Build a Mobility Proxy

Abstract

Continuous connectivity, less transient data loss, short delay in communication, and low cost are the goals in building mobile and wireless networking systems. Based on the current IP network architecture, forwarding makes mobile hosts possible to roam among wireless cells or base stations. By masterly configuring Linux IPCHAINS, IP Masquerading, and IP Aliasing, we build the mobility proxy, cooperating with SIP registrar, to forward data from the correspondent hosts to the moved mobile hosts. It not only allows continuous connectivity of TCP/IP traffic for the mobile hosts, but also decreases the cost to build a mobile and wireless networking environment. In addition to already equipped functionality that wired networking supplies, wireless networking further provides more convenience and efficiency.

1 Introduction

Activity and movement in our daily lives are so frequent that communication is becoming a more important issue in the modern world. Except for the traditional telephone lines and desktops, people depend much more on cellular and wireless equipments than before. However, the inability of being able to roam continuously and interrupted traffic are always annoying during the communication, so it is crucial for engineers to develop solutions for continuous connectivity.

Mobile IP [1] can be one of the solutions to take care of mobile roaming and it was suggested by many former researches. This method extends IP by allowing the mobile to effectively utilize two IP addresses, one for identification (permanent IP address) and the other for routing purposes. There are also home agent and foreign agent to record the location and validity of the mobile. However, the IP address is limited and precious, as well as the system for 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.

In this report, we introduce a method to build the mobility proxy in an application layer. It allows portability and continuous connectivity of TCP/IP or RTP/UDP/IP traffic in a mobile environment by using SIP Registrar and Linux ipchains 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, which 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 is based heavily on some of the most successful protocols to emerge from the IETF, and it is used to establish, change, and tear down multimedia calls between one or more endpoints in an IP-base network [3]. 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 mobile environment to develop continuous connectivity of TCP/IP traffic.

In Section 2, we survey related work that makes mobile and wireless equipments able to change their points of attachment to new 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. Finally, in Section 4, we present four network scenarios and experimental results, address the problems we came across, record the measurements, and end with conclusion and future work in Section 6.

2 Related Work

Some efforts and mechanisms have been done to improve efficiency and continuous connectivity of TCP traffic in a mobile network while they also have some unsolved problems. 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, snoop, 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.

3 Overview of Mobility Proxy

3.1 Motivation

IP address could not move with the mobile host, so there must be a SIP registrar to record the current location of the mobile host, and a mobility proxy to 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.

3.2 Proxy 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 in 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.

3.3 Architecture

To allow continuous connectivity of TCP/IP 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, then build a mobility proxy, which 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.

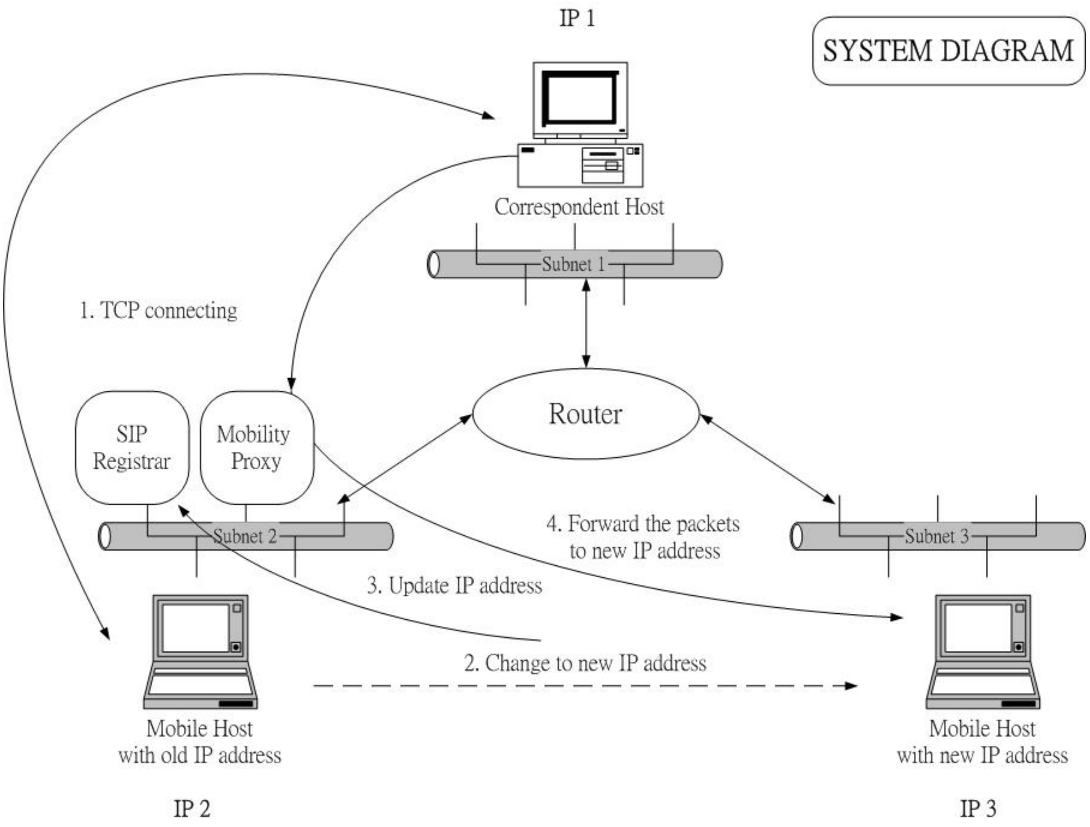

Figure 1

Figure 1 depicts how this kind of system works. Originally, there is TCP connection between the correspondent host and the mobile host using an application such as telnet, where the two hosts are in different subnets. We assume the IP address of the correspondent host is 10.1.3.10, and the IP address of the mobile host 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. 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 data from the correspondent host doesn't notice the move of the mobile host, so it will still send the data to the old IP address of the mobile host which is 192.4.18.193.

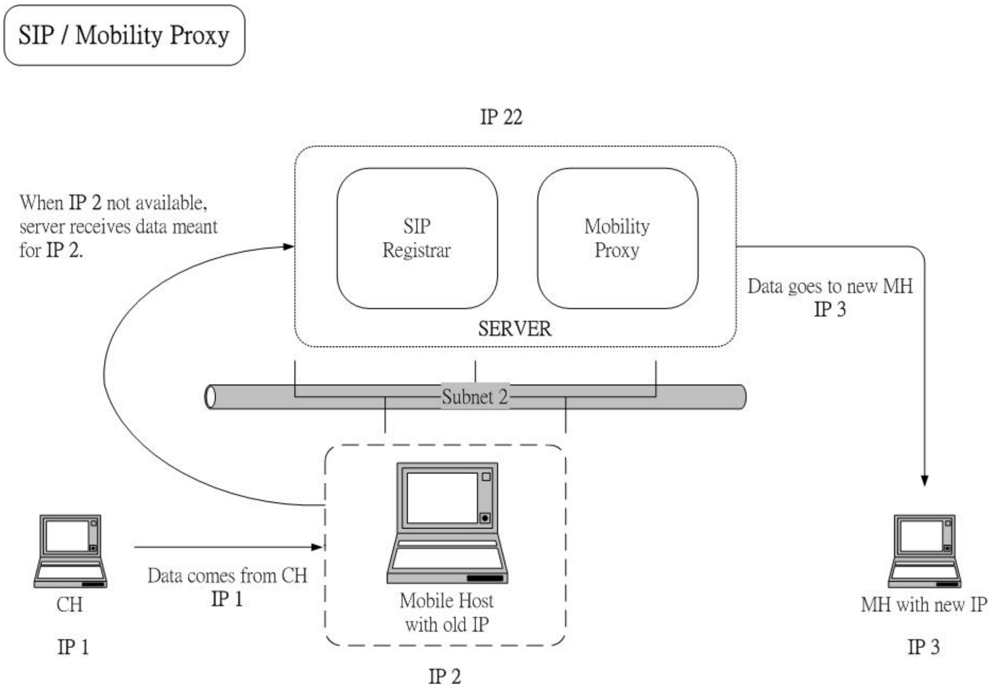

Figure 2

As 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 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 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 Experiment

4.1 Hardware

Three IBM T20 laptops, one IBM T21 laptop, one Acer 521TE laptop, one NetGear 4 port 10Base-T hub, several 3Com Ethernet cards, and Lucent ORiNOCO IEEE 802.11b AP-1000 access points and PC cards

4.2 Platform

The operating system used in the laptops is Red Hat Linux 7.0 with kernel version 2.2.16.

4.3 Application

We used some of the following applications such as ping, tcpdump, telnet, Video Conferencing Tool (VIC), Robust Audio Tool (RAT), whiteboard sharing (wb)

4.4 Configuration

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 [12]:

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

Next, make sure that packet forwarding is enabled (in recent kernels it is disabled by default). We could override this (as root) by typing

```
#echo 1 > /proc/sys/net/ipv4/ip_forward
```

Of course, we could put this somewhere in our bootup scripts so it is enabled every time. Then we have to install the program “ipmasqadm-0.4.2-3.i386.rpm” to the proxy server so that IP Masquerading is ready to work.

4.5 Command

Script “forward”:

```
#!/bin/bash
#create the IP same as the old MH IP (192.4.18.193 for example here) [13]
```

```
/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
```

FORWARD in proxy server picks up the data from the correspondent host and forward it to the moved mobile host.

4.6 Experimental Results

There are totally four types of network architecture we have tried in our experiments, and they are individually detailed in the Scenario A, B, C, and D.

Scenario A: MH moves in the same subnet

Figure 3 indicates that one of the four laptops acts as the router, which 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.

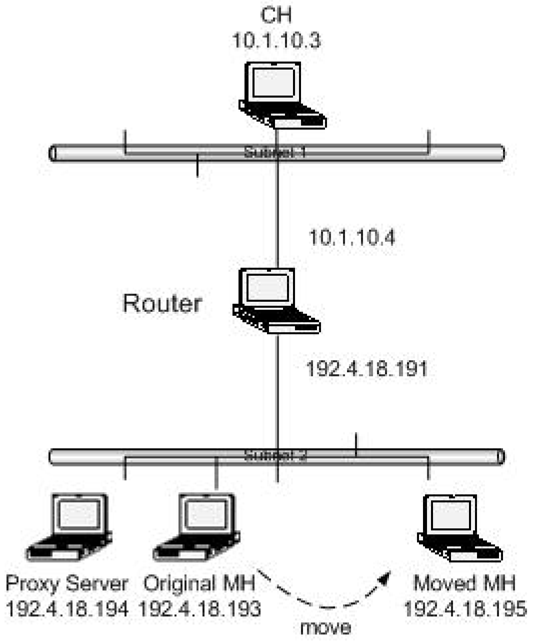

Figure 3

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 is moving, we execute in proxy server “forward” command, which is detailed in Section 4.5. 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 stop 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 telnet to the new mobile host 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.

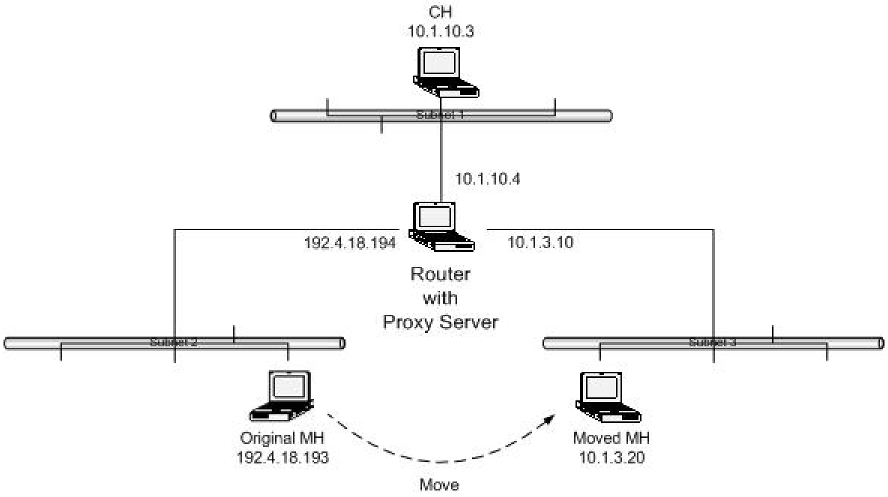

Figure 4

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 to the mobile host. As soon as the mobile host is moving, 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 re-telnet 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.

We also tried Video Conferencing Tool (VIC) and Robust Audio Tool(RAT) in this experiment. In the case of VIC, after the mobile host moving, observing from tcpdump, the mobile host keeps receiving the packets from the correspondent host, however, the video application will stop running, but it is not closed however.

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.

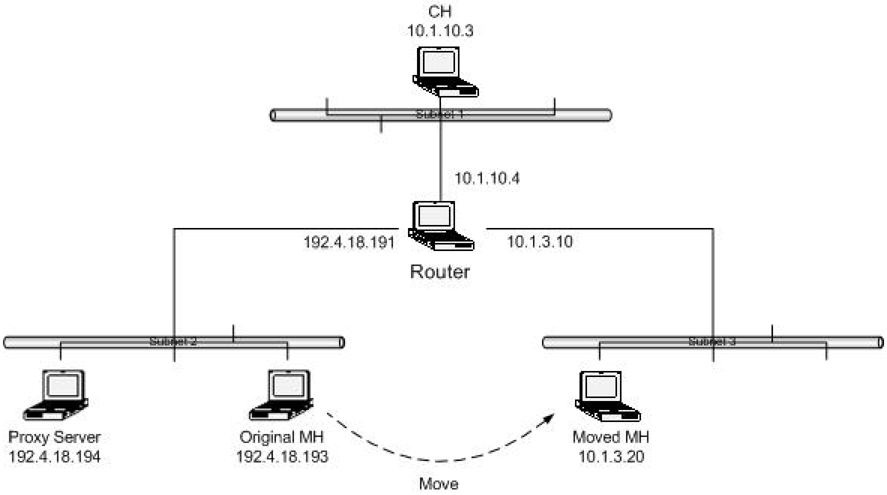

Figure 5

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 is moving. 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 is moving 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 “WaveLAN”, working in different channels, as illustrated in Figure 6, and the mobile host in the same subnet and with the same network name is moved between the two access points. It is interesting to find that the mobile host will change its point of attachment depending on the distance to the access points. Furthermore, from the several tests, we could observe that when the handoff is occurring, the SNR threshold of the mobile host is around 20dB.

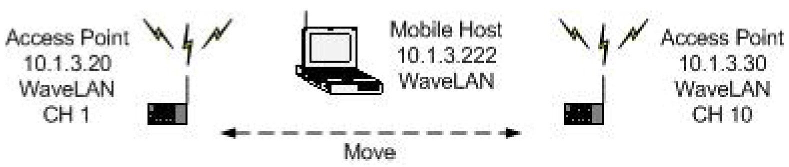

Figure 6

While the access points are put in different subnets, but with the same network name “WaveLAN”, working in different channels, as indicated in Figure 7, and the mobile host, of whom the IP address is assigned by a DHCP server, with the same network name moves between the two access points, the mobile host’s IP address will change accordingly to the subnet of the access point it attached to.

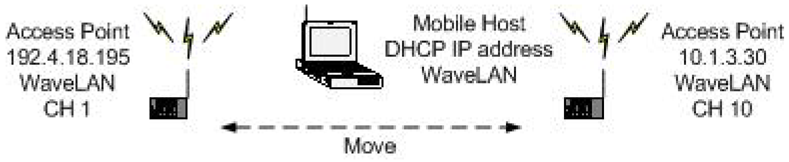

Figure 7

We then return to our experiment. The network architecture in Figure 8 is the common situation encountered in our life. 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 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 will move from the access point A to the access point B in this scenario.

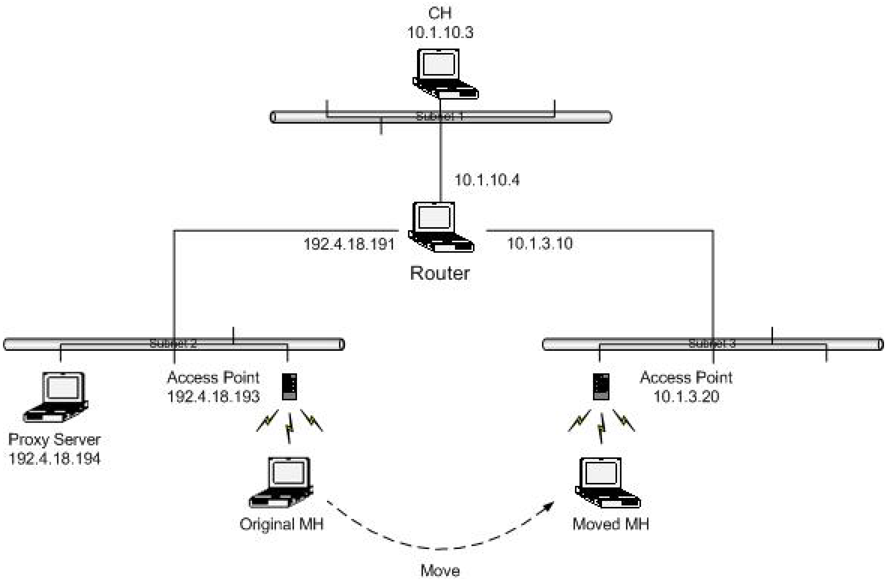

Figure 8

We set the gateway of the access points A and B to the router first in this scenario.

4.7 Measurement

We recorded several tcpdump outputs from different kinds of experiments that we have conducted to see how the packets are actually getting forwarded.
In Scenario A:

When we set the gateway of the mobile host to the router and the correspond host tried to telnet the original mobile host's IP address, we got the following record on the correspondent host.

```
-----------------------------------------------------------------------------------------------
22:22:39.527741 10.1.10.3.1121 > 192.4.18.193.23: S 1006580447:1006580447(0) win 32120 <mss 1460,sackOK,timestamp
709950[|tcp]> (DF)
22:22:39.528612 192.4.18.195.23 > 10.1.10.3.1121: S 1391720938:1391720938(0) ack 1006580448 win 32120 <mss
1460,sackOK,timestamp 302841[|tcp]> (DF)
22:22:39.528649 10.1.10.3.1121 > 192.4.18.195.23: R 1006580448:1006580448(0) win 0
22:22:42.527645 10.1.10.3.1121 > 192.4.18.193.23: S 1006580447:1006580447(0) win 32120 <mss 1460,sackOK,timestamp
710250[|tcp]> (DF)
22:22:42.528509 192.4.18.195.23 > 10.1.10.3.1121: S 1394720340:1394720340(0) ack 1006580448 win 32120 <mss
1460,sackOK,timestamp 303141[|tcp]> (DF)
22:22:42.528543 10.1.10.3.1121 > 192.4.18.195.23: R 1006580448:1006580448(0) win 0
-----------------------------------------------------------------------------------------------
```

From the first line, we know that the correspondent host sent a packet to Mobile host's original IP address (192.4.18.193), and the third line shows that the mobile host in the new IP address (192.4.18.195) responded a packet to the correspondent host. It demonstrates that the packets are forwarded by the proxy to the mobile host. Although the correspondent host received the reply from the mobile host, the telnet was not able to set up the connection.

While we changed the gateway of the mobile host to the proxy, we got the record on the correspondent host:

```
-----------------------------------------------------------------------------------------------
00:19:28.888243 10.1.10.3.1146 > 192.4.18.193.23: S 3834031959:3834031959(0) win 32120 <mss 1460,sackOK,timestamp
1060786[|tcp]> (DF)
00:19:28.889770 192.4.18.193.23 > 10.1.10.3.1146: S 4214543853:4214543853(0) ack 3834031960 win 32120 <mss
1460,sackOK,timestamp 453113[|tcp]> (DF)
00:19:28.889810 10.1.10.3.1146 > 192.4.18.193.23: . ack 1 win 32120 <nop,nop,timestamp 1060786 453113> (DF)
00:19:28.890822 10.1.10.3.1146 > 192.4.18.193.23: P 1:28(27) ack 1 win 32120 <nop,nop,timestamp 1060787 453113> (DF)
00:19:28.891666 192.4.18.195.23 > 10.1.10.3.1146: . ack 3834031987 win 32120 <nop,nop,timestamp 453113 1060787> (DF)
```

```
00:19:28.891719 10.1.10.3.1146 > 192.4.18.195.23: R 3834031987:3834031987(0) win 0
00:19:28.893775 192.4.18.195.23 > 10.1.10.3.1146: R 0:0(0) ack 1 win 32120 <nop,nop,timestamp 453113 1060787> (DF)
00:19:31.890504 10.1.10.3.1146 > 192.4.18.193.23: P 1:28(27) ack 1 win 32120 <nop,nop,timestamp 1061087 453113> (DF)
00:19:31.891420 192.4.18.195.23 > 10.1.10.3.1146: R 4214543854:4214543854(0) win 0
-------------------------------------------------------------------------------------------------
```

The first line indicates that the correspondent host sent a packet to the original IP address of the mobile host.

On the proxy server, we got:

```
-------------------------------------------------------------------------------------------------
23:35:00.272642 < 10.1.10.3.1146 > 192.4.18.193.telnet: S 3834031959:3834031959(0) win 32120 <mss
1460,sackOK,timestamp 1060786 0,nop,wscale 0> (DF)
23:35:00.272751 > 10.1.10.3.1146 > 192.4.18.195.telnet: S 3834031959:3834031959(0) win 32120 <mss
1460,sackOK,timestamp 1060786 0,nop,wscale 0> (DF)
23:35:00.273148 < 192.4.18.195.telnet > 10.1.10.3.1146: S 4214543853:4214543853(0) ack 3834031960 win 32120 <mss
1460,sackOK,timestamp 453113 1060786,nop,wscale 0> (DF)
23:35:00.273192 > 192.4.18.194 > 192.4.18.195: icmp: redirect 10.1.10.3 to host 192.4.18.191 [tos 0xc0]
23:35:00.273234 > 192.4.18.193.telnet > 10.1.10.3.1146: S 4214543853:4214543853(0) ack 3834031960 win 32120 <mss
1460,sackOK,timestamp 453113 1060786,nop,wscale 0> (DF)
23:35:00.273900 < 10.1.10.3.1146 > 192.4.18.193.telnet: . 1:1(0) ack 1 win 32120 <nop,nop,timestamp 1060786 453113> (DF)
23:35:00.273917 > 10.1.10.3.1146 > 192.4.18.195.telnet: . 1:1(0) ack 1 win 32120 <nop,nop,timestamp 1060786 453113> (DF)
23:35:00.274995 < 10.1.10.3.1146 > 192.4.18.193.telnet: P 1:28(27) ack 1 win 32120 <nop,nop,timestamp 1060787 453113>
(DF)
23:35:00.275009 > 10.1.10.3.1146 > 192.4.18.195.telnet: P 1:28(27) ack 1 win 32120 <nop,nop,timestamp 1060787 453113>
(DF)
-------------------------------------------------------------------------------------------------
```

From the first line, the proxy server received a packet meant to the original mobile host’s IP address (192.4.18.193) from the correspondent host, and the third line shows the packet was redirected to the current mobile host in new IP address (192.4.18.195). Since the gateway of the mobile host is set to the proxy server, the replies from the mobile host went through the mobility proxy, indicated by the fifth line, and from the eighth line, the mobility proxy changed the packet as if the packet was replied by the original mobile host’s IP address.

On the mobile host, we recorded:

```
-------------------------------------------------------------------------------------------------
23:17:30.143829 < 10.1.10.3.1146 > 192.4.18.195.telnet: S 3834031959:3834031959(0) win 32120 <mss
1460,sackOK,timestamp 1060786 0,nop,wscale 0> (DF)
23:17:30.143970 > 192.4.18.195.telnet > 10.1.10.3.1146: S 4214543853:4214543853(0) ack 3834031960 win 32120 <mss
1460,sackOK,timestamp 453113 1060786,nop,wscale 0> (DF)
23:17:30.144330 < 192.4.18.194 > 192.4.18.195: icmp: redirect 10.1.10.3 to host 192.4.18.191 [tos 0xc0]
```

```
23:17:30.144984 < 10.1.10.3.1146 > 192.4.18.195.telnet: . 1:1(0) ack 1 win 32120 <nop,nop,timestamp 1060786 453113> (DF)
23:17:30.146118 < 10.1.10.3.1146 > 192.4.18.195.telnet: P 1:28(27) ack 1 win 32120 <nop,nop,timestamp 1060787 453113>
(DF)
23:17:30.146167 > 192.4.18.195.telnet > 10.1.10.3.1146: . 1:1(0) ack 28 win 32120 <nop,nop,timestamp 453113 1060787> (DF)
23:17:30.146756 < 10.1.10.3.1146 > 192.4.18.195.telnet: R 3834031987:3834031987(0) win 0
23:17:30.148277 > 192.4.18.195.telnet > 10.1.10.3.1146: R 1:1(0) ack 28 win 32120 <nop,nop,timestamp 453113 1060787>
(DF)
----------------------------------------------------------------------------------------------------
```

The first line tells the mobility proxy did forward the packet from the correspondent host. In addition, the packets not only could be forwarded by the mobility proxy, but the telnet connection could also be setup.

In Scenario B:

We are conducting more experiments to get the output of tcpdump.

In Scenario C:

We configure the gateway of the mobile host to the router and telnet to the mobile host's original IP address. From tcpdump, we had the record on the correspondent host:

```
----------------------------------------------------------------------------------------------------
19:49:40.659797 > 10.1.10.3.1029 > 192.4.18.193.telnet: S 330039641:330039641(0) win 32120 <mss 1460,sackOK,timestamp
254465 0,nop,wscale 0> (DF)
19:49:40.660888 < 10.1.3.20.telnet > 10.1.10.3.1029: S 4133381342:4133381342(0) ack 330039642 win 32120 <mss
1460,sackOK,timestamp 1300180 254465,nop,wscale 0> (DF)
19:49:40.660934 > 10.1.10.3.1029 > 10.1.3.20.telnet: R 330039642:330039642(0) win 0
----------------------------------------------------------------------------------------------------
```

We could see the correspondent host sent a packet to the mobile host from the first line, and the third line indicates it also received a packet from the mobile host.

On the proxy server, we had:

```
----------------------------------------------------------------------------------------------------
23:22:38.776048 rav4.1029 > 192.4.18.193.telnet: S 330039641:330039641(0) win 32120 <mss 1460,sackOK,timestamp
254465[|tcp]> (DF)
23:22:38.776108 rav4.1029 > 10.1.3.20.telnet: S 330039641:330039641(0) win 32120 <mss 1460,sackOK,timestamp
254465[|tcp]> (DF)
23:22:38.776425 rav4.1029 > 10.1.3.20.telnet: S 330039641:330039641(0) win 32120 <mss 1460,sackOK,timestamp
254465[|tcp]> (DF)
23:22:38.776633 10.1.3.20.telnet > rav4.1029: S 4133381342:4133381342(0) ack 330039642 win 32120 <mss
1460,sackOK,timestamp 1300180[|tcp]> (DF)
23:22:38.777069 rav4.1029 > 10.1.3.20.telnet: R 330039642:330039642(0) win 0
```

```
-----------------------------------------------------------------------------------------------------
```

"rav4" is the correspondent host in 10.1.10.3, sending a packet to the mobile host's original IP address (192.4.18.193) shown in the first line, and in the third line the packet was forwarded to the new mobile IP address (10.1.3.20).

We also recorded on the mobile host:

```
-----------------------------------------------------------------------------------------------------
07:48:45.828499 < zulu.1029 > 10.1.3.20.telnet: S 330039641:330039641(0) win 32120 <mss 1460,sackOK,timestamp 254465
0,nop,wscale 0> (DF)
07:48:45.828551 > 10.1.3.20.telnet > zulu.1029: S 4133381342:4133381342(0) ack 330039642 win 32120 <mss
1460,sackOK,timestamp 1300180 254465,nop,wscale 0> (DF)
07:48:45.835930 < zulu.1029 > 10.1.3.20.telnet: R 330039642:330039642(0) win 0
-----------------------------------------------------------------------------------------------------
```

The packet from the correspondent host "zulu" in the first line was responded by the mobile host in the third line. Therefore, from the above, the packets to the mobile host were forwarded by the mobility proxy, and the correspondent host did get the reply from the mobile host, but the telnet connection still could not be set up.

On the other hand, we also tried the whiteboard sharing application. The gateway of the mobile host is still set to the router, and we recorded the information from tcpdump on the correspondent host:

```
-----------------------------------------------------------------------------------------------------
10:27:44.902509 10.1.10.3.1025 > 192.4.18.193.10000: udp 60 (DF)
10:27:44.934412 10.1.3.20.1046 > 10.1.10.3.10000: udp 60 (DF)
10:27:44.952006 10.1.10.3.1025 > 192.4.18.193.10000: udp 60 (DF)
10:27:44.962530 10.1.3.20.1046 > 10.1.10.3.10000: udp 60 (DF)
10:27:44.976601 10.1.3.20.1046 > 10.1.10.3.10000: udp 60 (DF)
10:27:44.989666 10.1.10.3.1025 > 192.4.18.193.10000: udp 60 (DF)
10:27:44.990589 10.1.3.20.1046 > 10.1.10.3.10000: udp 60 (DF)
-----------------------------------------------------------------------------------------------------
```

It is clear to see that the correspondent host sent a packet to the original IP address of the mobile host from the first line, and the mobile host sent a packet to the correspondent host in the second line.

On the mobile host, we got:

```
-----------------------------------------------------------------------------------------------------
21:47:20.223879 < 10.1.10.3.1025 > 10.1.3.20.10000: udp 60 (DF)
21:47:20.968690 > 10.1.3.20.1046 > 10.1.10.3.10000: udp 66 (DF)
21:47:23.223918 < 10.1.10.3.1025 > 10.1.3.20.10000: udp 60 (DF)
21:47:23.968692 > 10.1.3.20.1046 > 10.1.10.3.10000: udp 66 (DF)
21:47:26.223956 < 10.1.10.3.1025 > 10.1.3.20.10000: udp 60 (DF)
21:47:26.968691 > 10.1.3.20.1046 > 10.1.10.3.10000: udp 66 (DF)
```

```
21:47:29.223992 < 10.1.10.3.1025 > 10.1.3.20.10000: udp 60 (DF)
-----------------------------------------------------------------------------------------------------
```

According to the output on the correspondent host, we could realize that the packet was forward by the mobility proxy and the mobile host did get it, indicated in the first line; however, the mobile host sent a packet to the correspondent host without passing through the mobility proxy.

The packets from the correspondent host were forwarded to the mobile host, and the whiteboard application on the mobile host could work with these packets; however, although the correspondent host could get the packets from the mobile host, the whiteboard application in the correspondent host was unable to perform them.

In Scenario D:

We are conducting more experiments to collect tcpdump outputs.

5 Summary and Future Work

As part of this project, we investigated and experimented the mobility proxy over wired and wireless networks and used several types of traffic such as TCP/IP based traffic like telnet, and RTP/UDP/IP based traffic such as rat, vic, and wb. This experiment helped us in achieving the portability of this communication when the mobile host changes its IP address. As part of future work we plan to investigate iptables utility that comes with the most recent version of Linux 2.4 to see if it would provide any better mobility support. We would also plan to make this mobility support more automated.

References

[1] Mobile IP, RFC 2002, IETF, by Charlie Perkins

[2] W. Richard Stevens, *TCP/IP Illustrated, Volume 1*, Addison-Wesley, 1994

[3] Henning G. 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] Ajay Bakre, B.R. Badrinath, “I-TCP: Indirect TCP for Mobile Hosts”, *Proc. 15th Int’l Conf. on Distributed Computing Systems(ICDCS)*, May 1995

[5] Hari Balakrishnan, “Challenges to Reliable Data Transport over Heterogeneoous Wireless Networks”, *Dissertation*, Berkeley Univ., 1998

[6] Kevin Brown, Suresh 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] Rusty Russell, “Linux IPCHAINS-HOWTO”, 2000

[13] Harish Pillay, “Setting up IP Aliasing on A Linux Machine Mini-HOWTO”, 2001