SEAMLESS MULTIMEDIA COMMUNICATION OVER HETEROGENEOUS NETWORKS: A LINUX DAEMON APPROACH

S.V.Srikanth, Dr. Sarat Chandra Babu, Dileep K Panjala, I Trilok
Centre for Development of Advanced Computing (C-DAC)
{venkatas,sarat,dileepkp,triloki}@cdac.in

Abstract

Our proposed solution is a Linux daemon approach to vertical handoff for heterogeneous networks. We develop a daemon that accomplishes the handoff process between various radio interfaces supporting IP. This addresses seamless continuity, handoff decision & interface selection problems faced during the vertical handoff. This daemon works for all the multimedia networking applications running on Linux systems. The handoff can be achieved at the client devices and there is no need of any handoff agents or servers required. Linux being the fastest growing operating system for wireless devices and the flexibility it offers for the developer makes it the most suitable for the implementation of this solution. This daemon runs at the user level and the results show that the handoff is significantly faster for all the multimedia networking applications.

Keywords: Heterogeneous Networks, vertical handoff, Linux Daemon, WLAN, multimedia

1. Introduction

Today, there is a gradual trend towards using not only one network, but to utilize all interfaces available on a mobile device. To keep track of this evolution, the use of heterogeneous networks has gained focus [Nikolaou et al (2002)]. Heterogeneous Networks means networks of different types. Instead of only using one type of network, it is desired to utilize all interfaces available on a device, and hence choose the one best suited for communication. The challenge in this context will be to provide handoff without user interaction.

On most of the devices today wired and wireless LAN cards have become basic components, apart from supporting other radio access technologies such as Bluetooth, GPRS etc [Wei-Cheng Xiao et al (2007)]. If these devices are to connect to various networks and perform a continuous data transmission whatever location they roam to, while switching from one network to another, technique of vertical handoff is very much needed.

2. Problem

Various complementary access technologies namely WLAN, Bluetooth, GSM, WiMAX etc are developed with service intended to cater to different sections of mobile users. Each of these technologies has its own diverse set of standards, protocols and support homogeneous network mobility in their own way. For example, a mobile device can move between Access Points of a WLAN or WiMAX, handoff occurs when there is movement between cells of the same network type. With the current infrastructure and the increase in heterogeneous networks, it frequently possible to have wireless access of one type or another everywhere, yet there is not a single strong architecture, which embraces all these technologies and provides seamless interoperability between these disparate interfaces. Since these wireless infrastructures have developed independently of each other, we now have to integrate them in order to achieve seamless roaming [Mahesh (2005)]. The need of the hour is to achieve vertical handoff without user intervention.

There are several key issues that are associated with the implementation of vertical handoff 1) Seamless Continuity 2) Handoff Decision & 3) Interface Selection. First, we have to ensure that the device makes a switch between hardware interfaces to connect to the best available network. In our case, whenever a connection to Ethernet is available, we connect to it and switch to the WLAN or cellular when there is mobility. Next, the switching between the networks has to be transparent to the user. The wireless user should not have to configure anything for the switch to occur. In the homogeneous network handoff mechanism, the network infrastructure makes it possible for wireless devices to move without disrupting the existing connection. Similarly, heterogeneous network switching should also ensure that the ongoing session is maintained. We have to make sure that the switching is fast enough so that the user connection is kept intact at the TCP/UDP layer.

3. Assumption

Existing devices cannot incorporate unlimited number of interfaces and so we restrict our experiments to have two different networks interface cards 1) Ethernet card 2) WLAN card. It is also assumed that the proposed architecture also suits different network interface cards supporting IP. We use a 2.6.32 version of the Linux Kernel that is suitable for running on a Laptop or an embedded device with higher computing power.

4. Related Work

There are several approaches to achieve seamless mobility in heterogeneous networks. Some of the approaches are generalized as follows.

Currently, there is a significant trend toward using not only one wireless network, but to utilize all carriers available on a mobile device. In addition, for real-time applications like voice, it is desired to make a seamless change of network without user interaction. To keep track of this evolution, the use of heterogeneous networks has gained focus. Instead of only using one type of network, it is desired to utilize all carriers available on a device, and hence choose the one best suited. Furthermore it is proposed an application-layer handover scheme for session continuity in heterogeneous networks [Hakon et al (2007)]

Conducting the handover process at a higher layer can provide providing transparency to the heterogeneous seamless handover. By that way, efficient handover decisions for vertical handover are made with more number of constraints that will lead to high performance, accessibility and low cost. Making this possible is by providing Quality of Experience (QoE) and obtaining current information of the throughput by measurements in the network. Analyzing and interpreting the statistics collected through measurements are vital in terms of decision-making and to decide when to perform vertical handover. This project consists of the implementation of measurement module in two different approaches (Payload Dependent Approach and Payload Independent Approach) that will provide these statistics to the storage module in PERIMETER project. [Selim (2010)]

Now a days mobile devices support several interfaces although the protocol stack used in each interface tends to be interface specific at the lower layers. This limits the ability of a device to switch back and forth between networks as need and opportunity dictates. As a step towards providing a more flexible handover infrastructure, this project addresses the issue of integrating heterogeneous, mobile ad-hoc networks that use different MAC layer protocols. The goal is to provide an end-to-end communication abstraction that hides heterogeneity. [Patrick et al (2008)]

Another solution is to introduce handoff servers to realized vertical handoff. In USHA [Chen et al (2005)] system, all mobile hosts connect to the Internet with the help of a handoff server, which is equipped with multiple network interfaces with heterogeneous physical properties. Hence, mobile hosts can communicate with the handoff server in various physical connections. The IP tunneling technique (IP encapsulation) is used in USHA with the handoff server functioning as one end and the mobile host as the other. Upper layer communications are bounded to a virtual interface - the tunnel interface, instead of physical interfaces. All data packets are transmitted through this IP tunnel. When the handoff event occurs, the underlying physical connection of the virtual tunnel is automatically switched to the new physical interface.

5. Proposed Architecture

Across heterogeneous networks, when a user is in roaming state and still wishes to maintain all active sessions, an automatic handoff mechanism is needed to change the communication flow and the physical radio interface in use. Moreover, the handoff mechanism should be user unaware, so that the handoff process can be handled smoothly.

Figure [1] provides a high-level view of the Linux network stack. The upper layer is the application, which defines the users/clients of the network stack. The next layer is the transport layer, responsible for process-to-process communication. Below this is the network layer, which is responsible for routing the packets to their destinations. At the bottom of the stack is the link layer. The link layer refers to the device drivers providing access to the various physical layer mediums, such as Ethernet devices or WLAN devices etc [Tim Jones (2007)]. Now, depending on the active device drivers the link layer chooses the network card through which packets have to be transmitted. This card is bound till the session is completed. The problem we face here is that during the vertical handoff there needs to be a switching between different NIC cards depending on the availability of the active cards. Currently the Linux network stack doesn't have a support for vertical handoff.

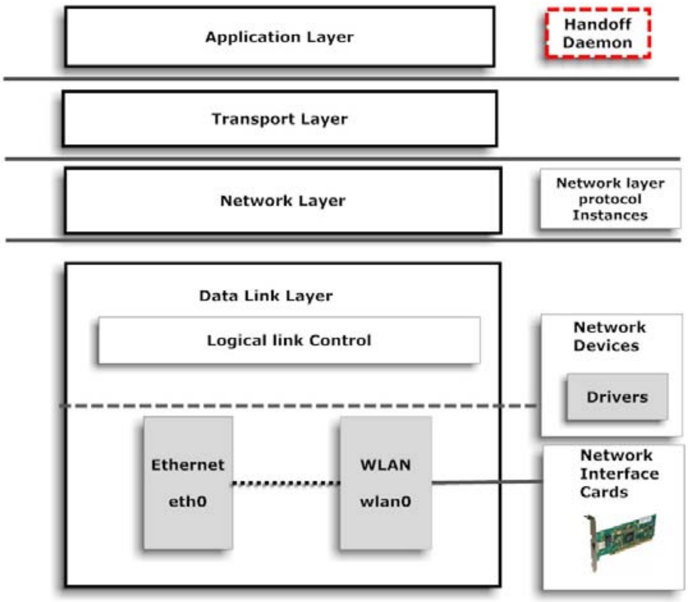

Fig 1: Proposed Architecture

Our proposed solution is development of Linux daemon to achieve vertical handoff for seamless multimedia communication over heterogeneous networks. We develop a daemon that accomplishes the handoff process between various radio interfaces supporting IP. This runs at the user level of the Linux operating system. The developed daemon always polls for the interfaces and checks for the interfaces that are alive. It also updates the network layer of the TCP/IP stack to maintain the session alive between the mobile station and the server. This architecture is well suited for all the multimedia networking applications running.

6. Implementation

This section details the complete implementation of the handoff daemon and the hardware used. The basic experimental setup consists of a:

- Sony VAIO laptop running Ubuntu Linux 9.04
  - Kernel version 2.6.32
  - Supporting 2 Interfaces
    - Ethernet (eth0)
    - WLAN (wlan0)
- Linksys WRT54G2 Accesspoint
- Server running Ubuntu Linux 9.04
- ACTi Camera

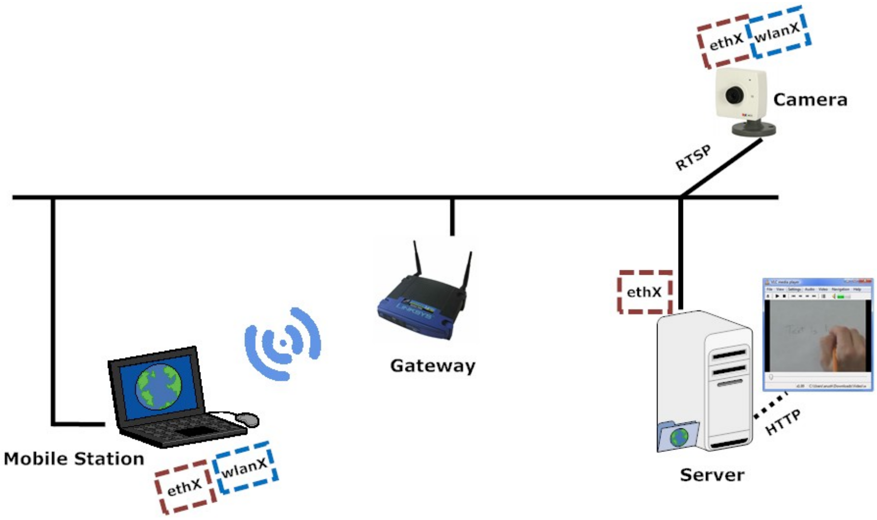

Fig 2: Experimental Setup

As shown in Fig [2] the whole setup is connected to LAN. The initial step is to validate and configure the network cards by using ifconfig. Once the configuration is done, then the next step is check the communication between the mobile station and the server for both the interfaces i.e. Ethernet & WLAN. When both the interfaces are alive, Linux kernel chooses one of the interfaces to transmit packets to the server. But when the chosen interface is plugged out, Linux kernel doesn't shift to the other interface for communication with the server. It immediately stops the session and waits for the chosen interface to get plugged in. For example if ethX is chosen as interface to send packets and is plugged out, Linux kernel doesn't have the mechanism to shift WLAN X to continue communication.

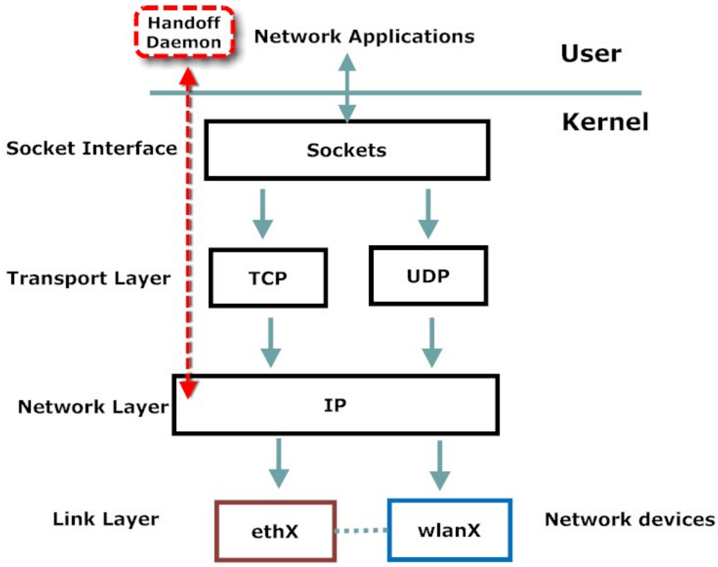

Fig 3: Daemon Implementation

To solve this problem, we developed and implemented a vertical handoff daemon. This daemon completely runs at the user space of the Linux operation system. The implementation details of the daemon are given below.

#!/bin/sh

VIRTUALIP="192.168.51.70"

check()

```
{
    ETH0=`cat /sys/class/net/eth0/operstate`
    WLAN0=`cat /sys/class/net/wlan0/operstate`
    ETH_IP=`ifconfig  | grep -r eth0 -A 1 -m 1  | grep "inet addr" | cut -d : -f 2 | cut -d ' ' -f 1`
    WLAN_IP=`ifconfig  | grep -r wlan0 -A 1 -m 1 | grep "inet addr" | cut -d : -f 2 | cut -d ' ' -f 1`

    if [   "$ETH0" = "up"  -a  "$ETH_IP" = "$VIRTUALIP"  ];     then
            echo "Eth0: $ETH0  IP: $ETH_IP"
    elif [ "$ETH0" = "down"  ]; then
            `sudo ifconfig eth0 up`
    else
            `sudo ifconfig eth0 $VIRTUALIP up`
    fi


    if [  "$WLAN0" = "up" -a  "$WLAN_IP" = "$VIRTUALIP"  ]; then
            echo "Wlan0 $WLAN0 IP: $WLAN_IP"
    elif [ "$WLAN0" = "down" ]; then
            `sudo ifconfig wlan0 up`
    else
            `sudo ifconfig wlan0 $VIRTUALIP up`
    fi

}
```

Initially according to our network, we set a virtual IP 192.168.51.70. Then our daemon continuously polls the status of eth0 interface at /sys/class/net/eth0/operstate. Simultaneously it also checks the status of wlan0 at /sys/class/net/wlan0/operstate. When the status of eth0 is up, it will be immediately assigned an IP 192.168.51.70 and packets are transmitted. But when the status shows down, it immediately assigns the same IP 192.168.51.70 to wlan0 interface and sends the packets to the server. In this way, switching is done between eth0 and wlan0 to keep the session alive and achieve communication between heterogeneous networks

7. Results

We have tested our daemon functionality for all the 3 forms of media i.e. data, audio & video. Open source Wireshark tool is used to validate the switching between the interfaces. The MAC ID of eth0 & wlan0 of mobile station are 00:24:bc: 43:1d:0c and 00:26:5c:f5: 39:89 respectively

7.1 For Data:

We used Secured Copy (SCP), Secured Shell (SSH) and our in house socket applications to check the functionality of our developed daemon.

7.1.1 Scenario 1: Switching from eth0 to wlan0

In this scenario, initially our mobile station will be connected to both eth0 and wlan0. Now, when we start our network application, packets flow from eth0 interface. Simultaneously we run our daemon in the user space. It will be continuously polling for the interfaces.

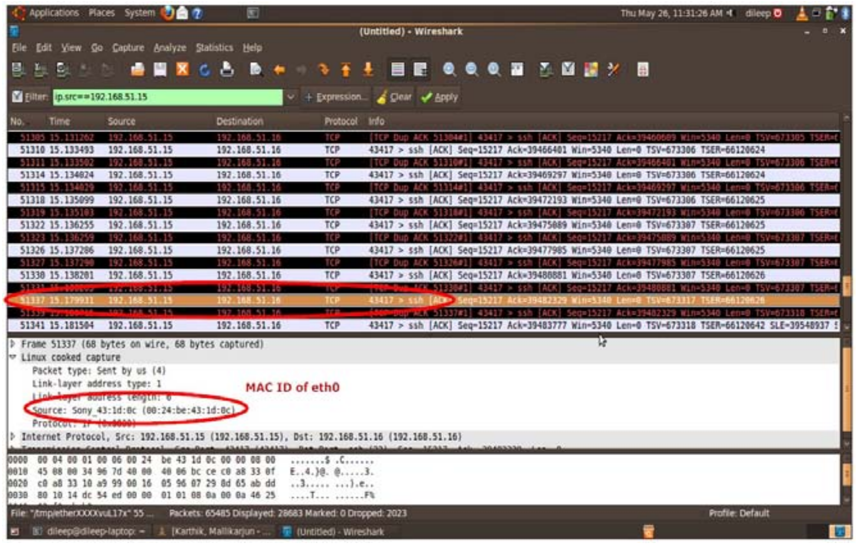

Fig 4: Communication with eth0 interface (data)

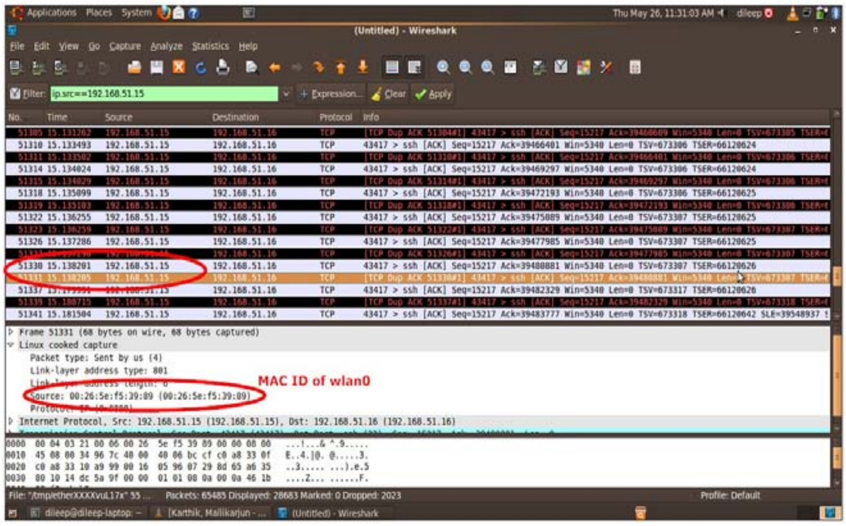

Fig 5: Switching from eth0 to wlan0 interface (data)

Once, when the eth0 interface is plugged off, immediately it switches to wlan0 to keep the session alive. The experimental results show that the switching between eth0 and wlan0 is around ~0.203 seconds

7.1.2 Scenario 2: Switching from wlan0 to eth0

This is in continuation to the above, the communication between mobile station and server happens with wlan0 interface. But, once when the eth0 interface is up, immediately the network interface is shifted to eth0 making the session alive. The experimental results show that the switching from wlan0 to eth0 is around ~0.047seconds

7.2 For Multimedia:

We tested our developed Linux daemon on both HTTP and RTCP based applications

7.2.1 HTTP Application:

To demonstrate the HTTP application we used VLC player as a medium to stream recorded video.
Server:
vlc -vvv /home/Movies --sout '#standard{access=http,mux=ogg,dst=192.168.51.16:1234}

Client:
vlc http://192.168.51.16:1234

Initially, streaming was done on eth0 network interface. When the Ethernet cable was plugged out, it took around ~ 3 seconds to shift to wlan0 and continue the session.

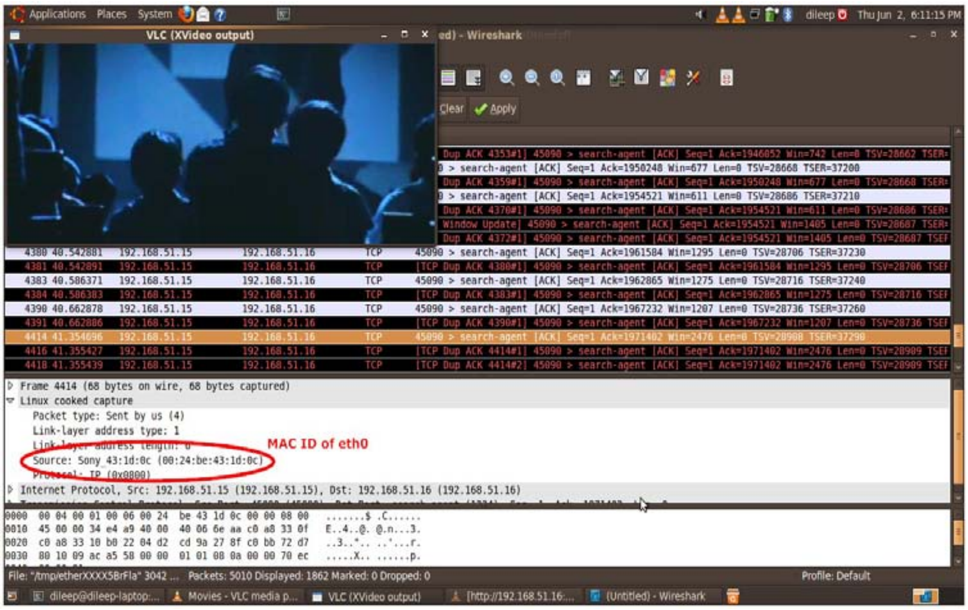

Fig 6: Communication with eth0 (HTTP)

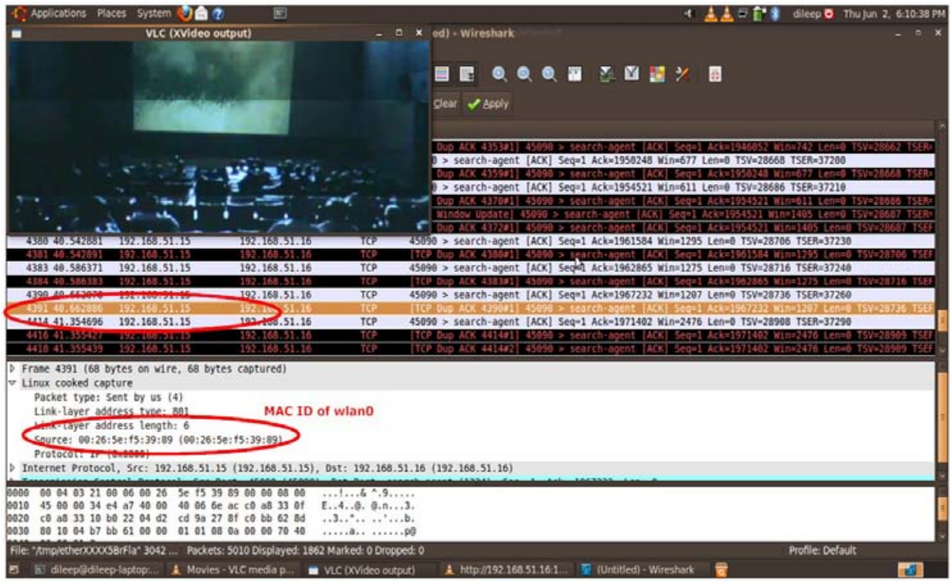

Fig 7: Switching from eth0 to wlan0 (HTTP)

Similarly, we tested the case of switching from wlan0 to eth0 running the HTTP application.

7.2.2 RTCP Application:

To demonstrate this application we used an ACTi camera and VLC player.

client:
vlc rtsp://user:user123@192.168.51.92:554/axis-media/media.amp

Once the camera is started, our mobile station receives packets from the eth0 interface.

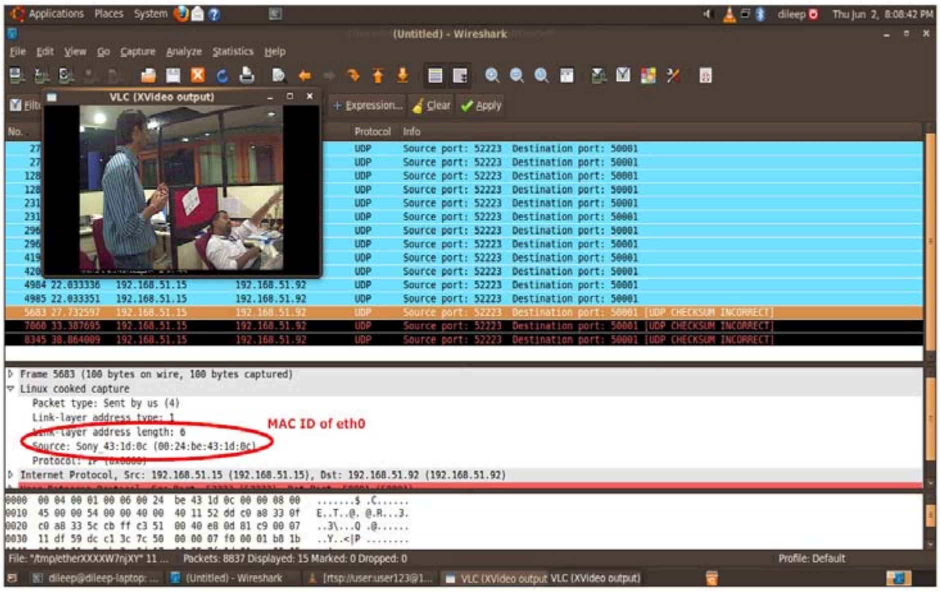

Fig 8: Communication with eth0 (RTCP)

Now when we plug out the cable, it immediately shifts to wireless interface within a span of around ~ 3 seconds

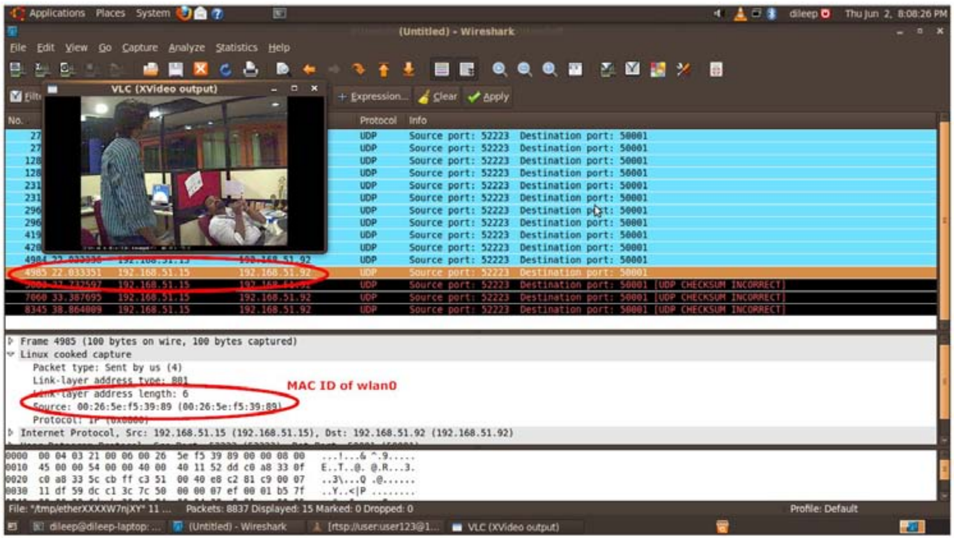

Fig 9: Switching from eth0 to wlan0 (RTCP)

Similarly, we tested the case of switching from wlan0 to eth0 running the RTCP application.

Table [1] shows the complete average time taken for vertical handoff between both the interfaces for data, Multimedia (HTTP) & Multimedia (RTCP) and total percentage of the CPU space taken by our developed Linux Demon.

Table1: Handoff Average Time & % CPU used

Medium	Handoff	Average Time	%CPU
Data	eth0 to wlan0	~0.203 seconds	1%
	wlan0 to eth0	~0.047 seconds	1%
Multimedia (HTTP)	eth0 to wlan0	~3.0 seconds	1%
	wlan0 to eth0	~1.0 seconds	1%
Multimedia (RTCP)	eth0 to wlan0	~3.0 seconds	1%
	wlan0 to eth0	~1.0 seconds	1%

8. Conclusion & Future Work

The proposed vertical handoff daemon was implemented and tested for multimedia networking applications successfully on Linux Operating System. It runs directly at the client devices and there is no need of any handoff agents or servers. This daemon is completely implemented at the user level of the Linux operating system. This architecture works for any network interface supporting IP. Currently, both the interfaces are occupying the bandwidth and there is redundancy in packets sent. In our future work, we will address the bandwidth, redundancy and dynamic address problem faced during vertical handoff process. A virtual interface layer will be introduced between the network layer & the link layer and finally integrate into the existing Linux kernel.

9. References

[1] Nikolaou, N.A., Vaxevanakis, K.G., Maniatis, S.I., and Venieris, I..S., Zervos, N.A. (2002): Wireless Convergence Architecture: A Case Study Using GSM and Wireless LAN. Mobile Networks and Applications, pp. 259–267
[2] Wei-Cheng Xiao, Shih-Hsuan Tang, Ling-Jyh Chen, and Cheng-Fu Chou, (2007): A Novel Seamless Vertical Handoff Solution, IEEE Consumer Electronics
[3] Hakon Eyde Kjuus (2007): Session Continuity in Heterogeneous Networks: A SIP-based Proactive Handover Scheme, University of Oslo
[4] Selim İckin (2010): Implementation Of Measurement Module For Seamless Vertical Handover, Blekinge Institute of Technology,
[5] Patrick Stuedi and Gustavo Alonso (2008): Transparent Heterogeneous Mobile Ad Hoc Networks, Swiss Federal Institute of Technology (ETHZ)
[6] L. J. Chen, Tony Sun, and Mario Gerla. (2005): USHA: A Practical Vertical Handoff Solution. MSAN
[7] M. Tim Jones (2007): Anatomy of the Linux networking stack from sockets to device drivers, Emulex Corp.
[8] Ashwin Kumar Chimata (2005): Path Of A Packet In The Linux Kernel Stack, University of Kansas
[9] Mahesh K. Subramanian (2005): A Kernel-based solution to seamless mobility in wireless networks http://www.*technology.asu.edu/files/documents/.../MaheshFinalReport_V6.pdf*
[10] Mohit Malhotra, Pramod P. J and S. V Srikanth (2011): Integration of IMS and 802.21 in a Heterogeneous Environment: An empirical analysis, IEEE ICDC
[11] J. Ylitalo et al. (2003): Dynamic Network Interface Selection in Multihomed Hosts, Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03), pp. 315-324.
[12] Seok Joo Koh, Sang Wook Kim (2005): mSCTP for Vertical Handover Between Heterogeneous Networks. Human.Society@Internet, pp.28-36
[13] Maria Rosa Frias Gonzalez (2009): Analysis of Interworking Functions for Heterogeneous Networks, Thesis, Aachen
[14] WireShark monitoring tool, Available from *www.wireshark.org*
[15] Linux Kernel Archives, Available from *http://www.kernel.org/*
[16] Ubuntu 9.04 Linux, Available from *http://www.ubuntu.com/*
[17] VLC player, Available from *http://www.videolan.org/**vlc**/*
[18] ACTi Camera, Available from *http://www.acti.com*