

# Vertical Handover

## A Transport Layer Approach

Eric Beda, Neco Ventura

Department of Electrical Engineering, University of Cape Town, Private Bag 7701

Rondebosch, Cape Town, South Africa

Tel: +27 +21 650 2813 Fax: +27 +21 650 3465

{ebeda, neco}@crg.ee.uct.ac.za

### Abstract

**In this paper we scrutinize the applicability of network layered mobility schemes, in particular Mobile IP, within a heterogeneous environment. We argue that a Transport Layer solution is better suited to support vertical handover mechanisms, because not only will it reduce the overall complexity of the handover process but also it will permit applications to take advantage of influencing handover management. We present a concept of Connection Identification that may be used to enhance packet multiplexing and de-multiplexing at the transport layer by eliminating network layer dependence of managing connections to support address migration and hence handover.**

### I. INTRODUCTION

One of the visions of future mobile communication is that of a multi-access environment in which terminals can be attached to several different overlaid access networks simultaneously. In a multi-access scenario, it would be possible to select network interfaces for separate connections. Users will have access to several different networked devices. If this becomes a reality, future services will emerge where it would be possible to hand over sessions between different terminals. The basic requirement for a mobile host is the ability to change its point of attachment to the network without losing its connection [1] [2]. The major challenge in providing a total host mobility solution is Interface Management, Location Management and Address Migration.

In internet environments, when a host moves and attaches itself to another network, it needs to obtain a new IP address. With this change of IP address, all existing TCP/IP connections to the mobile host terminates, as the IP routing mechanism cannot deliver the data to the correct end-point. There are many ongoing researches that address this mobility problem; solutions may be categorized by the protocol stack layer at which the solution is implemented. Generally speaking, mobility management can be better optimized the lower the protocol stack. However, the lower the layer, the more specialization is required, with the ensuing increases in complexity and limitations in scope. Therefore, there is a trade-off between optimization, complexity, and functionality that has to be considered when deploying terminal mobility [8]. As the percentage of real-time applications traffic over wireless networks keeps growing, the deficiencies of the network layer based Mobile IP in terms of high latency and

packet loss becomes more obvious. The question that naturally arises is: Can we find an alternative approach to network layer-based solution for mobility support? Since most of the applications in the Internet are end-to-end, a transport layer based mobility solution would be a natural candidate for an alternative approach. The transport layer is the lowest layer to support end-to-end services. By incorporating handover capabilities in terms of Address independence and transparency will permit applications to take advantage of influencing handover management; the advantages of this approach include:

- Only the end points would participate in the handover process.
- It can support concurrent usage of any type of access routers.
- Additional network components or modifications of intermediate routers are not required.

We believe that by introducing a concept of identifying all connections sessions independent of network layer attributes (IP addresses) will permit end hosts to move freely between networks of different access technologies seamlessly and hence cater for vertical handover. This approach will require significant changes to end systems. However, it is noted that software updates are necessary in any case, and this is an ongoing process. If IPv6 is to be implemented or if RSIP (Real Specific IP) [20][21] is to be used, end hosts would have to be updated. Similarly, if QoS support will be offered in access networks, end hosts will have to be upgraded with QoS capability [8].

The contribution of this paper is therefore as follows. In section II we examine current solutions to host mobility in heterogeneous networks. Section III provides an introduction on how TCP/IP multiplexes packets at end systems and hence why the current implementation cannot support vertical handover more importantly we illustrate the suitability of a Transport Layer solution for host mobility. Section IV introduces the concept of a connection identifier and describes XDP (extended datagram protocol) a protocol designed using our concept and which was used in our study. The results and analysis are then presented. We conclude in section V.

### II. RELATED WORK

#### A. Mobile IPv4

Mobile IP (MIP) is the standard proposed by IETF to handle mobility of Internet hosts for mobile data communication. For example, it enables a TCP connection to remain active when a mobile host moves from one point of attachment to another. Mobile IP is based on the concept of a Home Agent (HA) and Foreign Agent (FA) for routing packets from one point of

attachment to another. During the handover from the HA to the FA, a mobile host (MH) will need to register with the FA, wait for the allocation of channels, and update its location in the HA database. [3]. While MIP is a widely accepted concept in both research and industry, several problems exist when using MIP in a mobile computing environment. The most important issues of MIP identified to date include:

- High handover latency [4]: A MH needs to complete the following three steps before it re-establish the flow of data: (i) discovering of the new Care of Address (CoA), (ii) registering the new CoA with the HA, (iii) forwarding packets from the HA to the current CoA
- High packet loss rate [5]: During the HA registration period, some or all of the packets destined to the MH's old CoA will be lost since the old point of attachment can not communicate with the MH during this period, nor does it know the new point of attachment of the MH.
- Inefficient Routing [4]:

### B. Mobile IPv6

Mobile IPv6 addresses most of the disadvantages that were visible in its previous version. The main goal in its design considered the fact that future IP mobility frameworks need to consider the QoS constraints of active connections more closely when handling the usual requests of handover and that rerouting several optimizations can be done to improve the overall mobility performance.

The designers acknowledged that it is beneficial to forward packets directly between the corresponding and mobile hosts, as it enables the resultant path to be optimized for quality of service. However, it is unclear if and when all corresponding hosts would become mobility aware to provide this service. The major enhancements can be summarized as follows:

- Route Optimization
- Frequent handover and fast location updates
- Link layer assisted handover detection
- Tunneling across QoS domains.

Although MIPv6 promises route optimization that overcomes the problem of triangular routing as its ancestor experienced, the core of this protocol is based around depending on nodes that do not partake in the actual communication (Home Agent, Foreign Agent), most notably through the *Binding Updates* and *Foreign Agent Discovery*, these attributes contributes to handoff latencies due to the increase processing overheads at participating nodes, making deployment cumbersome and scalability dependent on willing participation of foreign entities. Our proposed solution only incorporates the end-points par-taking in the communication and is totally independent of foreign entities.

### C. MH-TCP

Christian Huitema [19] proposed to remove the fate sharing effect by allowing the set of addresses used by a TCP connection to change over time. He proposed to modify TCP by defining a new type of parameter, PCB-ID, to be used during the initial synchronization. This parameter identifies the "Protocol Control Block" associated to the TCP connection. When initiating a connection, the host attaches to the SYN packet the identifier of the local PCB. Hosts identify their local PCB and exchange "Extended TCP" packets, where a 32-bit PCB-ID replaces the 16-bit port number pair at the destination. This way, PCB location becomes independent of the IP addresses. The addresses in use are to be stored in this

PCB, and can change in due course of the connection. This approach supports the use of several addresses in parallel for the same connection, which is why the proposal is called "multi-homed TCP (MH-TCP)". Our concept however achieves PCB independence from IP addresses by the inclusion of a 64-bit connection identifier instead of a 32 bit PCB-ID. We believe that the idea of including kernel address space information (PCB-ID) within packet headers will make end systems more vulnerable to malicious attacks. Our philosophy is centralized on the idea that every connection should be identified independent of IP addresses; this identifier should only be used as an abstraction for IP addresses associated with the connection, it should not be used to identify intrinsic attributes of the connection, and this will cause another layer of dependency on this attribute forcing it to be static. We believe that the identifier should have provision to change during course of a session. This change should be done in a controlled manner private only to the end system associated with the session. Connections can be an association between more than two end systems to cater for bandwidth aggregation applications and future systems.

## III. BACKGROUND

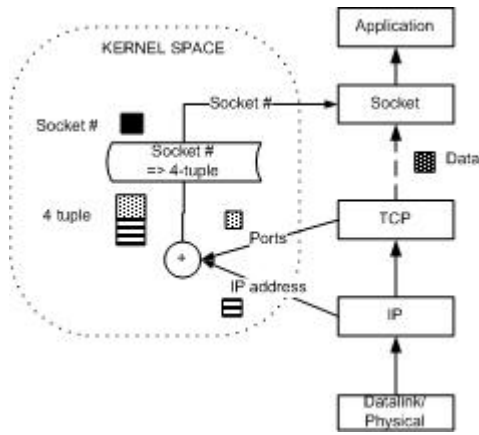
### A. Transport Layer

The transport layer allows multiple applications (connections) to communicate over a network, at the same time. To support multiple connections to a single host, TCP provides a set of ports within each host. This, along with the IP addresses of the source and destination (known as the 4-tuple association), makes a socket, and a pair of sockets uniquely identifies each connection, thus this uniqueness of a connection is core to TCP implementation. Next Generation Networks (NGN) promises an environment where network access terminals will be mobile and will support seamless movement between various access technologies. This will require mobile terminals to support address change without terminating the TCP session; the current 4-tuple association scheme used by TCP to uniquely identify a connection will not be suitable to support such mobility, since it will be necessary to change the IP parameters of the association during handover (vertical and horizontal). This research intends to solve this by proposing a new scheme to uniquely identify a connection independent of the IP substrate.

### B. Problem Statement

A TCP Protocol Control Block (PCB) contains state information for one endpoint of a given connection. A TCP demultiplexing (a.k.a. PCB-lookup) algorithm must find the PCB corresponding to the connection for each newly arrived TCP packet. The algorithm does this by mapping the packet's source and destination IP addresses and TCP ports to the proper PCB. Figure 1 briefly describes processes involved during TCP packet demultiplexing.

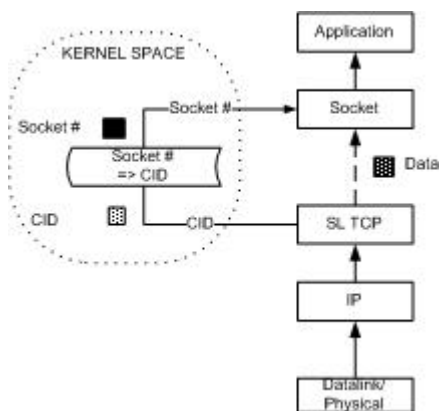
TCP stores the TCP Control Block (TCB) and PCB information of a particular connection in kernel space and the kernel module maintains the association between a socket number and four-tuple (TCP connection identifier). When an application sends data to a socket number, the kernel look ups the socket number in memory, and it obtains the four-tuple. It then sends the port addresses to the TCP layer and the IP addresses to the IP stack



**Figure 1 TCP Demultiplexing Process**

The process is done in the reverse order at the receiving end; this is described in figure 1. As shown the kernel maintains a mapping between the socket number and the four-tuple TCP connection identifier; hence the four-tuple has a one-to-one relationship with the socket number. If any parameter in the four-tuple changes, the association gets broken and so does the connection.

Originally IP addresses were good identifiers for hosts, because hosts were static and they could be identified on account of their topological location. However; the introduction of mobility and multi-homing has changed this situation. This implies that the binding between the four-tuple TCP connection identifier and socket structure defines an IP address dependency for TCP connections. Once a single address is changed within the four-tuple, the connection gets broken and the communication is temporarily terminated. If a TCP connection can be independent of its four-tuple identifier, then mobile terminals will be able to change their IP addresses (and hence network interfaces) at will without disrupting the TCP connection session. We believe that network layer independence from Transport Layer protocols can be achieved through the inclusion of a unique connection identifier (CID) within packet headers; this identifier will be sufficient to multiplex packets at the end hosts.



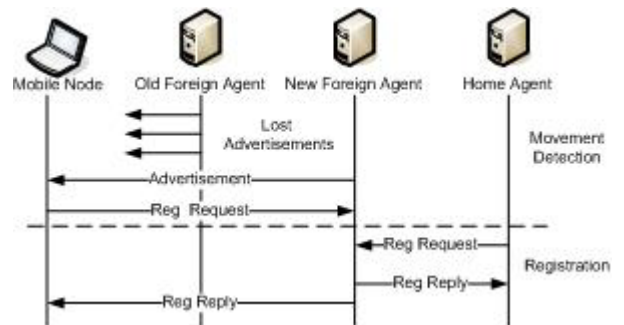
**Figure 2 Modified TCP Demultiplexing Process**

As figure 2 demonstrates, by introducing a connection identifier the dependency of the IP layer is zeroed out. This way whenever there is a change in four-tuple identifier, it will not affect the connection state; instead IP addresses are updated.

### C. Theoretical Evaluation

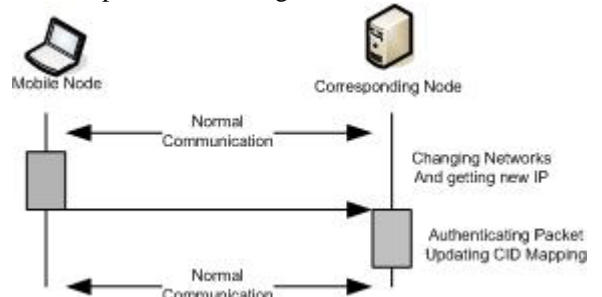
In brief MIP handover scheme is performed in two stages *Movement Detection* and *Registration* (Figure 3). Movement detection is based upon the Lifetime field within the main body of the ICMP Router Advertisement portion of the Agent Advertisement. Mobile nodes keep track of that Lifetime and if it expires, it sends an Agent Solicitation (asking for a new Agent Advertisement) since it presumes that it has been moved. Registration involves the mobile terminal informing its Home Agent (HA) of its new care of address (CoA). After these two stages packets may be successfully routed to the mobile terminal via its new CoA.

As shown in Figure 3 MIP handoff latencies are in the range of 6 RTT (Return Time Trip). This may be sufficient for some internet applications but as a requirement in next generation applications, handoff delays of this order are unacceptable, especially when the distance between the mobile host and home agent is large; By removing network layer dependency at the transport layer high handoff delays caused by packet indirection techniques (such as MIP) will be removed. As shown in figure 4 a transport layer solution promises to reduce handoff latencies close to degree zero (1 RTT). Delays only occur due to extra processing at the endpoints. These delays may be assumed negligible as modern processors have advanced and comparably very fast. Further delays will be contributed to by un-avoidable network characteristics and other deficiencies endured by packet switching networks in wireless environments.



**Figure 3 Mobile IP handover message signaling**

Delays may also arise when the mobile host is acquiring a new IP address; these delays are also un-avoidable and depend on the mechanisms deployed in the new network environment. It may be high if a considerable amount of signaling needs to be performed to authenticate the mobile terminal within the new domain prior to obtaining an IP address.



**Figure 4 Handover message flow in the proposed solution**

If IPv6 auto-configuration is implemented, these delays may be assumed to be negligible. IPv6 auto-configuration

mechanism promise to be the de facto method for acquiring IP address in future networks.

The proposed approach centralizes the intelligence of connection identification at the end hosts without any assistance from the network layer to maintain the connection other than using it as a tool to provide connectivity.

#### IV. OUR PHILOSOPHY

##### A. Overview

The mobility scheme proposed in this paper has been hugely influenced by the end to end argument as indicated in [13]. *“Certain required end- to-end functions can only be performed correctly by the end-systems themselves. A specific case is that any network, however carefully designed, will be subject to failures of transmission at some statistically determined rate. The best way to cope with this is to accept it, and give responsibility for the integrity of communication to the end systems.”* In light of this argument the maintenance of connections during normal TCP sessions should be assigned to the transport layer since it is the layer where all connection states are maintained. Current TCP connections are identified by pairs of host addresses and port numbers. This introduces dependency between the connection and these values. There are at least three cases where this dependency is unduly harmful:

- When the host moves to a new location and is assigned a new address,
- When the interface used by a multi-homed host to initiate the connection is temporarily disconnected
- When the host’s network is renumbered, e.g. after changing internet service provider.

This makes the endpoints prone or reliant on the network layer in maintaining the connection but as stipulated in [13] the tool for connectivity is IP and the intelligence behind it should be end to end rather than hidden in the network.

In contrast to other proposed transport solutions that choose to preserve TCP demultiplexing mechanism and support address migration functionalities by either maintaining multiple TCB states during address migration (introducing data redundancy) [15] or by overwriting the old TCB state right after a new one is created [7] (introducing packet losses); our solution supports address independence by introduction of a unique connection identifier (CID) that is maintained within the packet header so as to abstract the four-tuple used in traditional TCP demultiplexing; hence removing layer 3 dependency. Our approach requires two extensions to TCP connection-orientation methodology: the definition of a Connection Identifier parameter during client’s communication initialization and the carrying of this parameter in the data packets. There are three key concepts considered in the design of this approach; there combination can provide mobile terminals with explicit control to their mobility mode. The concepts are Addressing, Location Management and Connection Management.

##### 1) Addressing

IP addresses serves as a routing locator, reflecting as the addressee’ point of attachment in the network topology, this enables address aggregation based on address prefixes and allows for the routing to scale well [7]. We separate issues of obtaining an IP address from locating and seamless

communicating with mobile hosts; any mechanism for obtaining an address may be used, such as Dynamic Host Configuration Protocol (DHCP) and/or Auto configuration Protocol. An IP address is used entirely for routing purposes only and not for identifying connections.

##### 2) Locating Mobile Host

As a mobile host changes its network attachment it updates its IP address. There are two states where it can communicate with a corresponding host; either as a *Client* or as a *Server*. In case the mobile host is always a client (which is not an uncommon thing), nothing needs to be changed. The mobile host will be able to initiate connections with its new IP address and continue other sessions. The responsibility lies with the corresponding host to update its destination address to maintain connections with the mobile host. To support mobile servers and applications where internet hosts actively originates from the mobile hosts, we use the Domain Name Service (DNS) system to provide a level of indirection between the host current location [7] and variable IP address. In brief when a client initiates communication with a server it queries its DNS server by issuing it a ‘string’ that represents the server and receives an IP address of the server and with this it can initiate the connection. When a mobile server changes its attachment point, it must detect this (by using a user level daemon) and change the hostname-to-address (“A-record”) mapping in the DNS. This can be accomplished, by using the well-understood and widely available secure DNS update protocol [12].

##### 3) Connection Migration

When a communicating entity (mobile node) initiates a new connection, the session gets assigned a Connection Identifier (CID) parameter. The CID is a number with the purpose of uniquely identifying a communication session. Packets are multiplexed and demultiplexed to applications by using the CID instead of the TCP 4-tuple association. When a mobile host changes its IP address it resends the respective last packet it sent out to all its corresponding hosts (CN) in effect the CN updates destination address for that connection and acknowledges the change. It is thus imperative to define the source as "the address at which the source would like to receive replies to the message." This way mobile host can change addresses without compromising the connection session.

#### V. EVALUATION PLATFORM, EXPERIMENTATION AND RESULTS

In this section, we briefly describe XDP (Extended Datagram Protocol) the end-system mobility architecture used to illustrate CID concept in our evaluation platform.

##### 1) Transport Layer

The user datagram protocol (UDP) will act as a delivery service for our data. Although it could have been implemented directly on top of IP, UDP relieves the burden of a few difficult low level operations from our approach.

- UDP’s highly optimized payload checksum will be used to ensure validity of datagram’s received.
- UDP provides the port multiplexing feature, essential if a single server is to handle multiple connections.

The implementation uses the context information provided by underlying UDP to achieve flow control and packet retransmission. The UDP **data** is encapsulated with a Connection Identification Parameter (CID) to uniquely identify connection sessions. Each host maintains a mapping between the CID and the source and destination ports and addresses. XDP is implemented above UDP (figure 4) as a separate process providing reliable data transmission, data multiplexing and contains vertical handover intelligence.

### 2) Connection Initialization

An application creates a XDP socket and issues a connection request to a remote application. A CID is issued during socket creation and is then mapped to the TCP-4 tuple locally. XDP on the remote host picks up the connection request and also maps the CID to the 4 tuple locally and acknowledges the connection. XDP attaches the destination IP address acquired from the CID mapping to all outgoing packets and hands all routing control to the IP layer.

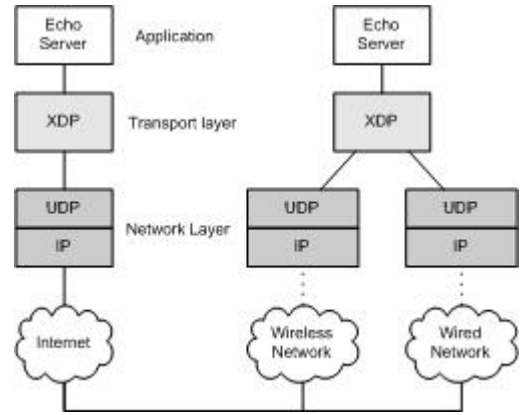
### 3) Incoming Data

An entity that has learned the CID of its partner can start sending XDP packets. XDP packets are regular UDP packets with encapsulated data section to accommodate CID parameter. Ideally XDP packet should have a format identical to normal TCP packets, except for the replacement of the port pair by a CID and be implemented directly above IP having its own header and IP protocol type (to be defined by IANA). XDP on each host maintains a local mapping between CID and the TCP four tuple. UDP receives packets from IP layer and delivers them to XDP. XDP then performs the following tasks in succession:

- Extracts packets CID.
- Extracts the local 4 tuple from the CID – 4 tuple association.
- Updates destination IP address within the 4 tuple if the corresponding hosts IP address has changed.
- Passes data to the application via its port numbers.

### 4) Experimental Setup

The architecture of the evaluation platform is shown in figure 5; the system consists of an echo server that sits on the internet and a mobile node running an echo client while switching between WLAN and LAN network interfaces. Both machines are equipped with an implementation of XDP on top of UDP/IP. Here XDP acts as a transport layer and UDP/IP as the network layer. The MN has two Network Interface Cards (NIC), the LAN interface is configured to have a static address while the WLAN interface obtains address dynamically using DHCP. Thus the LAN acts as the *always on* network.



**Figure 5 Experimental Setup**

### 5) Results

The experiment was run for an exchange of 400 data packets between the client and server. Vertical Handover was forced after 5 successive delivered packets at the server. Time at which each packet reached the server was logged onto a file indicating source address of the data packet. The time was averaged and is tabulated as shown in table 1 below.

	Average Time in Sec
RTT LAN	0.208068
RTT WLAN	0.339735
Handover LAN-WLAN	4.49723
Handover WLAN-LAN	0.181597

**Table 1 Averaged experimental results**

As expected the results show that Vertical Handover achieved in this way produced delays of order close to the return time trip. The handoff delay inconsistency shown in LAN-WLAN which is larger than the WLAN RTT is due to delays induced by DHCP while acquiring and registering a new IP address.

## VI. CONCLUSION

We illustrated a transport layer approach to support vertical handover in heterogeneous environments. It is a true end-to-end handover scheme and does not require any intermediaries (such as home agents) to participate in the flow control, and connectivity. Changes are confined to the end points, which could be applied to as a module and/or patch to terminals with multiple access technologies. There is very little overhead; handoff latencies are estimated to be approximately equal to the RTT (Return Time Trip) of participating packets hence catering for real time services. In conclusion, we would like to point out that true end-to-end techniques will become indispensable when IPSEC or other security mechanisms are employed to encrypt the IP payloads [11]. Drastic changes *will* be required in the future since the internet was not conceived to support mobility, security, Quality of Service (QoS), etc. Trying to incorporate these attributes is therefore a continual “retrofit” job which will perhaps not fix everything. However, as with any retrofit, backward compatibility and inter-operability with existing infrastructures are of utmost importance and Session-TCP technique satisfies these criteria.

## VII. REFERENCES

- [1] Charles E. Perkins, "RFC320: IP Mobility support for IPv4," Jan. 2002.
- [2] David B. Johnson and Charles E. Perkins, "Mobility Support in IPv6", Internet-Draft, Nov.2001, Work in progress.
- [3] Shaojian Fu, M. ATiquzzaman et al, "TraSH: A Transport Layer Seamless Handover for Mobile Networks" Jan 2004.
- [4] C. E. Perkins, "Mobile Networking Through Mobile IP," IEEE Internet Computing, Vol. 2, no. 1, pp. 58-69, Jan/Feb 1998.
- [5] "Low Latency handoffs in Mobile IPv6." IETF DRAFT, draft-ietf-mobileip-lowlatency- handoffs-v4-07.txt, Oct 2003
- [6] J. Ylitalo, T. Jokikyyny, "Dynamic Network Interface Selection in Multi-homed Mobile Hosts" 2002 – IEEE Computer Society.
- [7] Alec C. Snoeren and Hari Balakrishnan, "An End-to-End Approach to Host Mobility" 2000, MIT Laboratory for Computer Science.
- [8] Borko Furth, Mohammad Ilyas, "Wireless Internet Handbook", CRC PRESS.
- [9] Stephen A. Thomas "IPng and TCP/IP Protocols – Implementing the Next Generation Internet"
- [10] Jennifer C. Hou, Guanghui He, "Coordinated Congestion Control", University of Illinois at Urbana-Champaign.
- [11] Tom Goff, James Moronski "Freeze-TCP: A true end-to-end TCP enhancement mechanism for mobile environments"
- [12] EASTLAKE, 3<sup>RD</sup>, D. E. Secure domain name system dynamic update. RFC 2137, IETF, Jan. 1998
- [13] B. Carpenter "Architecture Principles of the Internet" RFC 1958, IETF, June 1996.
- [14] TCP for transactions RFC 1379
- [15] Hung-Yun Hsieh, et al "A Receiver-Centric Transport Protocol for Mobile Hosts with Heterogeneous Wireless Interfaces"
- [16] Dr WJ Buchanan "TCP/Socket Programming" Dec 2003
- [17] Pekka Nikander, Jukka Ylitalo. "Integrating Security, Mobility, and Multi-homing in a HIP way" Network and Distributed Systems Security Symposium, 2003.
- [18] H. Saltzer, D. P. Reed, and D. Clark, "End-to-end arguments in System Design," *ACM Trans. Computing Systems*, vol. 2, no. ,1984.
- [19] C. Huitema, "Multi-homed TCP" Internet draft, May 1995
- [20] Network Working Group "Realm Specific IP: Protocol Specifications" RFC 3103
- [21] Network Working Group "Real Specific IP for End to End IPsec" RFC 3104.

## VIII. BIOGRAPHY

Eric Beda received his BSc in Electrical Engineering and Computer Science with honors at the University of Cape Town in 2003 and is currently doing an MSc in Electrical Engineering at University of Cape Town expected to complete in Dec 2005