithm based on distance and RSSI measurements," *Vehicular Technology, IEEE Transactions on,* vol. 51, pp. 1460-1468, 2002.

[7] J. Huang, R. Feng, Y. Bi, J. Wu, and M. Song, "A IP Layer Soft-handover Approach for All-IP Wireless Networks," *Mobile Technology, Applications and Systems, 2005 2nd International Conference on,* pp. 1-4, 2005.

[8] T. Hobfeld, S. Oechsner, K. Tutschku, F. Andersen, and L. Caviglione, "Supporting Vertical Handover by Using a Pastry Peer-to-Peer Overlay Network."

[9] R. Singh, Y. C. Tay, W. T. Teo, and S. W. Yeow, "RAT: A Quick (And Dirty?) Push for Mobility Support," *Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications,* 1999.

[10] P. X. Liu, M. Q. H. Meng, P. R. Liu, and S. X. Yang, "An End-to-End Transmission Architecture for the Remote Control of Robots Over IP Networks," *Mechatronics, IEEE/ASME Transactions on,* vol. 10, pp. 560-570, 2005.

[11] R. C. Luo, K. L. Su, S. H. Shen, and K. H. Tsai, "Networked intelligent robots through the internet: issues and opportunities," *Proceedings of the IEEE,* vol. 91, pp. 371-382, 2003.

[12] B. Ford, P. Srisuresh, and D. Kegel, "Peer-to-peer communication across network address translators," *Proceedings of the 2005 USENIX Annual Technical Conference*.

[13] Pecel D., Erten Y.M., "The Control of robots which are moving between LANs over internet", TOK'06

[14] T. Kato, A. Idoue, and H. Yokota, "Mobile IP using private IP addresses," *Proc. 6th IEEE Symposium on Computers and Communications,* p. 491–497, 2001.

[15] N. Gaylani and Y. M. Erten, "Handling NAT traversal and mobility for multimedia traffic," *Consumer Communications and Networking Conference, 2006. CCNC 2006. 2006 3rd IEEE,* vol. 1, 2006.

[16] Wi-Fi Planet, "Deploying WLANs in Hospitals", Retrieved February 15, 2008 from http://www.Wi-Fiplanet.com/tutorials/article.php/2246471

[17] Wi-Fi Technology Forum, "Wi-Fi Wireless Networks, Hospitals and the Medical Profession", Retrieved February 22, 2008 from http://www.Wi-Fitechnology.com/index.php?name=News&file=article&sid=233

[18] TechWorld, "WiFi in Belfast hospital bags award", Retrieved February 27, 2008 from http://www.techworld.com/applications/news/index.cfm?newsid=6336

APPENDICES

APPENDIX A SOME SHELL CODE

Initiator

```
#!/bin/sh
ps aux | grep S95choser > tmp
pid=`awk '{print$1}' tmp`
echo "$pid"
kill -9 $pid
rm tmp
ps aux | grep connector > tmp
pid=`awk '{print$1}' tmp`
echo "$pid"
kill -9 $pid
rm tmp
ps aux | grep newactiveclient > tmp
pid=`awk '{print$1}' tmp`
echo "$pid"
kill -9 $pid
rm tmp
./connector &
./S95sonchoser
```

Killer

```
#!/bin/sh
ps aux | grep S95choser > tmp
pid=`awk '{print$1}' tmp`
echo "$pid"
kill -9 $pid
rm tmp
```

Time calculation 1

```
#!/bin/sh
cat /proc/uptime >> times12
iwlist wlan0 scanning >> tmp1.txt
cat /proc/uptime >> times12
```

Time calculation 2

```
#!/bin/sh
cat /proc/uptime >> times12
iwconfig wlan0 essid wireless1
uDHCPc
cat /proc/uptime >> times12
```

S95choser

```
#!/bin/sh
i=1
signal2=0
a=0
b=0
while [ $i -le 10 ]
do
cat /proc/uptime >> times
#EXTRACT WIRELLESS INFO
iwlist wlan0 scanning >> tmp1.txt
sed '/level:/s///g' tmp1.txt > iwconfig.txt
awk '/Quality/ {print$3} /ESSID/ {print$1}' iwconfig.txt >>choser.log
awk '/Quality/ {print$3} ' iwconfig.txt > tmp.log
#signal1=`awk '/Quality/ {print$3}' iwconfig.txt`
signal1=`tail -1 tmp.log`
let signal1=signal1*-1
signal2=`grep -v "$signal1" tmp.log`
let signal2=signal2*-1
sed '/ESSID:/s///g' choser.log > choser1.log
sed '/"/s///g' choser1.log > choser.log
rm choser1.log
ESSID1=`tail -2 choser.log | grep -v "$signal1"`
if [ $signal2 -eq 0 ];
        then
        echo 1
        ESSID2=$ESSID1
        else
        ESSID2=`tail -4 choser.log | grep -v "$signal1" | grep -v "$signal2" | grep -v "$ESSID1"`
fi
if [ $signal1 -eq 0 ];
        then
        echo 2
        ESSID1=$ESSID2
fi
#COMPARE SIGNALS
if [ $signal1 -eq 0 ];
        then
        echo 3
        a=3
```

```
elif [ $signal2 -eq 0 ]
        then
        echo 4
        a=4
elif test $signal1 -gt $signal2
        then
        echo 5
        a=5
elif test $signal2 -gt $signal1
        then
        echo 6
        a=6
fi
if [ $a -eq $b ] ;
        then
        echo 0 >> changed.log
elif [ $a -eq 3 ]
        then
        iwconfig wlan0 essid $ESSID2
        uDHCPc
        b=$a
        ./2newactiveclient 212.50.33.13 changed
        echo "changed" >> times
        cat /proc/uptime >> times
elif [ $a -eq 4 ]
        then
        iwconfig wlan0 essid $ESSID1
        uDHCPc
        b=$a
        ./2newactiveclient 212.50.33.13 changed
        echo "changed" >> times
        cat /proc/uptime >> times
elif [ $a -eq 5 ]
        then
        iwconfig wlan0 essid $ESSID2
        uDHCPc
        b=$a
        ./2newactiveclient 212.50.33.13 changed
        echo "changed" >> times
        cat /proc/uptime >> times
elif [ $a -eq 6 ]
        then
        iwconfig wlan0 essid $ESSID1
        uDHCPc
        b=$a
        ./2newactiveclient 212.50.33.13 changed
        echo "changed" >> times
        cat /proc/uptime >> times
fi
```

```
echo "$signal1 $signal2"
rm tmp1.txt
rm iwconfig.txt
rm choser.log
rm tmp.log
signal1=0
signal2=0
done
```

APPENDIX B NAT TYPES

Restricted Cone NAT:

A network address translator where all requests from the same internal IP address and port are mapped to the same external IP address and port. Unlike a Full Cone NAT, with this NAT an external host (with IP address X) can send a packet to the internal host only if the internal host had previously sent a packet to IP address X.

Full Cone NAT:

A network address translator where all requests from the same internal IP address and port are mapped to the same external IP address and port. Furthermore, any external host can send a packet to the internal host by sending a packet to the mapped external address.

Port Restricted Cone NAT:

A network address translator like a Restricted Cone NAT, but the restriction includes port numbers. Specifically, an external host can send a packet, with source IP address X and source port P, to the internal host only if the internal host had previously sent a packet to IP address X and port P.

Symmetric NAT:

A network address translator where all requests from the same internal IP address and port, to a specific destination IP address and port, are mapped to the same external IP address and port. If the same host sends a packet with the same source address and port, but to a different destination, a different mapping is used. Furthermore, only the external host that receives a packet can send a UDP packet back to the internal host.