

Application Layer Mobility Proxy for Real-time Communication

Ping-Yu Hsieh[#], Ashutosh Dutta^ψ, Henning Schulzrinne[#]

[#] Columbia University, New York USA, ^ψ Telcordia Technologies, NJ USA

Abstract:

By configuring some of the application layer tools such as Linux IPCHAINS, IP Masquerading, RTPtrans and IP Aliasing, we build the mobility proxy in cooperation with SIP registrar, to forward data from the correspondent hosts to the new location of the mobile hosts as the mobile keeps on changing its IP address. It not only provides continuous connectivity of both TCP/IP and RTP/UDP traffic for the mobile hosts, but also offers the flexibility to build a mobility proxy in a mobile and wireless networking environment independent of the underlying networking technology. In this paper we present some of the mechanisms involved and experiments conducted to build this mobility proxy in a laboratory environment.

1 Introduction

Continuous connectivity, low transient data loss, short delay in communication, and low cost are some of the major factors to be considered while building mobile and wireless networking systems. Based on the current IP network architecture, forwarding enables mobile hosts to roam among wireless cells or base stations.

Mobile IP [1] and its many of the variants [16], [17], [18] can be considered as some of the solutions to support wireless Internet Roaming for Inter-domain and Intra-domain mobility. These methods extend IP by allowing the mobile to effectively utilize two IP addresses, one for identification (permanent IP address or home address) and the other for routing purposes. There are also home agent and foreign agent functionality to track the location of the mobile and assign proper security association to it. However, the IP address is limited and precious, as well as the system for regular Mobile IP is really complicated and suffers from triangular routing when not used in optimized mode. Therefore it costs more time and money to build and maintain such systems and it becomes a deployment nightmare. Although there have been some application layer mobility management [19] suggested, that take care of most of the common drawbacks associated with basic Mobile IP.

In this paper, we introduce a method to build the mobility proxy in an application layer. It allows both portability and continuous connectivity of TCP/IP or RTP/UDP/IP traffic in a mobile environment by

using SIP Registrar and Linux ipchains/iptables utility. It is known that once the IP address changes, the ongoing TCP connection will disconnect, making it impossible for a mobile host to change its point of attachment to a new IP subnet [2]. By combining the SIP registrar, which records the location and other information of the mobile hosts, with the mobility proxy, that picks up and forwards the data from the correspondent host meant to the original IP address of the mobile host to the moved mobile host, continuous connectivity becomes achievable.

SIP [15] is used to establish, change, and tear down multimedia calls between one or more endpoints in an IP-based network [3]. It is based heavily on some of the most successful protocols to emerge from the IETF such as HTTP and SNMP. IPCHAINS is a default firewall tool in the mainstream Linux kernel from 2.1.102, and it could enable IP Masquerade, which is a networking function in Linux similar to one-to-many NAT (Network Address Translation) found in many commercial firewalls and network routers, to forward data. Therefore, by applying these powerful tools, it is simple and convenient for us to build a mobility proxy in a mobile environment and provide continuous connectivity of TCP/IP and RTP/UDP traffic.

This paper is organized as follows. In Section 2, we survey related work that enable mobile and wireless hosts to be able to continue multimedia communication even if they change their points of attachment while traversing between IP subnets. Section 3 introduces the components and functions used in building the mobility proxy, and then the description of the proposed approach and architecture. In Section 4, we present four network scenarios and experimental results, address the problems we came across, record the measurements. Some of the implementation details with configuration scripts are presented in section 5. Section 6 provides some analytical comparison for handoff with respect to Mobile IP. Finally we conclude the paper in section 7.

2 Related Work

There have been some efforts and mechanisms developed to improve efficiency and continuous connectivity of both TCP and RTP/UDP traffic in a mobile network while there are also some unsolved problems such as dependence on the underlying network and need to change the end systems. I-TCP is one of them [4]. It splits the connection at the wired and wireless border, maintains two TCP connections, thus making the poor quality of a wireless link hidden from the fixed network. However, the splitting connection of I-TCP violates TCP end-to-end semantics. Another approach working at the link layer, snooper, resides at an intermediate node, and caches data

from the correspondent host and inspects their TCP headers [5]. Once the mechanism determines that a packet has been lost, a buffered copy will be sent to the mobile host. Nevertheless, this method has its own flaws, too. Both the former two approaches could not deal with frequent handoffs although they prevent packet loss and bit-errors in a wireless environment. M-TCP resolves this problem by forcing the correspondent host to enter a TCP persist mode when an intermediate node detects a disconnection, but it also splits the connection [6]. One solution for maintaining end-to-end TCP semantics, Fast Retransmit, solves the problem caused by the short disconnections [7]. It forces the mobile host to triplicate to the last old ACK as soon as it finishes a handoff so that the congestion window of the correspondent host will reduce by one half and a packet will be retransmitted immediately. But it will not help too much if the mobile host is disconnected for a long time or frequently since the mobile host's congestion window will get shrunk soon. The proposal TCP-MD&R combining TCP-MD, which detects the movement of a mobile host early on, and TCP-R, which freezes data transmission during registration, minimizes packet loss during handoffs [8]. However, it still could not prevent the delay and complexity of a Mobile IP based wireless system. [20] proposes a new set of migrate options for TCP to provide a pure end-system alternative to network layer solutions. Obviously this approach requires changing the transport protocols in the end terminals. Thus there is a need to devise an application layer proxy that can parse the IP address change of the mobile and will not require any changes by the end-hosts or to the transport protocols. This proposed mechanism will provide a solution independent of the underlying networks.

3 Overview of Mobility Proxy

3.1 Motivation

As the mobile moves to a new domain and new subnet, it will acquire a new IP address if it is not using a Foreign Agent assisted COA. Thus there is a need to record the current IP address with a SIP registrar that can provide the current location of the mobile host. Thus it helps to provide the personal mobility features for pre-session mobility. SIP registrar in conjunction with the mobility proxy helps forward the data from the correspondent host to the moved mobile host. This assumes that end host is equipped with a SIP user agent which sends registration message to the registrar as soon as it moves to a new subnet and acquires a new IP address.

3.2 Mobile-Proxy System Components

Mobile Host (MH): Mobile host is a device which may communicate with the base stations and thus gets access to the internet. It may also be able to travel between different base stations belonging to different subnets.

Correspondent Host (CH): Correspondent host is what the mobile host communicates with. It may be either mobile or stationary.

SIP Registrar: SIP registrar records the IP addresses of the mobile hosts. When the mobile host changes its own IP address, the SIP registrar updates mobile's IP address in its database. When the mobile host moves to a new IP address then the old IP address of the mobile host will be not available, the SIP registrar will send a message with mobile host's updated IP address to the mobility proxy.

Mobility Proxy: Mobility proxy changes the destination IP address of the packets, forwarding the packets to the new IP address of the mobile host. After receiving the message with mobile host's new IP address from the SIP registrar, the mobility proxy will forward the data from the correspondent host to the mobile host with the new IP address. Therefore, the mobility proxy allows continuous connectivity even if the mobile host changes IP address.

IPCHAINS: ipchains is a firewall administration program that creates the individual packet filter rules for the input, forward, and output chains composing the firewall [9]. It replaces ipfwadm, which was used for the old IP Firewall code. We need ipchains to be configured so that we could use IP Masquerading to forward data.

IP Masquerading: IP Masquerading is a form of Network Address Translation (NAT) that allows internally connected computers that do not have one or more registered internet IP addresses to have the ability to communicate to the internet via one's Linux box's single internet IP address [10].

IP Aliasing: IP aliasing provides the possibility of setting multiple network addresses on the same low-level network device (e.g. two IP addresses in one Ethernet card) [11]. It is typically used for services that act differently based on the address they listen on.

IPTABLES: iptables is a direct descendant of ipchains, with extensibility, in Linux 2.4. It is a packet selection system built based on the netfilter framework. The packet selection is used for packet filtering, Network Address Translation, and general packet mangling [12].

RTP Trans: RTPtrans is an application layer forwarding tool that resides in the mobility proxy and forwards the traffic from one address to another one. These addresses can be either unicast or multicast.

3.3 Architecture

To allow continuous connectivity of TCP/IP or RTP/UDP traffic in a mobile environment, we must setup a SIP registrar first, which will record where the mobile host is and the new IP address of the mobile host. The mobility proxy will work in conjunction with the SIP registrar that will get the up-to-date information of the mobile host from the SIP registrar and then could forward the received data from the correspondent host to the moved mobile host in a new IP address. Thus we could achieve continuous connectivity with this approach.

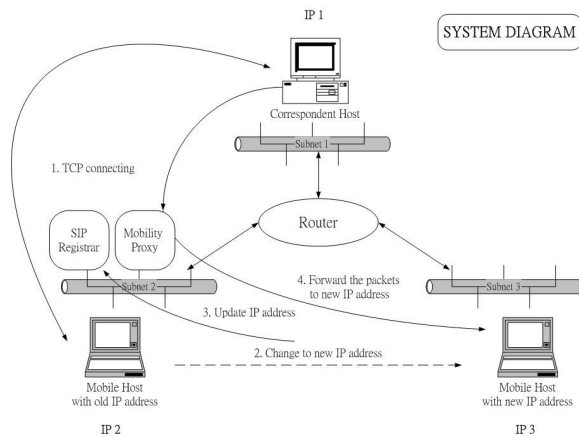


Figure 1

Figure 1 depicts how this kind of system works. Originally, there is TCP or RTP/UDP connection between the correspondent host and the mobile host using an application such as non-realtime application like telnet or real-time application such as vic/rat, where the two hosts are in different subnets. In our experiment the IP address of the correspondent host (CH) is 10.1.3.10, and the IP address of the mobile host (MH) is 192.4.18.193. As soon as the mobile host moves to a different subnet, the IP address of the mobile host will also be changed to a new one, say 10.1.3.20. This new address is obtained using an address acquisition mechanism scheme such as DHCP or PPP. It will then inform the SIP registrar about its move, thus the SIP registrar could update the new IP address of the mobile host. However, the correspondent host doesn't notice the mobile host's IP address change, unless some form of route optimization scheme for Mobile IP or SIP based mobility management scheme is used. Thus it still keeps sending the data to the old IP address of the mobile host, which is 192.4.18.193.

As shown in figure 2, we assume the SIP registrar and the mobility proxy are co-located in a server and its real IP address is 192.4.18.194, which is in the same

subnet as that of the mobile host's old IP address. Since the mobile host has already moved, the old IP address will not be available any more. Therefore, the server must create a virtual IP address 192.4.18.193 as soon as the mobile host updates its

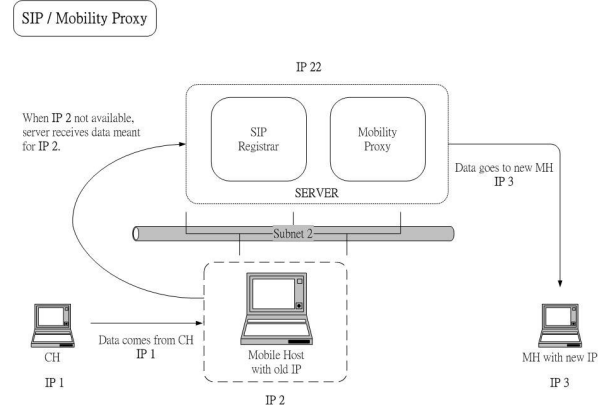


Figure 2

information to the SIP registrar so that the server could receive the data meant for the mobile host's old IP address, 192.4.18.193. At this time, the SIP registrar informs the mobility proxy to forward the data from the correspondent host to the mobile host's new IP address, 10.1.3.20.

4.1 Experimental Scenario

Most of these experiments were conducted in a laboratory environment equipped with IEEE 802.11b access points configured with multiple subnets. Three IBM T20 laptops, one IBM T21 laptop, one Acer 521TE laptop, one NetGear 4 port 10Base-T hub, one Cisco router, several 3Com Ethernet cards, and Lucent ORINOCO IEEE 802.11b AP-1000 access points and PC cards were used for the experiment. The operating system used in the laptops is Red Hat Linux 7.0 with kernel version 2.2.16 for ipchains experiments; in the case of iptables experiments, we tried in Red Hat 7.2 with kernel version 2.4.10. RTPtrans tool could be used irrespective of the operating system since this is an application layer tool. We used some of the following applications such as ping, tcpdump, telnet, Video Conferencing Tool (VIC), Robust Audio Tool (RAT), and whiteboard sharing (wb) for ipchains experiments; we tried several applications such as telnet, ftp, Video Configuration Tool (VIC), modified VIC, Robust Audio Tool (RAT), and tcpchat (a TCP application) in iptables experiments.

4.2 Experimental Results

There are totally four types of network architecture we have tried in our experiments while dealing with Linux tools such as Ipchains/Iptables facility. Experiments with ipchains utility are individually detailed as scenario A, B, C, and D in section 4.3. In the case of iptables experiments, we

have tried three scenarios, two for UDP/RTP and one for TCP based application.

4.3 Experiments using IPCHAINS utility

Scenario A: MH moves in the same subnet

In a regular scenario when a client moves within a subnet it does not change its IP address, as DHCP server usually provides the same address if the client gets re-connected within lease period. But in this experiment we have changed the IP address manually to simulate a PPP address assignment over CDMA 1XRTT link. As shown in figure 3, one of the four laptops acts as the router, that connects the correspondent host (10.1.10.3) in subnet 1, and the proxy server (192.4.18.194) and the mobile host (192.4.18.193) in subnet 2.

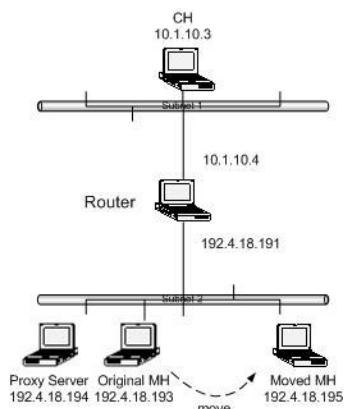


Figure 3

First, we set the gateway of the mobile host to the router, which is 192.4.18.191. Before the mobile host moves, the correspondent host initiates telnet to the mobile host. As soon as the mobile host moves, we execute in proxy server “forward” script that implements ipchains/iptables tools. After moving, the connection is disconnected and the mobile host does not receive any packets from the correspondent host. At this time, if we try to telnet to the old mobile host IP address from the correspondent host again, we could observe from tcpdump that the correspondent host send the connection request to the old mobile host IP address, and the request will be picked up and forwarded to the mobile host in the new IP address by the proxy server at the first time, then the mobile host responds the request to the correspondent host, followed by the correspondent host sending the packet with flag set to R (reset) back directly without passing through the proxy server any more. The connection could not be set up in this situation.

Next, we set the gateway of the mobile host to the proxy server, which is 192.4.18.194. Before the

mobile host moves, the correspondent host initiates telnet to the mobile host. Once the mobile host moving and “forward” executed, the connection is disconnected and the mobile host stops receiving any packets from the correspondent host. If the correspondent host tries to telnet again, through tcpdump, we know the connection request to the old mobile host IP address will be picked up and forwarded to the new mobile host IP address by the proxy server, however, the mobile host will respond through the proxy server, thus the correspondent host could communicate with the mobile host through the proxy server and connection could be set up at this time. On the other hand, if the correspondent host does telnet to the mobile host’s new IP address directly, we could not see any response from the mobile host at all, since the new mobile host IP address is masqueraded by the proxy, so the connection could not be set up.

Scenario B: MH moves to different subnet; proxy server within the router

Figure 4 illustrates that one of the laptops acts as the router with the proxy on it, which connects the correspondent host (10.1.10.3) in subnet 1, the original mobile host (192.4.18.193) in subnet 2, and the moved mobile host (10.1.3.10) in subnet 3.

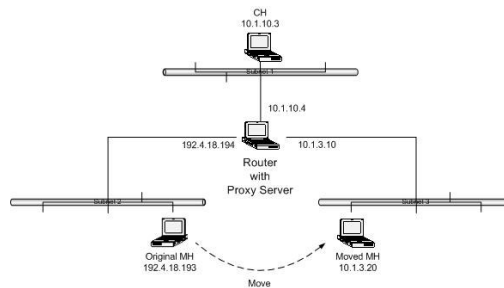


Figure 4

In this scenario, we just set the gateway of the mobile host to the router, which is 192.4.18.194 for the original mobile host and 10.1.3.10 for the moved mobile host, because the proxy is already within the router. Before the mobile host moves, the correspondent host initiates telnet session to the mobile host. As soon as the mobile host moves, we execute in proxy server “forward” command. After moving, the connection is disconnected, so the mobile host stops receiving the packets from the correspondent host. If the correspondent host tries to telnet again to the old mobile host IP address, from tcpdump, we observe that the correspondent host sends the connection request to the old mobile host IP address, the proxy picks up and forwards the request to the moved mobile host, the mobile host responds through the proxy, and thereafter the correspondent host could communicate with the mobile host again through the proxy. On the other hand, if the correspondent host initiates telnet directly

to the mobile host in the new IP address, the mobile host will respond directly and thus the two hosts could communicate with each other.

Scenario C: MH moves to different subnet

Unlike the above network infrastructure, the experiment we have in this section is the more realistic, as shown in Figure 5. We made one of the laptops to act as the router, which connects the correspondent host (10.1.10.3) in subnet 1, the proxy server (192.4.18.194) and the mobile host (192.4.18.193) in subnet 2, and the moved mobile host (10.1.3.20) in subnet 3.

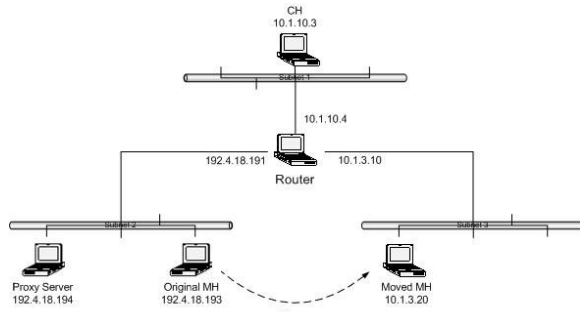


Figure 5

The gateway of the mobile host is set to the router, which is 192.4.18.191 for the original mobile host and 10.1.3.10 for the moved mobile host. The correspondent host again initiates telnet to the mobile, and the proxy server executes “forward” as soon as the mobile host has moved. After moving, the telnet connection seems disconnected, and the mobile host does not receive any packets from the correspondent host. However, once the mobile host moves back to subnet 2, it will continue the last disconnection-like connection. On the other hand, before the mobile host moving back to the subnet 2, if the correspondent sends the connection request to the old mobile host IP address, the proxy server will pick up and forward the request to the moved mobile host in the new IP address, and then the mobile host responds directly to the correspondent host without passing through the proxy server. However, the telnet connection could not be set up after all.

Scenario D: MH moves to different Access Point in different subnet

Before the experiment in this scenario, we first configure two access points in the same subnet and with the same network name or ESSID “WaveLAN”, but configured in different channels, The network architecture in figure 6 shows a configuration with multiple subnets configured over 802.11b networks. We made one of the laptops to act as the

router, that connects the correspondent host (10.1.10.3) in subnet 1, the proxy server (192.4.18.194) and the access point A (192.4.18.193) in subnet 2, and the access point B (10.1.3.20) in subnet 3. The mobile host moves from the access point A to the access point B in this scenario and gets a new IP address from a configured DHCP server in the respective subnet. It is interesting to find that the mobile host will change its point of attachment depending on the distance from the access points or SNR. Furthermore, from the several tests, we could observe that when the handoff takes place, the SNR threshold of the mobile host is around 20dB. We set the gateway of the access points A and B to the router first in this scenario.

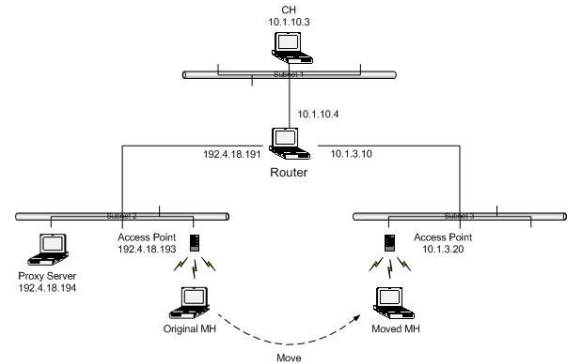


Figure 6

4.4 IPTABLES Experiments

We tried similar experiments with Linux’s iptables facility to make the connection forward to the mobile host in the new subnet. In the experiments of RTP/UDP applications, we used the applications such as rat, vic, and modified vic to verify the connectivity and portability; but for the experiments with TCP applications such as telnet, ftp, and chatcp, it was treated differently. For the TCP based application we break the original connection into two connections in the mobility proxy, so the packets could be forwarded to the new mobile host and the correspondent host is able to receive and deal with the packets from the mobile host in the new subnet. The following are two scenarios for RTP/UDP connection and TCP connection, and we are providing more detailed explanation about the experiments we dealt with.

Scenario A: Experiments for RTP/UDP Applications

In the experiments for RTP/UDP applications, we tried rat, vic, and modified vic applications to see the connectivity and portability of the application. In Figure 7, we made one laptop to act as the router, which connects the correspondent host (205.132.6.11) in subnet 1, subnet 2 which the unmoved mobile host (192.168.80.220) and the mobility proxy server (192.168.80.108) are located in, and subnet 3 that the mobile host is going to move to (192.168.90.220).

Besides, we installed an access point operating at channel 11 in subnet 2, and the other one operating at channel 2 with the same Network Name as that of the access point in subnet 2. In addition, we configured the same Network Name in the wireless network adapter installed in the mobile host so that the mobile host could communicate with one of the access points to stay on line.

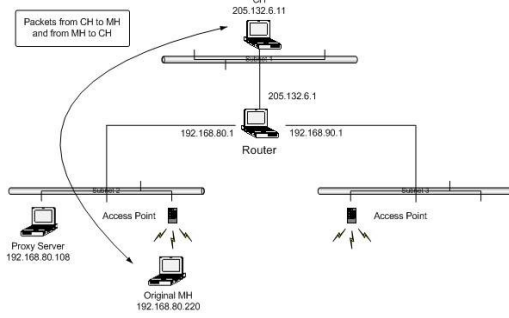


Figure 7

Before the movement of the mobile host, both of the correspondent host and the mobile host start the application and initiate connections to each other, as in Figure 8, the packets from the correspondent host were sent to the mobile host and the ones from the mobile host reached the correspondent host. The two hosts sent out and received the packets, and the applications in both hosts worked just right as we realized.

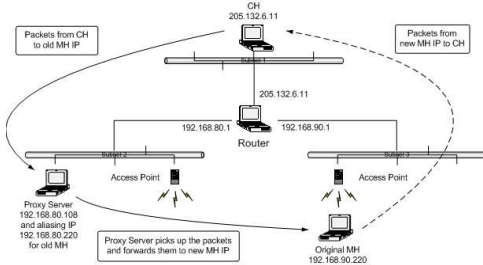


Figure 8

The mobile host then moves to subnet 3, getting the new IP address 192.168.90.220. Forward script is executed on the mobility-proxy by using SIP registration mechanism. Figure 8 shows how the packets were picked up by the mobility proxy and then got forwarded to the new destination of the mobile host even after the mobile host moved.

Scenario B: Experiments of TCP applications

While experimenting with TCP applications such as telnet, we used iptables tools and the mobility proxy, but we needed to break the connection at the mobile proxy. Thus figure 9 shows how the connection was set up before the move, and figure 10 shows how the telnet connection was maintained even after the move

is over and the mobile got a new address. Splitting of TCP traffic was made possible by using iptables tools and NAT module provided by Linux.

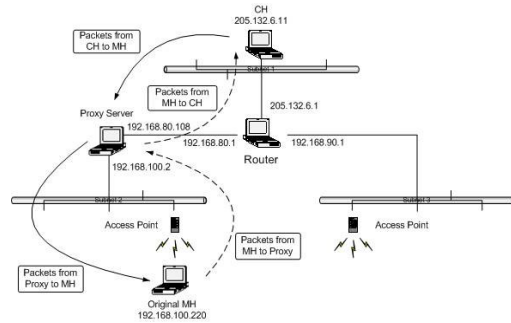


Figure 9 (Proxy splits the TCP connection)

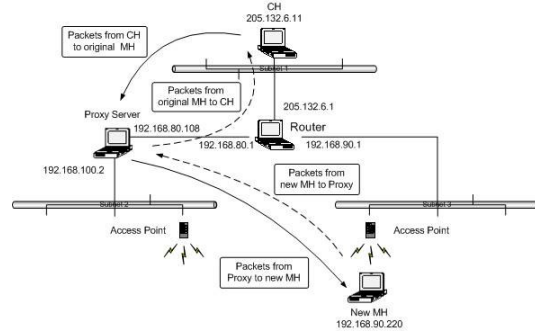


Figure 10

5. Configuration/Implementation snapshot

This section provides examples of some of the scripts those were used for the mobility experiments using ipchains and iptables.

5.1 For IPCHAINS experiments

Before forwarding the messages, we must configure proxy server to make ipchains enabled. For the 2.1 or 2.2 series kernels, the configuration options we will need to set are [13]:

```
CONFIG_FIREWALL=Y
CONFIG_IP_FIREWALL=Y
```

Ipchains:

Following shows a sample forward script run using ipchains tools.

```
#!/bin/bash
#create the IP same as the old MH IP
(192.4.18.193 for example here) [14]
/sbin/ifconfig eth0:0 192.4.18.193
#set the gateway of the proxy server (to the router
(192.4.18.191), for example).
route add -net 10.1.10.0 netmask 255.255.255.0
gw 192.4.18.191 dev eth0
#enable the masquerade function in forward chain of ipchains
```

```

/sbin/ipchains -F forward
/sbin/ipchains -A forward -j MASQ
#configure ipmasqadm to forward packets (port
23(telnet) for example here)
/usr/sbin/ipmasqadm portfw -f
/usr/sbin/ipmasqadm portfw -a -P tcp -L
192.4.18.193 23 -R 192.4.18.195 23

```

This forward script is invoked by a SIP registration. Execution of forward script picks up the data from the correspondent host and forwards it to the moved mobile host.

For IPTABLES experiments:

In this case, we have to enable iptables target extensions for NAT first. Next, we need to install iptables modules by:

```

#!/sbin/inssmod ip_tables
#!/sbin/inssmod iptable_nat
#!/sbin/inssmod iptable_filter

```

Then we could start to use iptables with NAT extension.

Five different scripts were used in the mobile host for changing the IP address (it was needed for the DHCP case however), setting up the NAT functionality on the proxy server. A sample script is given below.

Script “changeip-nat”:

```

#!/bin/ksh
ifconfig eth0 192.168.90.220 netmask
255.255.255.0 broadcast 192.168.90.255
route add -net 0.0.0.0 netmask 0.0.0.0 gw
192.168.90.1 dev eth0
route del -net 0.0.0.0 netmask 0.0.0.0 gw
192.168.80.1 dev eth0
iptables -t nat -A POSTROUTING -p udp --
destination-port ! 5060 -j SNAT
--to-source 192.168.80.220

```

This script is also used for RTP/UDP based applications to change the mobile IP address in the original subnet to the new one in the current subnet and configure the appropriate routing information. However, with this script, the packets initiated from the mobile host in the new subnet will be observed originating from the old IP address which is in the original subnet.

C. Script “changeip-old”:

```

#!/bin/ksh
ifconfig eth0 192.168.80.220 netmask
255.255.255.0 broadcast 192.168.80.255

```

```

route add -net 0.0.0.0 netmask 0.0.0.0 gw
192.168.80.1 dev eth0
route del -net 0.0.0.0 netmask 0.0.0.0 gw
192.168.90.1 dev eth0
iptables -t nat -D POSTROUTING -p udp --
destination-port ! 5060 -j SNAT --to-source
192.168.80.220

```

This script is also used in UDP/RTP based applications when the mobile host returns to the original subnet. It changes the mobile IP address in the new subnet back to the original one in the old subnet and configures the appropriate routing information.

6. Performance Results

There are several issues such as packet loss, duplicate packet detection, transmission delay that may result during mobile’s handoff from one subnet to another. Although these issues are not the focus of this paper, total handoff delay will mostly consist of layer 2 detection delay Δ_1 , delay due to IP address discovery process Δ_2 and delay due to media redirection to the new location Δ_3 . Most of the mechanism described here can be attributed to media re-direction part associated with SIP registration and packet forwarding techniques instituted at the mobility proxy. By instituting an application layer mobility proxy in each subnet, handoff delay due to media redirection will be reduced significantly. This solution looks prominent for Intra-domain mobility management. It was observed that during consecutive subnet movement it takes about few milliseconds to forward traffic to the new location of the mobile. There is still transient traffic before the forwarding is complete that can be taken care of using some kind of fast-handoff mechanism implementing localized multicast address as discussed in reference [21].

In order to provide an analytical approach let us assume that the packet generation rate at CH is P_{rtp} , time taken to register at the SIP proxy is T_{reg} , time taken to complete subnet movement including IP address acquisition and layer 2 movement detection is T_{subnet} , time taken for the mobility proxy to forward the packet after capturing is T_{forw} .

Thus total number of packets lost during the handoff using mobility proxy is

$$P_{\text{mobility_proxy}} = (T_{\text{subnet}} + T_{\text{reg}} + T_{\text{forw}}) * P_{\text{rtp}}$$

As it can be seen from the equation above time taken for registration and forwarding can be optimized based

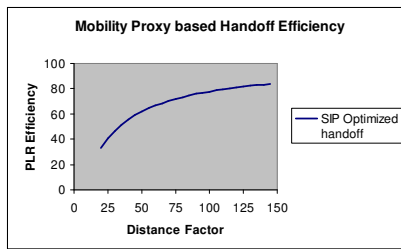


Figure 11

on the type of mobility proxy being used and forwarding techniques such as iptables, ipchains, or rtptrans. Where as time taken for subnet change is independent of the forwarding techniques and heavily depends upon layer 2 detection methods and IP address discovery mechanism. Figure 11 provides an analytical comparison of PLR with and without Mobility Proxy as the distance ratio factor increases.

7. Conclusions

As part of this paper we investigated and experimented an application layer mobility proxy over both wired and wireless networks and used several types of traffic such as TCP based traffic like telnet, and RTP/UDP based traffic such as rat, vic, and wb. This experiment helped us in achieving the portability of real-time and non-realtime communication when the mobile host changes its IP address. Many of the Linux based utilities such as ipchains/iptables were used to provide both portability and connectivity for both real-time (RTP/UDP) and non-real-time traffic (TCP/IP). Besides the Linux based utilities we also have tried an application layer forwarding techniques called "rtptrans" to achieve similar results. As part of future work we plan to study the scalability of this mobility proxy and will investigate as to how many clients it can support over a large number of subnets and when the movement rate is quite fast.

References

[1] Mobile IP, RFC 3220, IETF, Charlie Perkins
 [2] W. Richard Stevens, *TCP/IP Illustrated, Volume 1*, Addison-Wesley, 1994
 [3] H. Schulzrinne and Jonathan D. Rosenberg, "The Session Initiation Protocol: Providing Advanced Telephony Services Across the Internet", *Bell Labs Technical Journal*, Lucent Technologies Inc., October-December 1998
 [4] A. Bakre, B.R. Badrinath, "I-TCP: Indirect TCP for Mobile Hosts", *Proc. 15th Int'l Conf. on Distributed Computing Systems(ICDCS)*, May 1995
 [5] H. Balakrishnan, "Challenges to Reliable Data Transport over Heterogeneous Wireless Networks", *Dissertation*, Berkeley Univ., 1998

[6] K. Brown, S. Singh, "M-TCP: TCP for Mobile Cellular Networks", *ACM SIGCOMM Computer Communication Review*, Vol. 27, No. 5, October 1997
 [7] R. Caceres, L. Iftode, "Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments", *IEEE Journal on Selected Areas in Communications*, Vol. 13(5), June 1995
 [8] Jae-Woo Kwon, Hee-Dong Park, and You-Ze Cho, "An Efficient TCP Mechanism for Mobile IP Handoffs", *IEEE Catalogue Number 01CH37239*, 2001
 [9] Robert L. Ziegler, *Linux Firewalls*, New Riders, 1999
 [10] David Ranch, "Linux IP Masquerade HOWTO", 2000
 [11] Daniel Lopez Ridruejo, "The Linux Networking Overview HOWTO", 2000
 [12] Harald Welte, "The netfilter framework in Linux 2.4", 2000
 [13] Rusty Russell, "Linux IPCHAINS-HOWTO", 2000
 [14] H. Pillay, "Setting up IP Aliasing on A Linux Machine Mini-HOWTO", 2001
 [15] J. Rosenberg, Henning Schulzrinne, Gonzalo Camarillo et al, "Session Initiation Protocol" RFC 3261
 [16] S.Das, A. Misra, P. Agrawal and S.K Das, "TeleMIP: Telecommunication Enhanced Mobile IP Architecture for Fast Intra-Domain mobility", *IEEE PCS Magazine*
 [17] A.T Cambell, J. Gomez, S Kim, A.G. Valko, C-Y Wan, and Z. Turanyi, "Design, Implementation and evaluation of Cellular IP," *IEEE Personal Communication Magazine*, vol 7, no 4 August 2000
 [18] R. Ramjee, T.L Porta, S. Thuel, K Vardhan, and S.Y Wang, "HAWAII: A domain-based approach for supporting mobility in wide area wireless networks," *IEEE Intl conference on Network Protocols (ICNP 99)*, Toronto, Canada, November 1999.
 [19] H. Schulzrinne, Elin Wedlund, "Application Layer Mobility using SIP" *ACM Mobile Computing and Communications Review*, vol 4 no 3, p 47-57, July 2000.
 [20] A.C Snoeren, and H. Balakrishnan, "An end-to-end approach to host mobility," *In Proceedings of ACM Mobicom 2000*, Boston, MA, August 2000
 [21] A. Misra, S. Das, A. Dutta, A. McAuley, "IDMP based Fast-handoff and Paging in IP based 4G Mobile Networks," *in IEEE Communication Magazine*, March 2002