

# **VIP: a Virtual Interface-based aPproach for vertical handover in single-subnet networks**

Wei-Cheng Xiao, Shih-Hsuan Tang, and Cheng-Fu Chou  
Department of Computer Science and Information Engineering  
National Taiwan University  
{b89099, r93057, ccf}@csie.ntu.edu.tw

Ling-Jyh Chen  
Institute of Information Science  
Academia Sinica  
ccljj@iis.sinica.edu.tw

## **Abstract**

*We propose a simple and practical solution, called the Virtual Interface-based aPproach (VIP), for seamless vertical handover in single-subnet networks. VIP combines a virtual interface and ARP update messages to achieve vertical handovers, and can run stand-alone without modification of existing network protocols or installation of additional servers. Therefore, it is ready for immediate real-world deployment. Using testbed experiments, we evaluate VIP by observing TCP (FTP downloading) behavior in various vertical handover scenarios. The results show that the proposed approach can successfully maintain an application's connectivity in all test cases. Furthermore, the results show that handover from a low capacity link to a high capacity link does not cause service latency; however, during handover from a high capacity link to a lower capacity link, non-negligible latency can not be alleviated unless early handover notification is provided.*

## **1. Introduction**

With the growing demands of mobile and wireless applications, the need for a simple yet efficient seamless handover solution that can transparently switch a connection across different network technologies (i.e., vertical handover) is becoming increasingly important. For instance, a user may have to switch an ongoing connection from Ethernet to IEEE 802.11b because he/she has roamed out of the cable range, or switch the connection from IEEE 802.11b to IEEE 802.11a because the former's channels suffer from serious interference. A seamless handover solution with

both low latency and low packet loss is essential for mobile users who wish to receive continuous, uninterrupted network service while switching from one network connection to another. Additionally, the handover solution should be network-layer-transparent and infrastructure-modification-free so that existing network applications can accommodate the changes wrought by the rapid evolution of wireless technology.

The goal of seamless handover is to provide continuous connections for end-to-end data transmissions in the event of any link outage or handover anomaly. This involves two critical design issues: low latency and low packet loss. Low latency requires that the switch to the new path must be completed almost instantaneously; that is, service interruption should be minimized to provide the illusion of continuous connectivity. In the event of an actual connection failure, the architecture must attempt to re-connect the client as soon as the service is restored. Furthermore, packet loss due to the connection switching must also be minimized.

Various seamless vertical handover solutions have been proposed in the last decade. Based on their implementations in the OSI 7 layer network model, the schemes can be grouped into three main categories: network layer approaches [6] [13], transport layer approaches [2] [12] [15] [18], and upper layer approaches [4] [7] [11] [14] [16]. All the schemes have two properties in common: (1) they support vertical handover across different network domains; and (2) they are not stand alone approaches (i.e., they have to either modify existing network protocols like TCP/IP, or otherwise install additional servers on the Internet); thus, deployment of these approaches is considered costly.

In this paper, we propose a Virtual Interface based aPproach (VIP) that not only satisfies the requirements for

seamless vertical handover, but is also simple, light-weight, and ready for real deployment immediately. Unlike previous handover schemes, *VIP concentrates on the emerging scenarios of single-subnet networks (e.g., home networking, campus networking, and office networking), and trades the support of handover across multiple network domains for simplicity in the approach's design.* More precisely, VIP cleverly combines the use of a virtual interface and ARP update messages to achieve vertical handover, and can run stand-alone without modification of existing network protocols/applications or installation of additional servers/services. Using testbed experiments, we evaluate VIP by observing TCP (FTP downloading) behavior in various vertical handover scenarios. The results show that VIP can successfully maintain an application's connectivity in all cases. Furthermore, the results show that handover from a low capacity link to a high capacity link does not cause service latency; however, during handover from a high capacity link to a lower capacity link, non-negligible latency can not be alleviated unless early handover notification is provided.

The rest of the paper is organized as follows. In Section 2, we summarize related work and recap the definition of vertical handover. Section 3 describes the enabling mechanisms behind VIP. Section 4 contains the experiment results for various vertical handover scenarios. Finally, in Section 5 we present our conclusions.

## 2. Background and Related Work

Vertical handover is the process of switching an ongoing network connection from one interface to another interface [17], as shown in Fig. 1. With the eventual unification of cellular and IP networks, providing seamless vertical handover support for mobile users has become increasingly important and desirable. The biggest challenge in designing a seamless vertical handover scheme lies in maintaining an established application session, such as a connection-oriented flow like TCP, while providing smooth (low loss) and fast (low delay) data transmission.

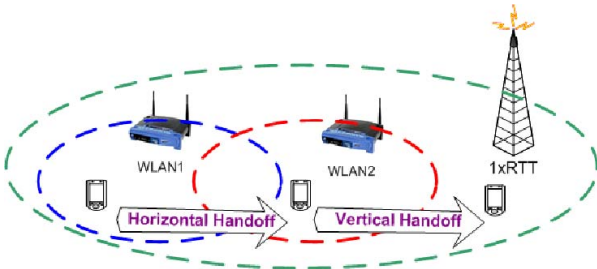


Figure 1. Horizontal and Vertical Handover

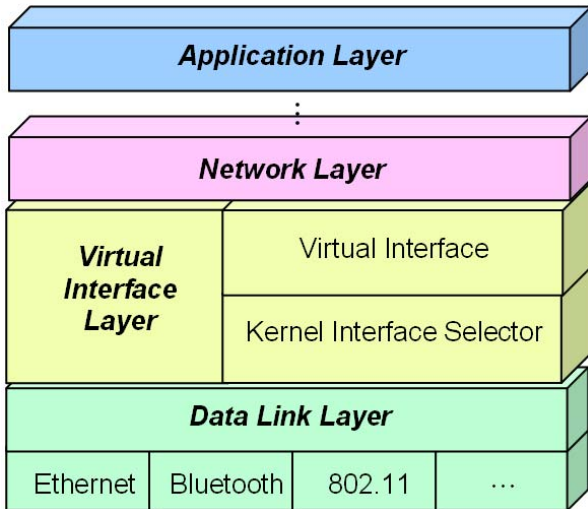
Mobile IP, a network layer solution for vertical handover scenarios, is the current IETF standard for Internet mobility support [13]. It ensures the delivery of packets to a mobile host by using a home address, instead of sending them directly via the mobile host's current Internet address (i.e., its care-of address). An address translator is deployed on the home agent, and an IP tunnel between the home agent and a foreign agent is carefully maintained; therefore, after handover to the foreign agent's network, all ongoing traffic destined for the home agent will be forwarded to the foreign agent (and then to the mobile host) through the IP tunnel. As a result, handover to the upper layer applications is transparent, and the connectivity can be maintained. IPv6 [6], the subsequent network layer protocol, provides native mobility support. By combining the home address and the care-of address into a 128-bit Internet address, IPv6 can easily support vertical handover scenarios without deploying foreign agents and address translators on the home agents.

Besides network layer solutions, several handover solutions use transport layer approaches [2] [12] [15] [18]. In [2], the authors proposed the use of Indirect TCP (I-TCP), which deals with handovers by splitting a connection into two segments: a fixed (wired) connection, and a mobile (wireless) connection. However, I-TCP is expensive to deploy and suffers from loss of end-to-end semantics. Meanwhile, in [12] [18], a multi-homing technique is used to support vertical handover. The deployment of the above solutions requires upgrading of both of the transport layer and the applications on mobile hosts and Internet servers. Consequently, the deployment cost is still too high. Another end-to-end approach supports host mobility by dynamically updating the Domain Name System (DNS) and adding a set of Migrate options to TCP [15]. However, this scheme not only requires mobile hosts to be updated, but also calls for all DNS servers worldwide to be upgraded, which is obviously impractical.

Additionally, several proxy-based solutions have been proposed to address network heterogeneity problems, such as mobility and vertical handover [4] [7] [11] [14] [16]. Using a split-connection proxy, the mobile host can maintain application sessions virtually, even in the presence of vertical handovers. Although the end-to-end semantics can be preserved, since the handover is transparent to the transport and application layers, such proxy-based solutions suffer from scalability problems due to the limited resources and computational capability of individual proxies.

## 3. Proposed Approach: VIP

In this section, we present our proposed vertical handover scheme, called VIP (a Virtual Interface-based approach). Unlike previous handover schemes, VIP is de-



**Figure 2. Basic system design of the OSI network architecture.**

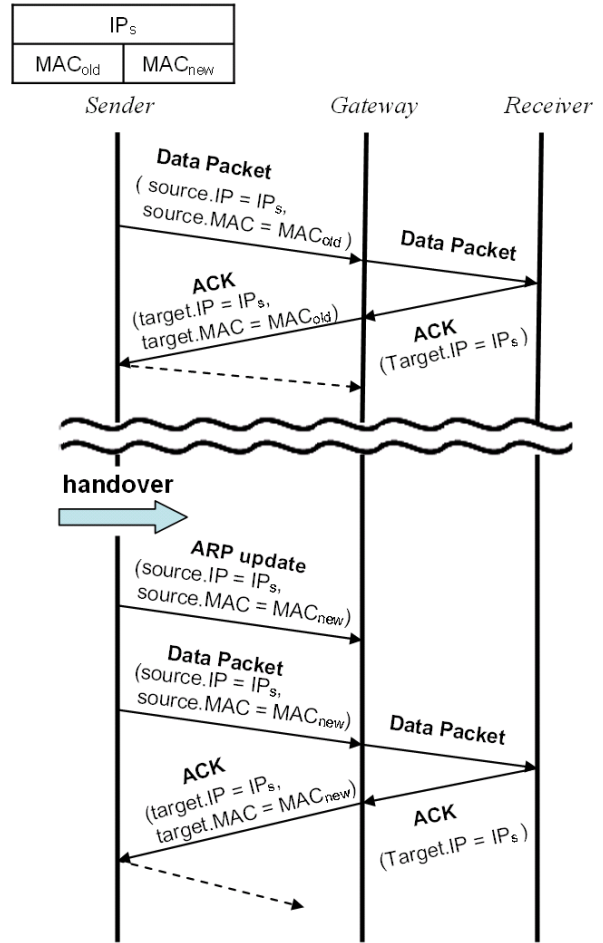
signed for the specialized and emerging scenarios of single-subnet networks (e.g., home networking, campus networking, and office networking), and trades the support of universal handovers across multiple network domains for simplicity in the approach’s design. As a result, VIP is simple and practical, and does not need to change upper-layer applications/protocols or deploy additional servers on the Internet. In the following, we discuss VIP in details.

### 3.1. Basic Ideas

As mentioned previously, VIP is designed for the specialized and increasingly popular scenarios where vertical handover takes place within the same subnet (e.g., a user may employ an Ethernet connection in his/her cubicle, but has to hand over to Wi-Fi when leaving for the conference room). It is not the goal of VIP to provide a universal solution for vertical handovers across the heterogeneous networks of different network domains.

In VIP, a *Virtual Interface Layer* (VIL) is inserted between the network layer and the data link layer, as illustrated in Fig. 2. The VIL is composed of two components, namely the *Virtual Interface* (VI) and the *Kernel Interface Selector* (KIS). It is the responsibility of KIS to determine the “best” physical network interface and link it to the virtual interface (VI). The decision about the “best” network interface (i.e., the one that can yield the best network performance at the time) is beyond the scope of this paper, but the issue has been addressed by several previous studies [3] [10] [19] [20].

Note that the VI is a dummy network interface that does



**Figure 3. When handover occurs, an ARP message is sent out to update the ARP table on the gateway.**

not exist physically. However, the VIP scheme assigns an IP address to the VI (instead of the selected physical network interface) and treats it as the default networking interface of the mobile host (i.e., the communications of the upper layer applications are bound to the VI, instead of to physical interfaces). Once a physical interface has been selected and linked (by KIS), the VIP scheme ensures that all outgoing traffic is sent from the VI, through the selected physical interface, and then to the network. The scheme also ensures that all incoming traffic is forwarded from the active (selected) physical interface, to the VI, and then to the system kernel. Since all network interfaces (including the VI and the physical ones) are within the same subnet, end-to-end connectivity can be easily maintained during vertical handovers. Next, we describe our implementation in details.

### 3.2. VIP Implementation

The proposed VIP scheme is implemented by dynamically invoking the ARP update messages that link the MAC address of a newly selected interface to the  $VI$  to complete a vertical handover. We denote the IP address of the  $VI$  as  $IP_s$ , the MAC address of the physical interface employed before the vertical handover as  $MAC_{old}$ , and the MAC address of the newly selected network interface for vertical handover as  $MAC_{new}$ . Fig. 3 illustrates the vertical handover process, where a mobile host (sender) communicates with a receiver on the Internet via a gateway. The vertical handover takes place on the first hop link (i.e., the link between the sender and the gateway).

As shown in Fig. 3, the mobile host (MH) uses the original physical interface to communicate with the receiver initially. In the packet header of each outgoing packet of the MH, the MAC address and IP address are set to  $MAC_{old}$  and  $IP_s$  respectively. Note that, while the  $IP_s$  is employed as an identity for Internet routing purposes,  $MAC_{old}$  is used for data transmission on the last hop (i.e., from the gateway to the sender). Therefore, the gateway host also maintains an ARP table to translate IP addresses into MAC addresses for last-mile connection. Of course, the ARP table has an entry with the  $IP_s$  mapped to  $MAC_{old}$  before the vertical handover occurred.

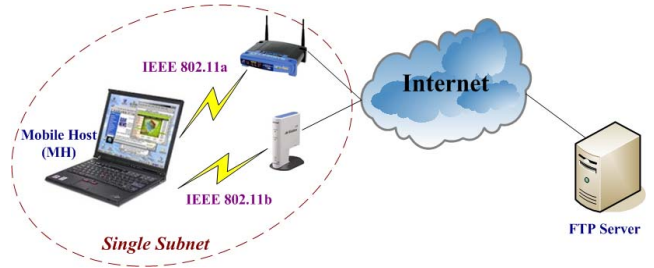
When a vertical handover is triggered<sup>1</sup>, the sender immediately transmits an ARP update message (i.e.,  $IP_s$  mapped to  $MAC_{new}$ ) to the gateway, which will then update its ARP table accordingly. As a result, all subsequent incoming packets (i.e., from the gateway to the sender) will be forwarded to the interface whose MAC address is the same as  $MAC_{new}$ . Thus, vertical handover (from the initial interface with  $MAC_{old}$  to the newly selected interface with  $MAC_{new}$ ) can be achieved seamlessly.

## 4. Evaluation

### 4.1. VIP Testbed

First, we describe the VIP testbed and experiments. The mobile host is a Linux based laptop (Debian Linux with kernel version 2.6.12.6) with an Intel Pentium M 1.73 GHz CPU and 512MB Memory, as shown in Fig. 4. The Internet gateways are set up using two wireless access points: a Linksys WAP55AG router, and a Buffalo WLA2-G54C

<sup>1</sup>As mentioned previously, we do not implement an intelligent handover manager [3] [10] [19] [20] since it is beyond the scope of this study. Instead, in the current work, all vertical handovers are triggered manually by simply writing commands to a pre-defined file in the /proc filesystem of the Linux system, which will promptly respond accordingly. The /proc filesystem can be thought as an interface for exchanging messages/commands between the user and the kernel spaces.



**Figure 4. Illustration of the VIP testbed. The MH is equipped with an IEEE 802.11a interface and an IEEE 802.11b interface, both of which operate within the same subnet.**

access point. In the experiment, the Linksys router is configured to serve as an IEEE 802.11a access point, while the Buffalo access point provides an IEEE 802.11b connection. Both the 802.11a and 802.11b networks are configured within the same subnet.

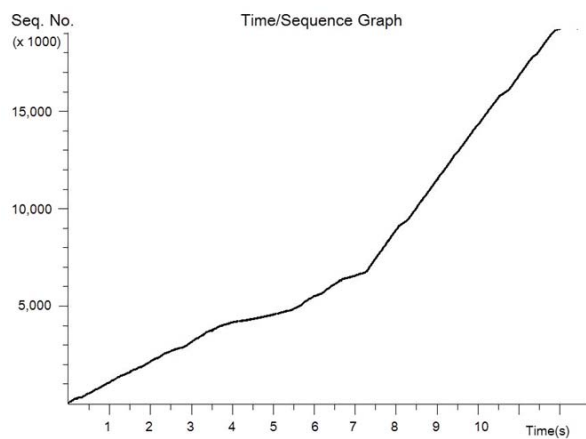
Specifically, the mobile host is equipped with a built-in Intel PRO/Wireless 2200BG NIC and a PCMCIA-based wireless NIC (which is a D-Link DWL-AG660 802.11 a/b/g Wireless NIC). The former is used to set up an IEEE 802.11b connection to the Buffalo access point, and the latter is used for the IEEE 802.11a connection to the Linksys router.

First, we evaluate vertical handover from the 802.11b connection to the 802.11a connection (i.e., LOW to HIGH), followed by handover from 802.11a to 802.11b (i.e., HIGH to LOW). In each experiment, the MH downloads a large file from a remote FTP server during the vertical handover. We run Ethereal [1] on the MH to capture all outgoing/incoming packets on the virtual interface for further analysis. The experiment results are presented below.

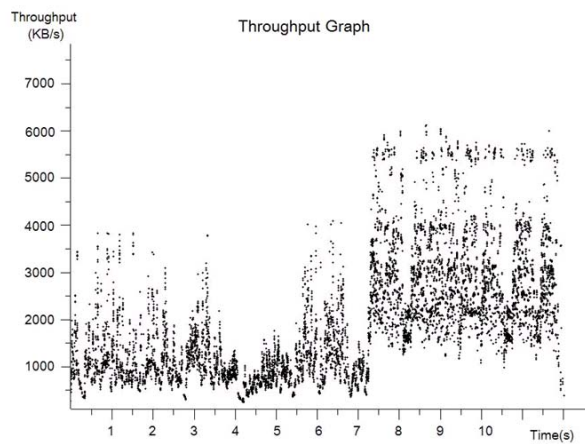
### 4.2. Experiment I: Vertical Handover from Low to High

In the first experiment, we perform vertical handovers from the low bandwidth link (i.e., IEEE 802.11b, 11Mbps mode) to the higher one (i.e., IEEE 802.11a, 54Mbps mode). The vertical handover is triggered manually at about 7.2 seconds. Fig. 5 depicts the TCP performance during the handover.

From Fig. 5, we observe that the average TCP throughput increases from 1,000 KBps to about 2,700 KBps after the LOW-to-HIGH vertical handover, and the sequence number curve of the TCP connection does not exhibit additional latency during the handover. Vertical handover to the upper layer FTP applications is seamless and transparent. Moreover, the TCP session can take advantage of the

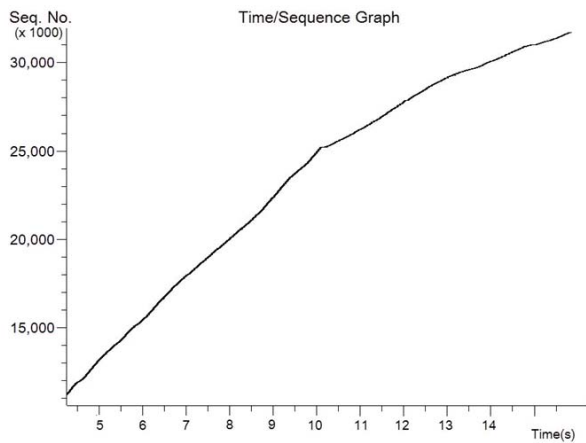


(a) TCP Sequence Number

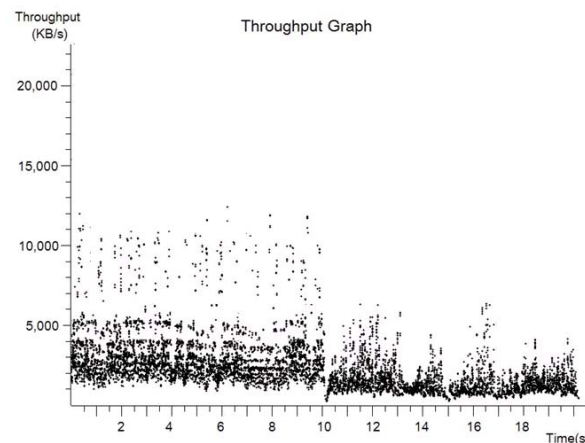


(b) FTP Throughput

**Figure 5. The changes in the TCP sequence number and FTP throughput before and after handover in the first experiment (802.11b to 802.11a).**



(a) TCP Sequence Number



(b) FTP Throughput

**Figure 6. The changes in the TCP sequence number and FTP throughput before and after handover in the second experiment (802.11a to 802.11b).**

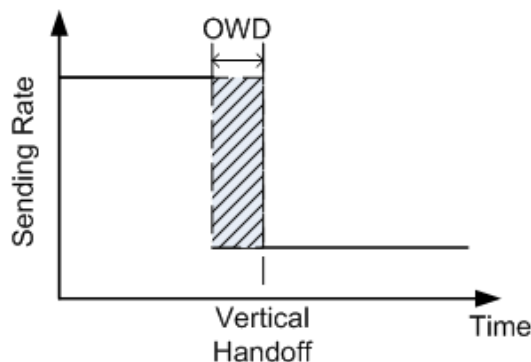
increased bandwidth after the vertical handover because of its AIMD-based congestion control algorithm.

#### 4.3. Experiment II: Vertical Handover from High to Low

In the second experiment, the vertical handover is from the high bandwidth link (i.e., IEEE 802.11a, 54Mbps mode) to the lower link (i.e., IEEE 802.11b, 11Mbps mode), and is triggered manually at about 10.0 seconds. Fig. 6 shows the TCP performance during vertical handover.

Fig. 6-a shows that the sequence number curve of the TCP connection increased monotonically during the transmission, but held for several microseconds when the HIGH-to-LOW vertical handover was triggered. Meanwhile, Fig. 6-b shows that the simultaneous TCP throughput dropped to almost zero when the vertical handover occurred, and then returned to about 1,000 KBps after a few microseconds. The results demonstrate that the TCP session can be maintained by the proposed VIP scheme during HIGH-to-LOW vertical handover; however, the ongoing TCP connection will be frozen for several microseconds. We repeated the experiment 10 times and found the “freeze period” was about 0.12ms in average.

The reason for the “freeze period” is because the TCP connection stored a few packets on the last hop link (i.e., between the MH and the IEEE 802.11a access point), and these outstanding packets are lost when HIGH-to-LOW vertical handover occurs, as shown in Fig. 7. The bulk packet loss causes contiguous TCP timeouts and thus degrades the throughput performance of FTP downloading. This problem can be resolved by employing specialized TCP variants (e.g., Freeze-TCP [8]), or by implementing other QoS enhancement schemes (e.g., Explicit Handover Notifications [9] or ERR+EHN [5]).



**Figure 7. The shaded region represents the amount of outstanding data that will be lost during HIGH-TO-LOW vertical handover.**

## 5. Conclusion

We have proposed a simple and practical vertical handover solution, called VIP, for single subnet scenarios. VIP is seamless and does not necessitate any modifications of the current network infrastructure. Using testbed experiments, we evaluated VIP in two home networking scenarios. The results show that our method can maintain an application’s connectivity (i.e., TCP session) without introducing additional latency during a vertical handover. We also discussed the QoS issue of TCP connection in HIGH-to-LOW vertical handover scenarios. In our ongoing work, we will investigate vertical handover scenarios, in which the link capacity and round-trip delay may change dramatically after handover, and design appropriate “adaptation” schemes so that applications can adapt more effectively in mobile computing environments.

## 6. Acknowledgements

This material is based upon work supported by the National Science Council under grant number NSC 95-2221-E-001-025.

## References

- [1] Ethereal: A network protocol analyzer. <http://www.ethereal.com/>.
- [2] A. Bakre and B. Badrinath. I-tcp: Indirect tcp for mobile hosts. In *IEEE ICDCS*, 1995.
- [3] L.-J. Chen, T. Sun, B. Chen, and M. Gerla. A smart decision model for vertical handoff. In *The 4th ANWIRE International Workshop on Wireless Internet and Reconfigurability*, 2004.
- [4] L.-J. Chen, T. Sun, and M. Gerla. Usha: A practical vertical handoff solution. In *IEEE MSAN*, 2005.
- [5] L.-J. Chen, G. Yang, T. Sun, M. Y. Sanadidi, and M. Gerla. Enhancing qos support for vertical handoffs using implicit/explicit handoff notifications. In *QShine*, 2005.
- [6] S. Deering and R. Hinden. Internet protocol, version 6 (ipv6) specification. Technical report, IETF RFC 2460, December 1998.
- [7] V. Ghini, G. Pau, M. Roccetti, and P. S. M. Gerla. Smart download on the go: A wireless internet application for music distribution over heterogeneous networks. In *IEEE ICC*, 2004.
- [8] T. Goff, J. Moronski, and D. S. Phatak. Freeze-tcp: A true end-to-end tcp enhancement mechanism for mobile environments. In *IEEE Infocom*, 2000.
- [9] A. Gurtov and J. Korhonen. Measurement and analysis of tcp-friendly rate control for vertical handovers. *ACM MCCR*, 2004.
- [10] B. Liang, A. H. Zahran, and A. O. Saleh. Application signal threshold adaptation for vertical handoff in heterogeneous wireless networks. In *IFIP Networking*, 2005.

- [11] D. A. Maltz and P. Bhagwat. Msocks: An architecture for transport layer mobility. In *IEEE Infocom*, pages 1037–1045, March 1998.
- [12] A. Matsumoto, M. Kozuka, K. Fujikawa, and Y. Okabe. Tcp multi-home options. Technical report, draft-arifumi-tcp-mh-00.txt, IETF Internet draft, October 2003.
- [13] C. Perkins. Ip mobility support for ipv4. Technical report, IETF RFC 3344, August 2002.
- [14] M. Schlager, B. Rathke, S. Bodenstein, and A. Wolisz. Advocating a remote socket architecture for internet access using wireless lans. *Journal of Mobile Networks and Applications*, 6:23–42, 2001.
- [15] A. Snoeren. *A Session-Based Approach to Internet Mobility*. PhD thesis, Massachusetts Institute of Technology, December 2002.
- [16] A. C. Snoeren and H. Balakrishnan. An end-to-end approach to host mobility. In *ACM MobiCom*, 2000.
- [17] M. Stemm and R. H. Katz. Vertical handoffs in wireless overlay networks. *ACM Mobile Networking (MONET)*, 1998.
- [18] R. Stewart, Q. Xie, K. Morneault, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson. Stream control transmission protocol. Technical report, IETF RFC 2960, October 2000.
- [19] H. Wang, R. H. Katz, and J. Giese. Policy-enabled handoffs across heterogeneous wireless networks. In *ACM WMCSA*, 1999.
- [20] F. Zhu and J. McNair. Optimization for vertical handoff decision algorithms. In *IEEE WCNC*, 2004.