Proceedings of the 5th WSEAS International Conference on Telecommunications and Informatics, Istanbul, Turkey, May 27-29, 2006 (pp70-75)

A L2 Transmission Feedback based buffering scheme for Mobile IP handovers in an IEEE 802.11 wireless network.

PETER DE CLEYN
University of Antwerp - IBBT
Dep. of Mathematics and Computer Science
PATS Research Group
Middelheimlaan 1, 2020 Antwerp
BELGIUM

CHRIS BLONDIA
University of Antwerp -IBBT
Dep. of Mathematics and Computer Science
PATS Research Group
Middelheimlaan 1, 2020 Antwerp
BELGIUM

Abstract: The IEEE 802.11 wireless network uses a hard handoff mechanism and thus makes it hard for higher layer mobility protocols to implement lossless handovers. In this paper we will show that using IEEE 802.11 transmission feedback data, we can optimize our smooth Mobile IP handoff scheme.

Key–Words: IEEE 802.11, Mobile IP, fast handovers, crosslayer, smooth handovers

1 Introduction

Nowadays, wireless local area networks (WLAN), in particular those based on IEEE 802.11[4] technology, are deployed widely for a large variety of environments (home, enterprises, public hot spots,...). When the stations are mobile and may change subnet, not only the link-layer handoff procedure determines the perceived quality, but also the network layer handoff mechanism has an important impact.

Mobile IP (MIP)[11] was developed to provide a network layer handoff mechanism, but although it performs quite satisfactory in macro-mobility situations, it is not quite adequate when considering fast moving users which will change quickly from one network to another. The signalling delay introduced by MIP causes undesired connection interrupts. Various micro-mobility solutions were developed to trigger this flaw in the basic MIP behavior. The most know of them are Cellular IP[5], HAWAII[6], Hierarchical Mobile[7]. By handling as much as possible the handover locally, L3 delays will be minimized.

These protocols however do not take the L2 handover process in to account. The L2 handover itself will also need a certain time and introduces again some delay. The IETF introduced two fast handover protocols to tackle this problem: Pre-registration and post-registration[8]. The first one will try to make L3 registrations and prepare a L3 handover before an actual L2 handover takes place, while the latter will try to register via the nFA while its L3 connection is still managed by the oFA. with the old Foreign Agent (FA) while the L2 connection point has already been changed. Both protocols will need L2 information to perform their specific task.

In [3] we already discussed that it is not possible to deploy pre-registration on top of an IEEE 802.11 network. This protocol needs to be aware of the next point of attachment in the wireless network, but due to the 'break-before-make' behavior of the 802.11 handover procedure, this information is not available. The post-registration protocol could be adapted to fit on top of an IEEE 802.11 network as we showed in [3]. The introduced protocol makes it possible to perform smooth handovers, but the buffering scheme introduces some new problems. This scheme is based on a buffering solution at the oFA, by continuously queueing incoming packets at the oFA. After handover, an IP tunnel would be built between both FAs and the contents of this buffer would be flushed towards the new FA using an IP tunnel. Although this extension allows for loss free handovers, it has the drawback of possibly introducing duplicate packets at the receiver side. Because the old AP is unable to determine the actual time a Mobile node is leaving its network,due to the break-before-make behavior of the IEEE 802.11 , packets had to be buffered continuously in to a circular buffer. When this AP was eventually made aware of the handover, the entire buffer would be flushed towards the nAP and the station, including packets already delivered to the MN.

In this paper, we introduce a L2 transmission feedback based buffering scheme which will guarantee a smooth L2/L3 handover while minimizing the introduction of duplicate packets. The remainder of the paper is as follows: the next section will describe the mobility protocol used and will introduce the new buffering scheme. Section 3 will describe the implementation details of this buffering scheme and in section 4 we will discuss the results we obtained from our tests. We will round up this paper by formulating some conclusions.

2 Protocol Description

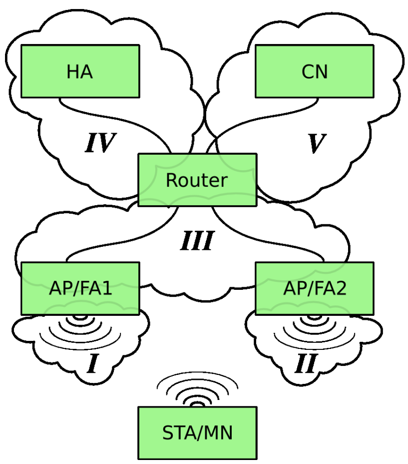

Figure 1: Reference network

In this paper, we will use a reference network as illustrated in figure 1. A mobile station is here connected to one of the two available Access Points (AP). Each AP also serves as a Foreign Agent (FA) for a different subnet (I or II). Having AP and FA functionalities available in one device makes it easy to couple their functionalities. In what follows, the terms AP and FA will both be used to describe the same device, but a distinction will be made in L2 or L3 functionality. Both FAs our connected in a common subnet with a central router. This router will provide the connectivity between the subnet of the FAs (III) and the Home subnet and the subnet (V) in which the Corresponding Node (CN) resides.

When we consider our extended post-registration Mobile IP fast handover scheme [3], a handoff will consist of the following phases:

1. Before handoff, the MN is connected via the oFA, which has a buffer for this MN in `STORE_AND_FORWARD` mode. All incoming packets for the MN will be stored in a buffer, while a copy is forwarded to the MN.
2. On deteriorating link conditions with the current AP, the station will start to scan for new reachable APs. We assume active scanning will be used to actively search for the best available AP. This implies that the station breaks contact with its current AP and starts probing all channels. This scanning phase completes by selecting the best available AP, based on link conditions.
3. Once the new access point is selected, the station will authenticate and associate with this access point. From this moment on, a new L2 connection is established.
4. Using the L2 identification of the newly associated STA (its MAC address), the nFA will query its neighbors to learn its oFA.
5. The oFA receiving the request will change the state of its buffer for this connection from `STORE_AND_FORWARD` to `STORE`. Newly arriving packets will be buffered and no packets will be forwarded until a L3 connection is established via the nFA.
6. An IP tunnel is constructed between the oFA and the nFA and the buffer state at the oFA will be changed to `FORWARD` in order to tunnel the queued as well as new arriving packets for the MN towards the nFA, which will deliver them to the MN.
7. A MIP registration can be performed afterwards (e.g. on an arriving advertisement from the nFA) to establish a direct link via the nFA.

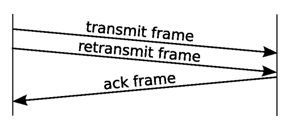

Figure 2: Successful transmission

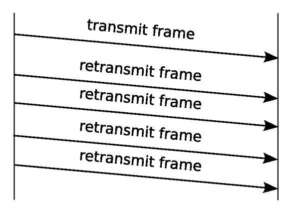

Figure 3: Failed transmission

We now introduce a new buffering scheme, that allows us to minimize duplicate packets, as well as to eliminate packet loss during handover. It is based on the feedback from the L2 frame transmission mechanism in 802.11 networks. In order to introduce some level of reliability into frame transmission, the 802.11 standard defines acknowledged data transmission. A frame which is lost or whose acknowledgment does not reach the sender due to the present state of the wireless network (collisions, low signal strength, collisions,...) will be retransmitted and on receipt of the acknowledgment, a successful transmission will be finished. This scenario is illustrated in figure 2. A sender will retransmit the packet for a predefined number of times before considering the packets as lost as illustrated in figure 3. The L2 knowledge about the transmission status of a frame can be used to optimize the buffering scheme we described earlier. We will now use this knowledge about the transmission status

Proceedings of the 5th WSEAS International Conference on Telecommunications and Informatics, Istanbul, Turkey, May 27-29, 2006 (pp70-75)

of frames to provide a more adequate buffering mechanism which will only stores packets for tunnelling if their transmission failed.

A FA will again have a buffer for each MN it has registered. This buffer however will be split up in two levels. At first all packets which will be delivered to the lower layer for delivery will be stored in a FIFO queue. Packets will be removed from this queue whenever transmission information for this packet is available. When L2 reports a successful transmission of a packet, it will be pulled from the queue and discarded. Whenever a transmission failure is reported, the pulled packet will not be discarded but stored in the second stage circular buffer. So this circular buffer will only contain the most recently packets failed to be delivered to the MN. In the event of failing link quality and a following handover, this queue will fill up with the packets that the AP tried to transmit towards the STA, but which it could not receive. These packets together with those not yet transmitted, which will reside in the first stage queue, will later be tunneled to the new FA. Packets which were already be successfully transmitted will thus no longer be retransmitted, while the oFA is able to buffer the needed packets without the knowledge of a MN leaving its coverage.

3 Implementation Details

3.1 Node configuration

The results described in the following section were obtained from our testbed which, for this set up, consisted of six off the shelf pcs. The basic setup for all machines was the same and summarized in the following list:

- cpu: AMD Sempron ™or Athlon ™2400+
- memory: 512 Mb
- nic: 2 Intel PRO/1000 (router node has 3)
- OS: Mandriva Linux release 2006
- kernel: 2.6.12-12mdk

Furthermore, the wireless nodes were equipped with a PCMCIA-PCI adapter hosting a Linksys WPC55AG wireless interface. These interfaces are equipped with an Atheros 5212 chipset and we used the Madwifi drivers[2] (old branch version 0.9.6.0) to control them.

The fasthandover protocol with feedback buffering was implemented using the Click Modular Router [9] version 1.4.3[9] with backports for some elements from the CVS release. Using Click makes it possible to implement packet handling by construcing a directed graph of elements. Each element in this graph will perform some basic actions on the packet and combinding these elements will result in the desired packet handling.

With the madwifi driver, it is possible to activate a raw device athXraw on top of the athX wireless interface. Using this device it is possible to send and receive raw 802.11 frames to and from the driver. These raw packets are encapsulated with Radiotap[1], which contain transmission parameters like link rate and rertansmission count for the outgoing packet or transmission details for incoming packets. Outgoing packets sent to a raw device will be returned to Click once they are processed. Their headers will be updated with the specific feedback information about the transmission of this frame. This information can then be further used in the buffering scheme as described below. This raw interface makes it possible to implement parts of the 802.11 mac in order to support Infrastructure based wireless networks as only acknowledging of frames and the optional RTS/CTS mechanism is still handled by the driver/firmware.

Handovers were forced by disabling the broadcast of beacons of the AP to which the STA was connected to. Whenever a STA detects it missed n consecutive beacons in a row (in the tests below $n = 3$), it will start a new scanning phase to find a new AP to connected to. Probe request which arrive at the oAP will not be answered, making it impossible for the STA to detect its previous AP and forcing it to associate with the other available AP which does broadcast beacons and answers probe requests at that time.

3.2 Feedback buffer implementation

The two stage feedback buffering mechanism is implemented in Click as illustrated in figure 4. A FA will need to set up such a buffer for each of its registrated MNs. Packets entering the buffer will have been marked earlier as newly arrived packets (`FRESH`), successfull transmission reports (`FEEDBACK_SUCCESS`) or packets reporting transmission failures (`FEEDBACK_FAILURE`). Based on the state of the buffer, `FORWARD`, `STORE` or `STORE_AND_FORWARD` , a packet will be sent along the Click graph without buffering via `output1` , it will be stored in the first stage buffer `SimpleQueue` or a copie will be stored, while the packet gets forwarded. The `ReversedPullSwitch` connected to the first stage buffer, will accept pull requests from both the `FeedbackUnqueue` element, as the general `Unqueue` element. The latter will only be active at a buffer flush, emptying the entire buffer, stage one as well as the circular buffer. In general, this `unqueue` element will be inactive, making `FeedbackUnqueue` in charge of `ReversedPullSwitch` `FeedbackUnqueue` is feeded with feedback packets, filtered by `IPFilter` to only provide packets forwarded by this node, as only these packets will be buffered in the first place. When a transmit success is received, the queue will be pulled via the `ReversedPullSwitch` and the resulting packet will be discarded. On arrival of a transmit failure, the unqueued packet will be stored in the second stage circular buffer, implemented by a `FrontDropQueue` . Using this scheme, the first stage queueu will always hold packets which are in transit. Their feedback state has not been received yet. Packets which reside in the `FrontDropQueue` had already been transmitted by the linklayer, but were not reported as successful. When a bufferflush is issued to this compound buffer by activating the `Unique` element, the `Priced` will ensure packets from the second stage circular buffer will be released first, followed by the packets from the first stage buffer. In this way, packets lost due transmission failure will be retransmitted in order followed by those packets which were not transmitted yet.

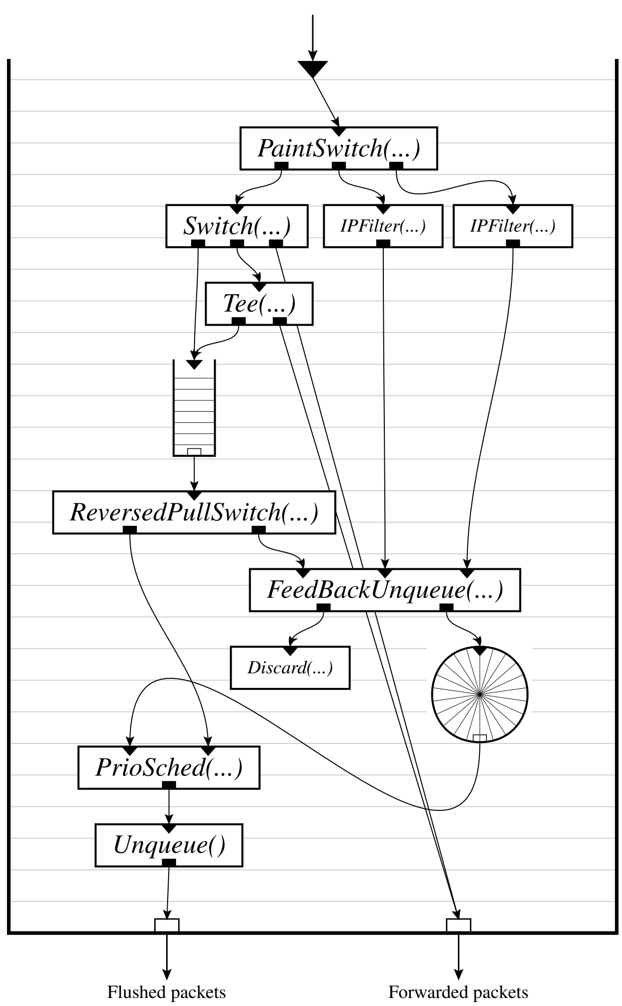

Figure 4: The compound Click element implementing a feedback buffer

4 Results

In this section we will discuss some results obtained from our testbed. We will first discuss some results for USP CBR connections after which TCP connections will be discussed.

4.1 UDP CBR traffic

The following results were obtained by sending a UDP CBR stream from the CN towards the MN. The stream was generated at 30 packets per second and had a payload of 500 bytes, resulting in a 130 kbps stream leaving the CN. The UDP payload started off with an RTP header, which made it easy to analyse for packet ordering and loss and duplicate accounting in ethereal [10]. Our testbed automation platform, Terran, performed 20 successive tests in a wireless 'clean' environment. No other access points or clients were active in the wireless channels used as well in the surrounding channels in order to minimize the effect of channel interference. In every test, the STA first associated with AP1 and immediately registers with the associated FA. A second later, the RTP stream was started. Again 20 seconds later, a handover interval of 60 seconds was defined. Whit in this one minute window, AP1 will randomly stop sending out beacons, triggering the STA to start the scanning phase. When associated with the new AP, the MN will listen for incoming Router Advertisements, which are broadcasted every 25 seconds, and registers itself with the new FA. No extra delay was introduced between router and HA, so signalling delay should be minimal. After this handover window, the test will run for another 25 seconds after which it will be aborted and data collected.

In order to clearly illustrate the problem, we will start by discussing a basic Mobile IP scenario. In figure 5 we plotted the number of packets lost during handover. It is clear that a big variation between the various tests exist. This is caused by the randomly chosen moment of handover with respect to the advertisement interval. The maximum number of lost packets would occur when handover was performed when an advertisement was just send out, making the node wait for the full advertisement interval. If we define the advertisement interval in seconds as a_i and p_r the packet rate per second, we can define the maximum expected packets to be lost as $M_{pl} = a_i * p_r$, which results to 750 packets in our scenario. The minimal loss will occur, when an advertisement will arrive immediately after the association.

When we take a look at our optimized handover scheme in which we continuously buffer packets in a circular buffer in order to anticipate a handover, we

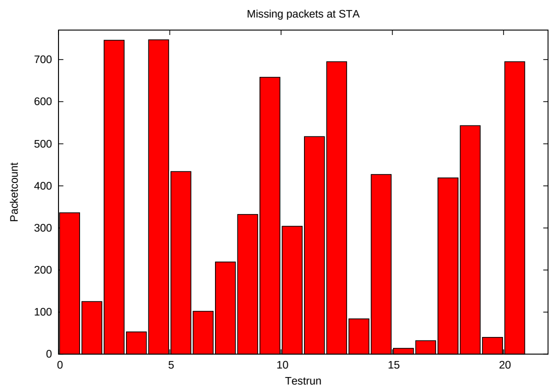

Figure 5: Number of missing packets at the STA with standard MIP.

expect the elimination of packet loss when the buffer is tuned to the traffic destined for the specific MN. As shown on figure 6, packet loss is completely eliminated with buffersize of 15 and 30. With a buffersize of 5, packet loss still occurs. There is however a drawback to this solution, which is illustrated in figure 7. When the buffer is overdimensioned, packet duplication will be introduced. This is caused by the fact that, at handover instance, the circular buffer will not only contain packets that were transmitted unsuccessfully, but also those that were successfully transmitted, but not yet removed from the queue by newly arrived packets.

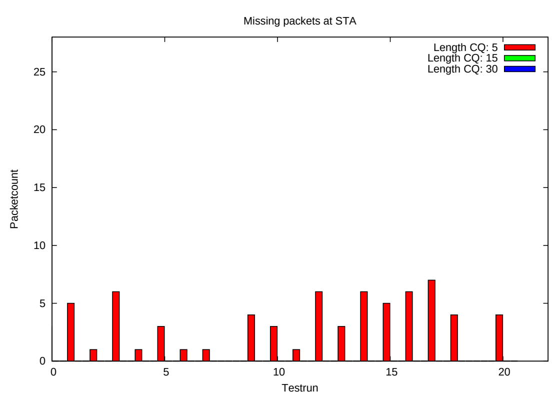

Figure 6: Number of missing packets at the STA with circular buffering.

The in this paper discussed solution to queue packets based on their transmission statistics proved to be very effective. It seemed that even with a small feedback queue of only 5 packets and a fresh queue of 5 packets, a complete lossless handover over all 20 tests could be achieved. Packet duplication could drastically be reduced as is shown in figure 8. A maximum of 2 duplicate packets was observed during our tests. These duplicates were caused by frames that were successfully received by the station, but the ack of these events were not received anymore by the oAP. As such, they were classified as transmit failures and queued for retransmission.

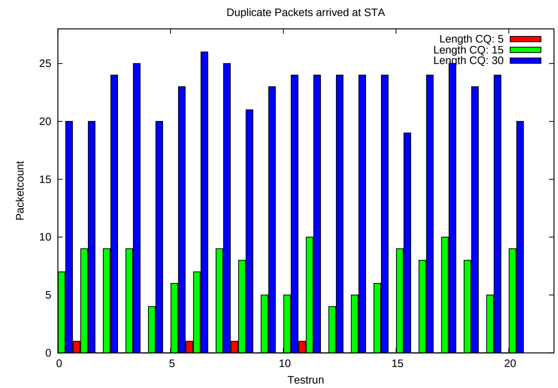

Figure 7: Number duplicate packets arrived at the STA with circular buffering.

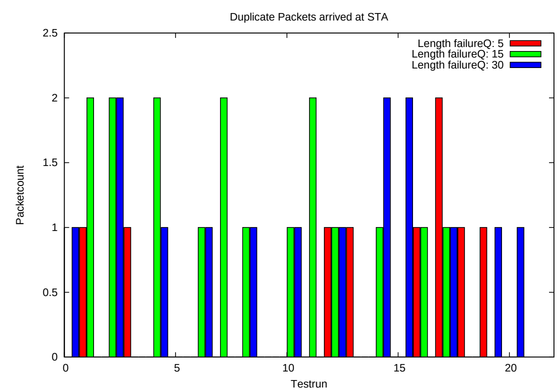

Figure 8: Number of duplicate packets at the STA with feedback buffering.

4.2 TCP connections

The UDP traffic of the previous chapter was limited to unidirectional traffic. TCP traffic with its flow control will set up a bidirectional between CN and MN. We will discuss here the scenario where a MN will fetch a file using ftp from a server located at the CN. After handshake, a data flow will be set up, sending maximum size data packets from the CN downstream towards the MN. The mobility scenario will be the same as the one in the previous section. One handoff will occur in a one minute interval while a file download is active. The download will start 15 seconds before this interval, to get a stabilized TCP connection.

When using a basic MIP scheme, the total number of retransmitted packets varies between 46 and 54 packets (Figure 10. Compared to the high number of packet loss in the UDP scenario where the data rate was much lower, this number might seem quite low. Flow control mechanisms of TCP however, limit the total amount of retransmitted packets.

In figure 9 we show the number of retransmitted packets during the ftp session. It is clear that our feedback buffering scheme minimizes packet duplication as compared to the MIP or the circular buffering scheme. The MIP and circular buffering schemes obtain almost similar results, altough the circular buffering does outperform MIP a bit. If we take a look at the total transmission time of the ftp connection (Figure 10, we do see that the circular buffering does outperform the MIP scenario. This phenomenon is caused by the sliding windows used by tcp for its flow control. In both case, MIP and circular buffering, the send window will be reduced and timeouts will occur, causing retransmissions of packets from the source. The connection itself in case of circular buffering will be restored much faster then in the MIP case, explaining the shorter transmit times. The feedback based queueing is clearly the better of the two, as it combines an even slightly faster transmission with a higher goodput, as packet duplication is far lower than in the other scenarios.

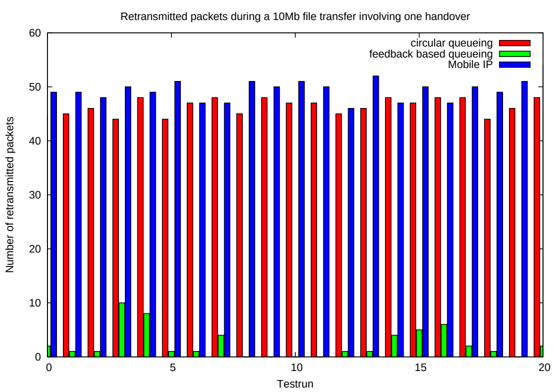

Figure 9: Number of retransmitted during a 10Mb file transfer.

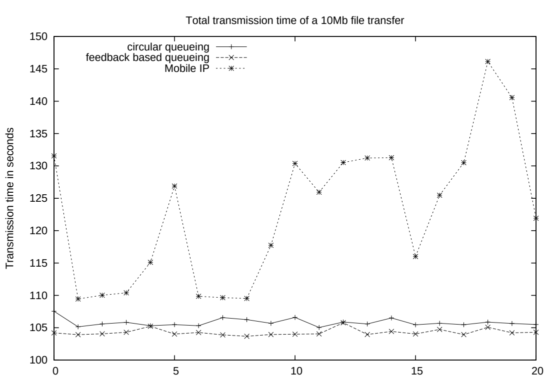

Figure 10: Transmission time of an ftp session of a 10Mb file.

5 Conclusions

In this paper we show that by using transmission feedback information from the IEEE802.11 link layer protocol, we can optimize a Mobile IP smooth handover scheme as proposed in [3]. We showed that packet loss can be avoided, by taking a look at a UDP CBR scenario while we also showed using with a TCP connection, that feedback data can minimize packet retransmission in a TCP connection optimizing transmission time as wel as network usage.

References:

[1] ieee80211_radiotap - software 802i11 stack packet capture definitions. Man Pages 9: Kernel manpages.

[2] Madwifi: Multiband atheros driver for wifi. http://www.madwifi.org.

[3] P. De Cleyn, N. Van den Wijngaert, L. Cerdà, and C. Blondia. A smooth handoff scheme using ieee802.11 triggers - design and implementation. *Computer Networks*, 45(3):345–361, 2004.

[4] IEEE Computer Society LAN MAN Standards Committee. Wireless lan medium access control (mac) and physical layer (phy) specifications. ansi/ieee std. 802.11. IEEE, New York, 1999.

[5] A.T. Campbell et al. Cellular ip. IETF Internet draft: draft-ietf-mobileip-cellular-00-txt, December 1999.

[6] R. Ramjee et al. Ip micro-mobility support using hawaii. IETF Internet draft: draft-ietf-mobileip-hawaii-00.txt, June 1999.

[7] E. Gustafsson, A. Jonsson, and C. Perkins. Mobile ip regional registration. IETF Internet draft: draft-ietf-mobileip-reg-tunnel-06.txt, March 2002.

[8] K. El Malki. Low latency handoffs in mobile ipv4. IETF Internet draft: draft-ietf-mobileip-lowlatency-handoffs-v4-11.txt, October 2005.

[9] Robert Morris, Eddie Kohler, John Jannotti, , and M. Frans Kaashoek. The click modular router. In *Proceedings of SOSP*, pages 217–231, Kiawah Island, South Carolina, December 1999.

[10] Angela Orebaugh, Greg Morris, Ed Warnicke, and Gilbert Ramirez. *Ethereal Packet Sniffing*. Syngress Publishing, February 2004.

[11] C. Perkins. Ip mobility support for ipv4, revised. IETF Internet draft: draft-ietf-mip4-rfc3344bis-02.txt, October 2005.