



# A smooth handoff scheme using IEEE802.11 triggers—design and implementation ☆,☆☆

Peter De Cleyn<sup>a</sup>, Nik Van den Wijngaert<sup>a</sup>, Llorenç Cerdá<sup>b,1</sup>, Chris Blondia<sup>a,\*</sup>

<sup>a</sup> *Department of Mathematics and Computer Science, University of Antwerp, Middelheimlaan 1, B-2020 Antwerp, Belgium*

<sup>b</sup> *Department of Computer Architecture, Technical University of Catalonia (UPC), C/Jordi Girona 1-3, 08034 Barcelona, Spain*

Available online 9 April 2004

## Abstract

This paper proposes a handoff scheme in a wireless access network where IEEE802.11 is used as link layer protocol and Mobile IP as network layer protocol. The scheme uses triggers available from IEEE802.11, together with packet buffering in the old Access Point and packet forwarding from the old to the new Access Point in order to provide a smooth handoff. The proposed scheme has been implemented on a Linux based testbed and it has been analysed by means of an ns simulation and an analytical model. The paper reports on the results obtained from the testbed, the simulation and the analytical model, both for constant bit rate traffic (in particular streamed RTP video) and TCP traffic.  
© 2004 Elsevier B.V. All rights reserved.

*Keywords:* Handoff; Mobile IP; IEEE802.11; Wireless access network

## 1. Introduction

Nowadays, wireless local area networks (WLAN), in particular those based on IEEE802.11

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.

A link layer handoff handles a change of access point (AP) to which a station is connected. As described in [1], a handoff can be divided into three phases. The first phase, referred to as detection phase, consists of the activities to discover the need for a handoff. The second phase, called search phase, is the acquisition of information needed to actually perform the handoff. Finally, the third phase is the actual execution of the handoff. It has been shown through experiments [1] that a handoff in IEEE 802.11 networks may lead to an

\* This work was supported by the Fund for Scientific Research Flanders under project G.0315.01, by Belspo Belgium under project IAP P5/11 MOTION (Mobile Multimedia Communication Systems and Networks) and by IWT under project 020152—End-to-end QoS in IP based Mobile Networks.

☆☆ This work was supported by the Ministry of Ed. of Spain under grant CICYT TIC-2001-0956-C04-01 and by the Department of Universities (Generalitat de Catalunya) under grant CIRIT 2001-SGR-00226.

<sup>\*</sup> Corresponding author. Tel.: +32-3-2653903.

*E-mail addresses:* [peter.decleyn@ua.ac.be](mailto:peter.decleyn@ua.ac.be) (P. De Cleyn), [nik.vandenwijngaert@ua.ac.be](mailto:nik.vandenwijngaert@ua.ac.be) (N. Van den Wijngaert), [llorenc@ac.upc.es](mailto:llorenc@ac.upc.es) (L. Cerdá), [chris.blondia@ua.ac.be](mailto:chris.blondia@ua.ac.be) (C. Blondia).

<sup>1</sup> Tel.: +34-93-4016798.

interruption of the connectivity between mobile node and network of up to 2 s, depending on the card that is used. This poor handoff performance is due to the duration of the detection and the search phase. This may result in an unacceptable packet loss and throughput degradation. Techniques to reduce this link layer handoff time are proposed in [1].

When a handoff involves also the change of access network (or subnet), layer 3 handoff mechanisms need to be provided to ensure the correct routing of downstream traffic. Mobile IP (MIP) [2] is the generally accepted method to handle this network layer mobility problem. MIP makes it possible for a Mobile Node (MN) to move among different IP sub(networks), even of different types, without the need to change its IP address (referred to as the MN's home address). Traffic is routed to a temporarily Care-of Address (CoA) assigned to the MN when away from its home network. The mapping between the MN's home address and the CoA is stored in the Home Agent (HA), an entity present in the home network. All traffic for the MN is intercepted by the HA and tunnelled to this CoA. Acting as the tunnelling endpoint for all the traffic intercepted by the HA, packets are de-tunnelled at the Foreign Agent (FA) (an entity present in the foreign network) and delivered to the MN, which is still using its home address to send packets. Every time the MN moves to a new network, it registers its new CoA with the HA through the FA.

Unfortunately, MIP suffers from two types of latencies that can deteriorate its performance. A first type of latency is due to the signalling delay between MN and HA: an FA is unable to process traffic for an MN until registration with the HA is completed. If this HA is a long routing distance from the FA, all traffic sent to the MN is either sent to the wrong CoA (the old one) or dropped (arriving at the new one before the registration is complete). Second, there is the handover induced latency: when e.g. using IEEE802.11 as the access technology, the MN is unable to receive any traffic at all. IEEE802.11 is a break-before-make protocol, meaning it will end its connection with the current AP before it is able to discover and associate itself with the new one.

The first of the latencies mentioned above is assumed to be improved adequately through the use of Hierarchical MIP (HMIP, also referred to as Regional Tunnel Management) [3]. This technique introduces a tree-like hierarchy of foreign agents. FAs are divided in sub-domains, branches of the tree, which are each governed by a Gateway FA (GFA). This GFA acts as the HA for an MN when it moves between FAs of the same branch of the tree. Multiple levels of hierarchy are possible, minimizing the path needed to complete a new registration. The handoff induced latency is more problematic, as this is caused by properties of the underlying MAC layer. The IETF has proposed a low latency handoff scheme [4] to counter the effects caused by the gap in layer 2 communication. This scheme describes methods for an MN to conduct its registration with the new FA (nFA) while still being connected to the old FA (oFA) or a way for the MN to postpone this registration until after the layer 2 handoff and still receive traffic sent to the oFA. The first is called Pre-Registration while the latter is called Post-registration. Both of these techniques rely on layer 2 triggers to be present in the system. If the underlying layer provides these triggers and manages to deliver them in time, the layer 3 handoff can proceed with very low latency. Unfortunately, IEEE802.11 does not provide the required triggers to implement the proposed solutions (see [5]). When an MN moves towards a new AP, it will disconnect from the old AP and scan for other APs in range. The choice of the new AP is based on the result of these scans. Hence, the use of Pre-Registration is ruled out as the identity of the new AP is not known in advance. The same problem holds for Post-registration.

In this paper, we propose the use of a buffering scheme combined with the limited available layer 2 triggers in IEEE802.11 to obtain a handoff scheme without packet loss. In Section 2 the reference architecture is defined and the problems that occur during handoff are discussed in detail. Section 3.1 presents the proposed protocol. In Section 4.1 we show how this protocol can be implemented and some experimental results are given. Section 5 discusses the simulation results obtained by an ns-simulation, while Section 6 presents an analytical

evaluation. Finally conclusions are drawn in Section 7.

## 2. Reference architecture and protocol

In this section we first describe the used reference network and identify the hosts used in our scenarios. The section then continues with a general description of the handover process and the description of the existing problems it introduces. To conclude we describe shortly the proposed solutions by the IETF and their drawbacks.

### 2.1. Reference network

The reference network is illustrated in Fig. 1. In order to optimize the exchange of information between L2 and L3 agents, the 802.11 Access Point (AP) and MIPv4 Foreign Agent (FA) functionalities are combined within one access router. In the following discussion, we will use the terms AP and FA to indicate the specific service offered by the same router and we assume that information between both functionalities can be exchanged. The old and new access routers are part of a different subnet and are connected to an Internet using a gateway router (GW). The Communicating Node (CN) and the Home Agent (HA) are also connected to this Internet.

### 2.2. Standard Layer 2 and Layer 3 handoff: components of the connection gap

In this section we identify the different components that contribute to the connection gap during

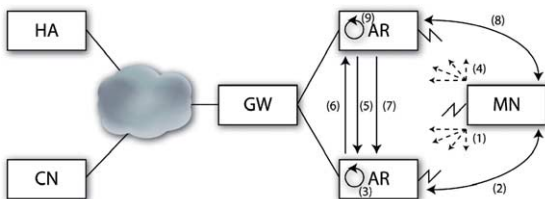


Fig. 1. Reference network and signalling.

handoff. The first segment in the connection gap is created by the L2 handover (Fig. 2(a)). IEEE802.11 uses a hard handoff mechanism, meaning that while searching for another access point (AP) to connect to, the mobile station (STA) disconnects from its current access point in order to scan the available channels for valid APs. The scanning can be performed in two ways: active or passive. In passive mode, the mobile station will listen for beacons periodically sent out by the APs, in general every 100 ms, and will use the supplied information to choose its next AP.

When using the active mode, the STA itself will send a probe request on a channel and wait for the possible responses. The STA waits for a certain period of time and will then change to the next channel and repeat this action. After this scanning phase, it will be able to choose the most appropriate AP from the ones that answered its probes.

Once the new AP (nAP) has been selected, the STA has to authenticate with this AP after which it can associate with it. The L2 connection is restored as soon as the association is finished.

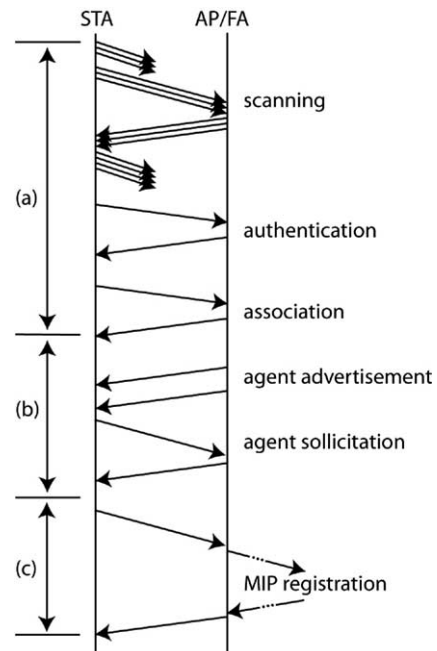


Fig. 2. Handover procedure.

While the STA is discovering new APs and is registering with the selected nAP, no L2 connection with the STA exists and thus no data can be transmitted to or from the MN.

After re-establishing the link layer connection, Mobile IP can come into the picture. The Mobile IP process at the MN should detect the change of access router and start the registration process with its Home Agent (HA). The detection of this change, using agent advertisements and solicitation [2] (Fig. 2(b)), can take up a considerable amount of time as we will show in Section 4.2. Once this nFA is located, the actual Mobile IP registration can start. The distance to the HA (Home Agent) and the network load conditions, will influence the length of this third component in the connection gap, but it will remain smaller than the L3 handover detection.

In order to optimize this registration process while taking into account both the L2 and L3 connection gap, the IETF proposed low latency handoffs [4]. Post- and Pre-registration depend on L2 information to establish a low latency handoff. Pre-registration tries to perform an MIP registration before the actual L2 handoff takes place. This requires the MN to know its next FA before leaving the current one. Using IEEE802.11, we cannot use this solution, as the scanning phase prevents us to select the next AP without leaving our current point of attachment.

Post-registration on the other hand tries to delay the MIP registration. After an L2 handoff, the MN remains registered with the oFA, but packets arriving at the oFA are tunneled towards the nFA and delivered via the new point of attachment. Post-registration also depends on L2 events at the oFA, indicating the start of an L2 handoff and is as such not acceptable when using IEEE802.11 as link layer.

### 3. A smooth handoff scheme using IEEE802.11 triggers

#### 3.1. Bridging the connection gap

The proposed scheme tries to bridge the L3 components (Fig. 2(b) and (c)) of the described

connection gap, by acting on triggers available for IEEE802.11. Due to the hard handoff mechanism we are however still confronted with the L2 component. Consequently, the scheme wants to provide a smooth, but not seamless handover.

In order to guarantee a lossless handover, datagrams arriving at the oFA while the STA is scanning and thus is disconnected, need to be buffered. As the oFA is unaware of the moment were the STA will disconnect from its AP, the oFA will have to buffer continuously, using a circular buffer. As a consequence, the oFA possibly introduces packet duplication when flushing the content of this buffer towards the nFA. Once the L2 handover finishes, the nFA can contact the oFA based on data received during association. A Bidirectional Edge Tunnel (BET) [4] will be set up between these two agents—using *Handoff request* and *reply* messages (HRqst/HRply) [4]—and the buffered packets at the oFA will be tunneled towards the nFA and delivered to the MN. The MN remains MIP registered with the oFA and all newly arriving datagrams will get forwarded via the tunnel towards the MN. Traffic originating from the MN will also pass through this tunnel and will be transmitted by the oFA towards its destination. L3 connectivity is thus re-established as soon as possible, i.e. right after L2 association. The MN can then decide to perform an MIP registration right away or even postpone this a moment, as L3 connectivity already has been realised.

#### 3.2. Layer 2 and Layer 3 information exchange

As mentioned in the above protocol description, the nFA should have the necessary L3 information, the IP address of the MN and its oFA, in order to set up the BET with the oFA, when the STA associates with the nAP. However, at the instant of association, only the L2 (MAC) address of the STA is known. We consider three possibilities for the nFA to obtain the necessary information: L2 handover adaptation, a distribution scheme or querying.

##### 3.2.1. L2 handover adaptation

In this scenario, adaptations should be made to the signalling during handoff. The purpose is to

provide the nAP with the necessary information during or just after registration. IEEE802.11 [6] has a flexible design of Management frames and the standard provides ways to extend the current frames. Non-compatible devices should just ignore any extra information provided.

Management frames in IEEE802.11 are built from information elements which in turn have a fixed layout. The first byte is a type field and it is followed by a length field, indicating the size of the variable length component following these two bytes. The type values 7–15 and 32–255 are reserved for future use, so we can define two extra types: a Home Address Information Element and a Previous Foreign Agent (PFA) Information Element, both with a 4 bytes payload field containing the respective IP addresses.

A first possibility is to add these Information Elements to the end of an association request, as illustrated in Fig. 3 and thus informing the nAP about the L3 status of the STA from the first moment on. A second possibility is to define also a new management frame, consisting of just the newly defined Information Elements. In this case the subtype value in the control fields of this frame should be set to a reserved value between 0110–1011 or 1101–1111. This frame should be sent by the STA right after receiving the association response.

### 3.2.2. Distribution scheme

Instead of depending on the STA to provide the necessary information to the nFA, the FAs can exchange information between neighbouring agents. A FA could transmit on a regular basis the L2 address, home address and home agent address of its registered nodes to its neighbours. This solution, however, does not guarantee that an FA has the necessary information when an STA associates with this AP. The advantage is of course that no changes are needed to the device drivers.

### 3.2.3. Querying

This last method will be the easiest to use, but probably also the less efficient. When receiving an association request from an MN, the FA will query its neighbouring agents and check if an MN

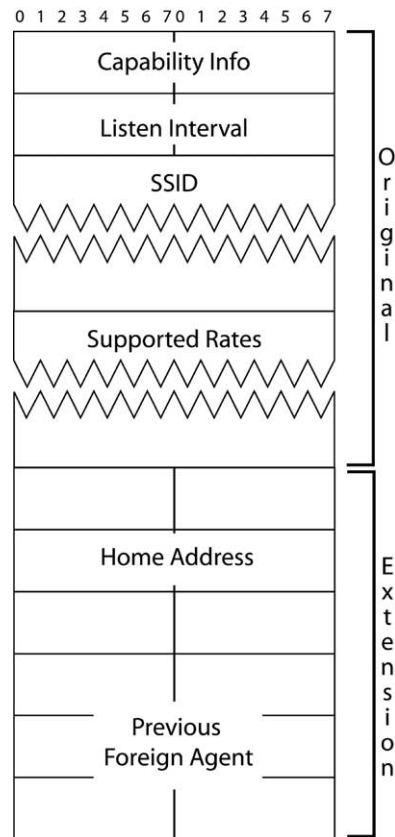


Fig. 3. Extended association request.

with the supplied MAC address is currently, as the oFA is not yet aware of the handoff, registered with this node. This querying can be done by sending an extended HRqst to the neighbouring FAs. This extension will contain the L2 address of the STA. The actual oFA will then respond with an HRply which now also contains the Home Address and Home Agent Address of the MN. Neighbours who received a HRqst but do not have the MN registered, can ignore the request or reply with a negative HRply.

### 3.3. Detailed protocol description

In this section we describe in detail the various stages an MN must run through from first connecting to a foreign network until MIP registration in another foreign network after performing a handover.

### 3.3.1. L2 connection to oFA

When associating with the oAP, the MN will communicate its Home Address and the IP address of its previous FA towards the nAP using an extended association request (Fig. 3). As this is the first connection in a foreign network for this MN, the PFA field will be empty. The AP will reply with the an association response to finalise the handover.

### 3.3.2. L3 connection to oFA

After the L2 connection with the oAP is realised, the MN can perform an MIP registration cycle with its Home Agent (HA) (Fig. 1(2)). When the oFA receives the registration reply (regrep), it will enable the buffer for the MN (Fig. 1(3)) and start buffering incoming packets.

### 3.3.3. L2 handoff

When a handover is needed, based on e.g. link quality readings, the STA will start a new scanning phase. (Fig. 1(4)) The oFA is unaware of this event and will continue to buffer and transmit packets on the air channel for the MN. The L2 handoff is performed as in 3.3.1, but this time the PFA field will be filled in with the IP address of oFA.

### 3.3.4. L3 update

Once the nFA has received the association request, it can contact the oFA using the info from the PFA field and request a tunnel with a HRqst (Fig. 1(5)) for the MN which is now connected with the nAP. This is the first event that informs the oFA of the MN's move. In response to the HRqst, the oFA will stop transmitting packets for the MN on the air link, but incoming packets will still be buffered. The oFA will acknowledge the request with a HRply (Fig. 1(6)) to the nFA. When using the querying method 3.2.3 the HRply will contain the L3 info needed by the nFA to build up its part of the BET. In this case the oFA needs to wait for a flush request (Fig. 1(7)) from the nFA before flushing the buffered packets in to the tunnel. In the other two cases the buffer can be flushed right away. In order for the MN to send traffic in to the network, it should install the new access router as its default gateway. To be able to do this, the nFA should send an ICMP

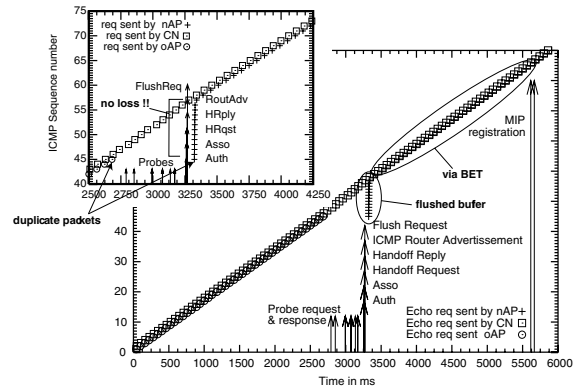


Fig. 4. Handover overview.

Router Advertisement packet [7] after sending an association response informing the MN of its next hop. Although the MN is still registrated with the oFA but connected via the nAP, the MN is now able to communicate fully using the BET.

Fig. 4 illustrates this whole process. The figure is constructed from a network capture and shows the ICMP echo requests towards the MN. We can clearly see the flushed buffer and the packets that pass through the tunnel when else no connection was available. Also remark that the proposed solution depends on the correct buffer dimensioning. Oversized buffers will introduce packet duplication, while too small buffers may lead to packet loss.

### 3.3.5. L3 handoff

As the L3 connection is restored, the MN can now decide to register with its nFA or it can postpone this L3 handoff (Fig. 1(8)). In the latter case, the tunnel lifetime of the original registration should be long enough to allow this delayed MIP registration.

## 4. Implementation and experimental results

In this section we first describe the set up of the testbed used to obtain the experimental results. We describe the hardware and software, as well as some techniques used to implement the various

parts of the agent functionalities. In a second part the obtained experimental results are presented and discussed.

#### 4.1. Implementation

The described experiments were performed on a Linux based testbed with software access points. The structure of this network is as shown in Fig. 5. The router in the center (GW) was used to control and monitor the other machines. For this purpose all hosts were equipped with at least two network interfaces. The first one was dedicated to control traffic and all hosts were connected within the same subnet with the controlling machine. The actual test network was constructed using a C-class subnet divided into several smaller subnets using a 27 bits wide subnet mask. The ARs could reach each other directly without an intermediate hop.

As stated earlier, the access point and foreign agent functionalities are incorporated into one access router. We used two identical Asus Pundit systems, equipped with an Intel Pentium 4 2.4 MHz and 512 Mb of memory. These barebones come with a PCMCIA-slot on board which made it easy to supply them with our Linksys WPC11 wireless cards. These cards were chosen as they use a prism2 chipset and thus are compatible with the HostAP 0.1.2 [8] drivers. These drivers provide the needed AP functionality, as they can put the wireless card in Master mode. In order to enable this mode, the firmware of the cards (factory default 1.4.2) had to be upgraded. We decided not to flash the cards, but to load the secondary firmware in to RAM. The latest available firmware release (1.8.0) was used in the tests. Each AP used a different channel, sufficiently spaced to avoid inter-

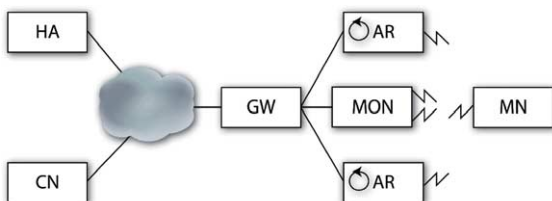


Fig. 5. Testbed architecture.

ference. In our tests channels 6 and 11 were used. The APs were configured to use a different ESSID (Extended Service Set ID) to simplify forced handovers. Both wireless NICs were also configured with different subnets. This forced a MIP handoff after performing an L2 handover. To provide the basic Mobile IP functionality, we chose for the HUT Dynamics implementation (0.8.1). This software was adapted to accept and handle the new messages as described in Section 2.2. The ARs used Mandrake 9.2, kernel 2.4.22-10, as OS.

The buffers are implemented using the netfilter [9] framework available in the Linux kernel. When an FA receives a request from an MN, it prepares an iptables rule to divert all incoming traffic for this MN, based on its home address, towards a userland application, before a routing decision is made. Using the iptables library, this user space application accepts packets from the QUEUE target module. Each buffer keeps an internal state, which is set by the FA depending on received MIP messages, which steers the decision to copy the received packets in to its buffer and forward the original to its destination or to drop it. When a buffer handles a flush request, it has to make sure that packets are not sent out back to back. A small traffic shaping mechanism is built in to ensure packets are not dropped at the nFA because of a too high arrival rate.

The Mobile Node is a Dell Inspiron 8500 laptop, with a Mandrake 9.1 on it using kernel 2.4.18. It is equipped with the same wireless NIC and drivers as the ARs. The mobile node software is adapted to react on the ICMP Router Advertisement. In order to force a handover when needed, the MN's ESSID is changed. This causes the MN to scan for a new AP with this new ESSID and to connect with the other AP. Because of the small scale of our network and the implementation ease, we chose to implement the querying Scheme (3.2.3) to translate L2 to L3 data.

To collect data, an extra node in our network is added to monitor the needed connections. This Monitoring Host is equipped with two WPC11 cards, both in monitor mode, to sniff the two used channels. The monitor mode enabled us to see the L2 signalling as well as the generated

traffic. The wired interface of this node is connected to the same hub where the CN is also connected to. This enabled us to capture traffic both on sending and receiving side of our set up on the same machine and thus using the same internal clock, which makes comparing time-stamps more accurate.

4.2. Experimental results

In this section we first show the need for a Mobile IP enhancement, by discussing the reaction time of Mobile IP on a change of Foreign Agents. We then investigate our protocol for UDP and TCP traffic.

In Fig. 6 we plot the time needed for the MN to react on an FA change with varying solicitation intervals. In all cases, agent advertisements are sent by the FA every 5 s. This delay is the second component we described in Section 2.2.

The maximum rate to send solicitation was defined to one every second. Knowing that the average handover latency was 0.85 s and registration latency is 30 ms, we can expect the minimum average delay introduced by a handover for packets arriving at the oFA during this handover to be at least 2.88 s. In the above discussed protocol we do not depend on MIP registration to re-establish the connection, so we can reduce this delay even with a non-optimal buffer configuration as is shown in Figs. 7 and 8.

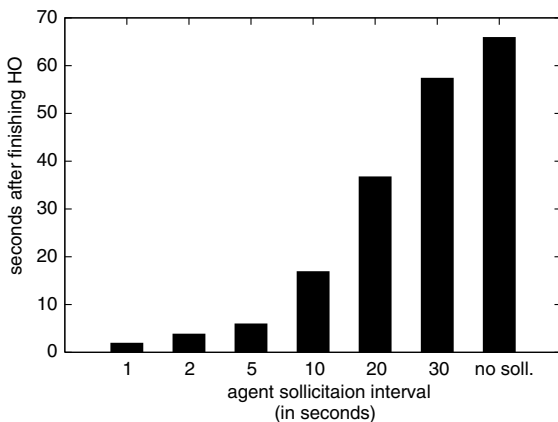


Fig. 6. Time before Mobile IP starts to react on an FA change.

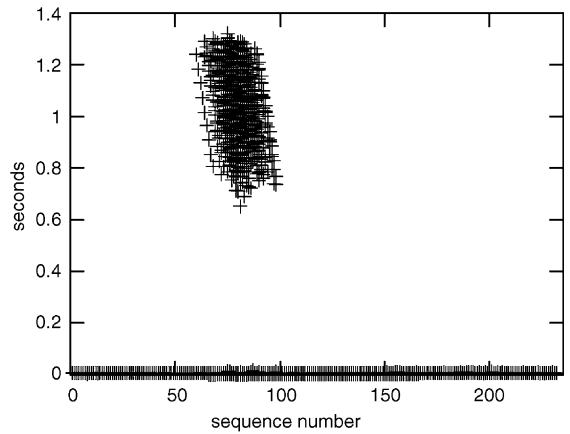


Fig. 7. End-to-end delay.

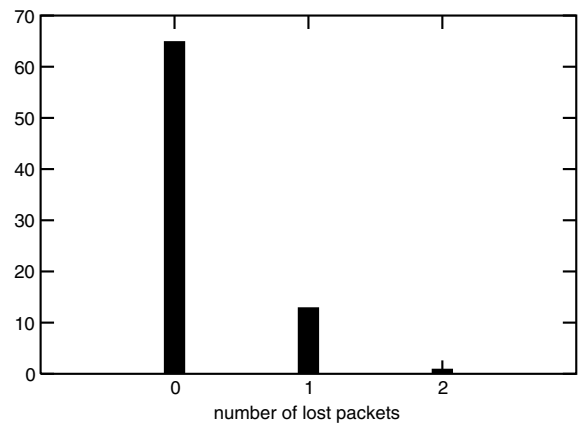


Fig. 8. Histogram of packet loss.

Figs. 7–9 show the behaviour of a CBR traffic stream with a non-optimal buffer. Packets of 1430 bytes were sent every 50 ms. A buffer of 20 packets is used to handle the handover. As Fig. 9 clearly shows, this buffer is almost twice as large as needed, introducing packet duplication. Packet loss, however, is almost always avoided (Fig. 8) and the extra delay for the buffered packets remains between 0.65 and 1.32 s as is shown in Fig. 7. These numbers remain well below the predicted 2.88 s we would find if we do not take immediate action after L2 handover completion.

As stated before, a not perfect dimensioned buffer will introduce packet loss or packet dupli-

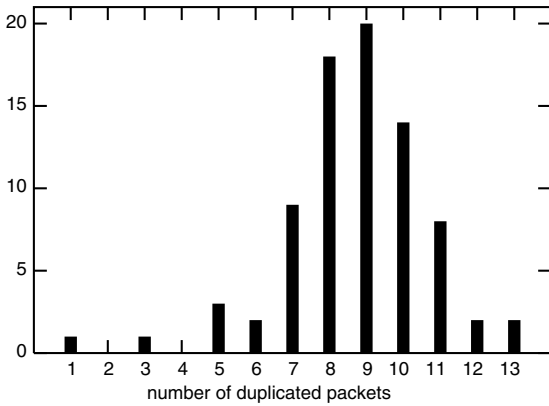


Fig. 9. Histogram of packet duplication.

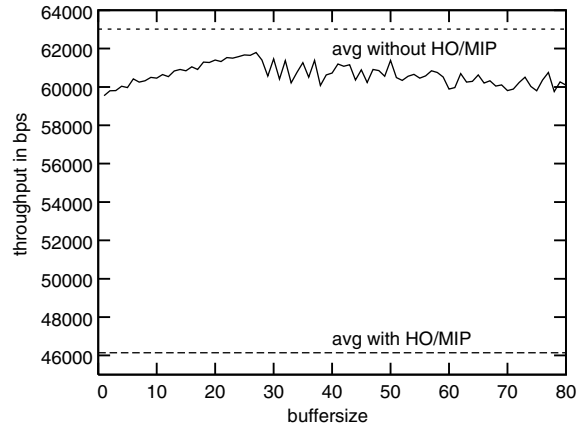


Fig. 11. TCP throughput.

Fig. 10 clearly illustrates this phenomenon, showing us the behaviour of streamed RTP video towards the MN with various buffer sizes. The optimal buffer size is located around 20 packets. A smaller buffer size will introduce packet loss. Without a buffer all packets sent during a handover are lost. Larger buffer sizes will decrease the packet loss, reaching a minimum from 20 packets onwards. Buffer sizes above 20 will introduce a growing number of duplicate packets. If packet loss remains very small, the image quality will be minimally affected, and the same is true with a small amount of duplicated packets. An increased amount of duplicate packets will, however, increase the jitter introduced on the stream. The player thus needs a large enough playout buffer to

compensate this jitter or packets may still get dropped by the application.

TCP will also suffer from this buffer behaviour. Fig. 11 shows the drop in throughput when loosing or duplicating packets. The maximum bandwidth for this connection was limited to 512 kbit in the central router. This resulted to the maximum throughput without handover as shown at the top of the graph. At the bottom is the average over again 100 runs with one handover during the transfer and just standard MIP. Figures for our implementation were obtained by averaging the throughput measured at several runs per buffer size. These results show that buffering and forwarding clearly increases the achieved throughput, although the occurrence of a handoff still has a negative influence on the throughput. We can observe again that an optimal buffer size for this stream exists and is situated around 28 packets.

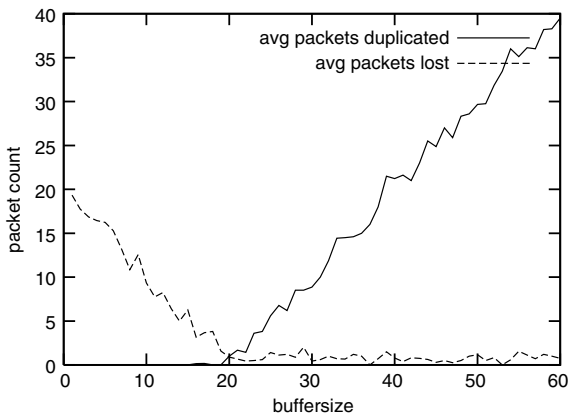


Fig. 10. Packet delivery.

### 5. Simulation results

We have implemented the protocol described in Section 2.2 using the *network simulator*, ns. We have modified the ns version with micro-mobility support that can be found in [10]. In [10] the handoffs are completely managed at layer 3. We first modified this version of ns in order to have L2-trigger support. We used it in [5] to confront

MIP without L2-triggers with other IP-mobility protocols using L2-triggers: MIP with L2-triggers, Pre-Registration and Post-registration. To obtain the results presented in this paper we have further modified ns adding active scanning and the buffer mechanism described in Section 2.2.

Fig. 12 shows the topology used in the simulations: traffic is sent from the CN to the MN while the MN handoffs from AR1 to AR2. Both AR use non-interfering channels. We have considered TCP downloads of 80 packets of 1500 bytes each which are transmitted during a handoff. We have used TCP-Sack with a granularity of 0.1 s and an advertised window of 60 packets. All links in the fixed part are 10 Mbps and have a propagation delay of 1 ms. In the wireless part we have used 11 Mbps. Furthermore, we have introduced exponentially distributed background traffic with packets of 1500 bytes loading the links in the fixed part to 0.8 and 0.5 in the wireless. Thus, a maximum rate of approximately 2 Mbps is available for the TCP download. Background traffic in the wireless part is sent from the ARs to background wireless nodes (one for each AR).

The results have been obtained repeating the handoff of the MN between AR1 and AR and averaging the measures until the confidence interval was reasonably small (we simulated more than 50 handoffs for each scenario). The router advertisements were sent from ARs every 1 s. In order to avoid phase effects, we started the sequence of router advertisement and beacons randomly each time the mobile movement between ARs was initiated. For the circular buffer at the AR we used a timeout timer of 0.5 s (i.e. only packets with an age

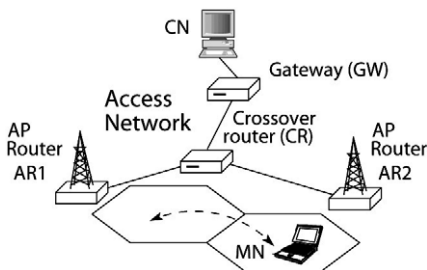


Fig. 12. Simulation topology.

lower than 0.5 s were forwarded from the old AR to the new AR).

We define the *handoff latency* as the time since the connection is lost with the current AR (because the MN goes out of coverage with the AR), until the MN is associated with the new AR. The rule the MN uses for the handoff detection is not to receive a beacon during 2.2 beacon intervals. In order to obtain results for different handoff latencies we used different values of the 802.11 beacon period. Note that using 2.2 beacon intervals, the handoff detection takes 1.7 beacon periods on average.

The active scanning triggers were fixed to  $\text{MinChannelTime} = 2$  ms and  $\text{MaxChannelTime} = 10$  ms. The MN waits  $\text{MinChannelTime}$  before scanning a new channel if no activity is sense, otherwise it waits  $\text{MaxChannelTime}$  (see [6]). The active scanning was repeated until a probe response was received. This may happen more than once in case of collision of the probe request. However, since we did not saturate the wireless channel, this event rarely happened in the simulations. Assuming only one scanning per handoff and neglecting the packet transmission time, the average handoff latency is approximately given by  $9 \times \text{MinChannelTime} + \text{MaxChannelTime} + 1.7 \times \text{beacon period}$ . Fig. 13 confronts this approximation with the average handoff latency measured in the simulations. As shown in the figure, we have varied the beacon period from 20 to 500 ms.

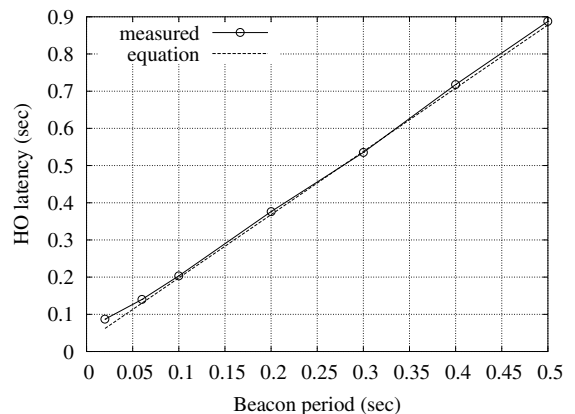


Fig. 13. Handoff latency—beacon period.

Fig. 14 depicts the events occurring during a handoff between AR1 and AR2 without using the buffer at the AR (case A) against using it (case B). The figures show: (i) the instants when the packets are sent, marked with a thick dot (indicated as *Tx instants* in the figure); (ii) the time when the MN goes out of coverage with the current AR; (iii) the lost packets, marked at the transmission time with a square; (iv) the reception instants of these packets, indicated with a different cross when the packets are delivered by AR1 or AR2 (indicated as *Rx from AR1* and *Rx from AR2*). In order to better follow the reception instants, these packets are also connected with a line; (v) the packets tunneled from AR1 to AR2, marked with a circle (indicated as *Rx from AR2 tunneled from AR1*); (vi) the transmission instants of the beacons transmitted by AR1 or AR2 that are received by the MN; (vii)

the transmission instants of the HO-Req and HO-Rep packets used to establish the tunnel between AR1 and AR2 (indicated as *PReq/PRes*); (viii) the transmission instant of the router-advertisement sent by the AR and the corresponding registration request sent by the MN. Note that after registering with AR2, the gateway delivers the packets destined to the MN directly to AR2, and thus, they do not use the tunnel between AR1 and AR2. This packets can be identified in Fig. 14A because they are not marked with the circle (which is used to identify the tunneled packets).

Fig. 14A shows that 20 packets are lost during the handoff. This forces TCP to timeout approximately 0.25 s after the first packet is lost and to initiate a slow-start phase. Fig. 14B shows the same handoff when a buffer of 10 packets is used. The trace shows that still losses occur (since more than 10 packets were lost). However, sack is able to fast recover the lost packets and go again to the congestion avoidance phase, thus, resulting a shorter transmission time.

In order to measure the impact of the handoff on the TCP connection we shall focus on the goodput of the file download. We define the goodput as the file size transmitted ( $80 \times 1460 \times 8$  bits) divided by the transmission time. Fig. 15 shows respectively the goodput (case A) and the expected number of packets lost (case B) for different values of the beacon period. Recall from Fig. 13 that we vary the beacon period in order to have different values of the handoff latency (see Fig. 13). Fig. 15 shows some interesting results that will be explained with the help of the traces shown in Figs. 15–17. First of all, Fig. 15A shows that using buffers may allow a significant increase of goodput. For instance, using a beacon period of 20 ms, we have respectively a goodput of 950 and 1454 kbps without buffer and using a buffer of 20 packets. However, it is surprising that with the same beacon period and a buffer of 5 packets, we obtain a goodput of 812 kbps, i.e. lower than without buffer! The explanation of having lower goodput using the buffer of 5 packets is explained with the traces of Fig. 16. The trace (A) shows that without buffer, the TCP connection continues the transmission after the handoff when a retransmission timeout occurs approximately after 0.25 s

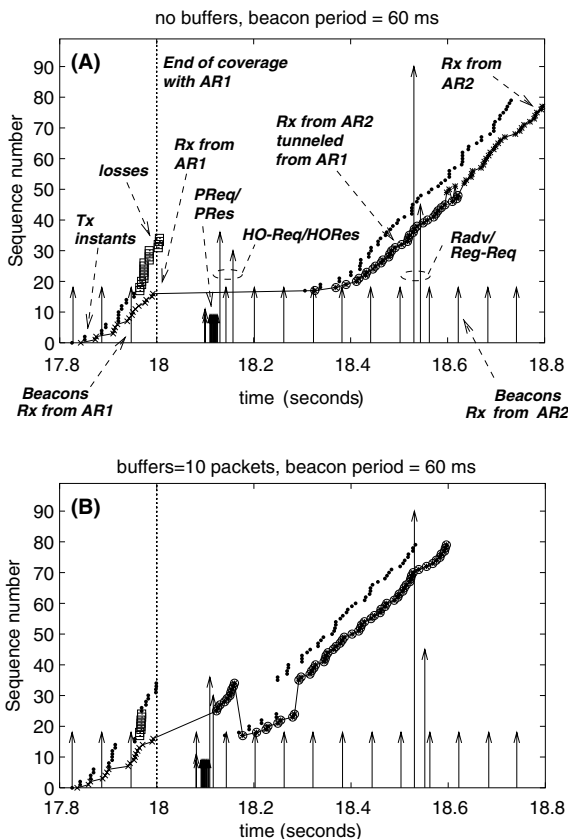
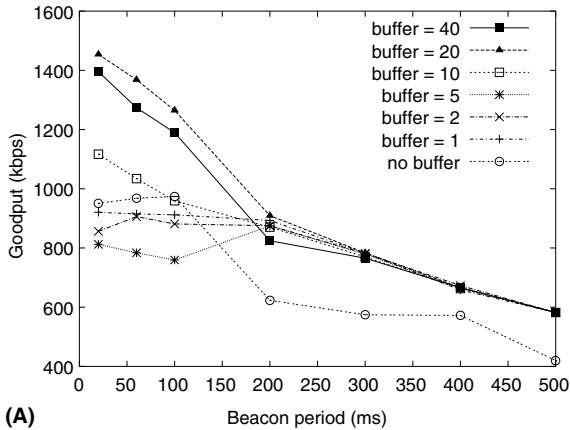
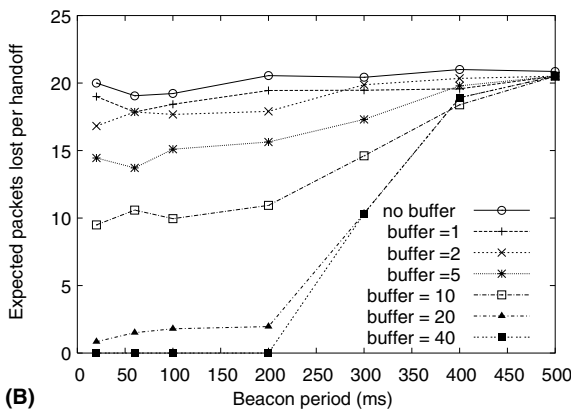


Fig. 14. TCP trace without buffering (A), and using a buffer of 10 packets (B).



(A)

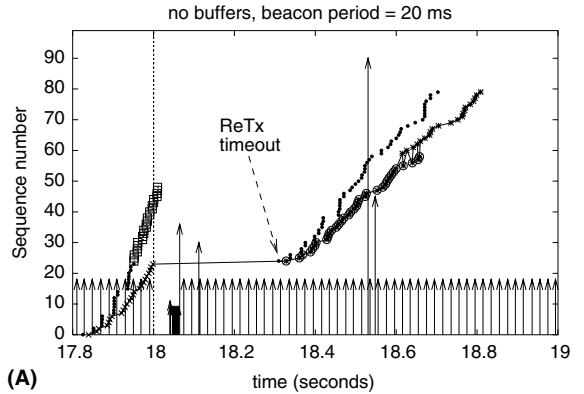


(B)

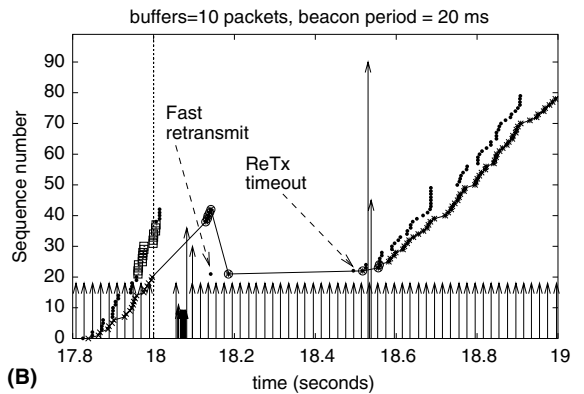
Fig. 15. Goodput (A), and expected number of lost packets (B) for TCP sources.

from the end of coverage with AR1. Trace (B) shows that the acks due to the 5 buffered packets, trigger a fast-retransmit in the TCP sender. However, since too many packets were lost, this fast-retransmit is not enough to recover, and the transmission stops until the retransmission timeout occurs. Therefore, this failed fast-retransmit increases the transmission time, and thus, the goodput decrease. We note that another TCP implementation that would transmit more packets when the partial ack from the fast-retransmit is received (e.g. newreno), would have been able to recover without waiting for the retransmission timeout.

Another interesting result from Fig. 15 is that when increasing the handoff latency, the goodput improvement tends to be the same for all buffer



(A)



(B)

Fig. 16. TCP trace with a beacon period of 20 ms without buffering (A), and using a buffer of five packets (B).

sizes  $\geq 1$ . Of course, since we use an age timer of 0.5 s for the packets in the buffer, for high handoff latencies TCP is possibly experiencing retransmission timeouts, and only this retransmitted packet falls within the age timer (and thus, only one packet is forwarded by the tunnel irrespectively of the buffer size). In addition, Fig. 15 shows that the goodput starts converging when there are still significant differences between the number of packet losses for buffer sizes  $\geq 1$ . For instance, for a beacon period of 200 ms, the goodput falls between 825 and 900 kbps for  $\geq 1$ , while it is only of 620 kbps when no buffer is used. This result is explained by the traces of Figs 17A–C. Fig. 17A shows the trace when no buffer is used. This trace shows that the packet sent when the first retransmission timeout is triggered is lost. Since the backoff is duplicated, TCP waits 0.5 s before the

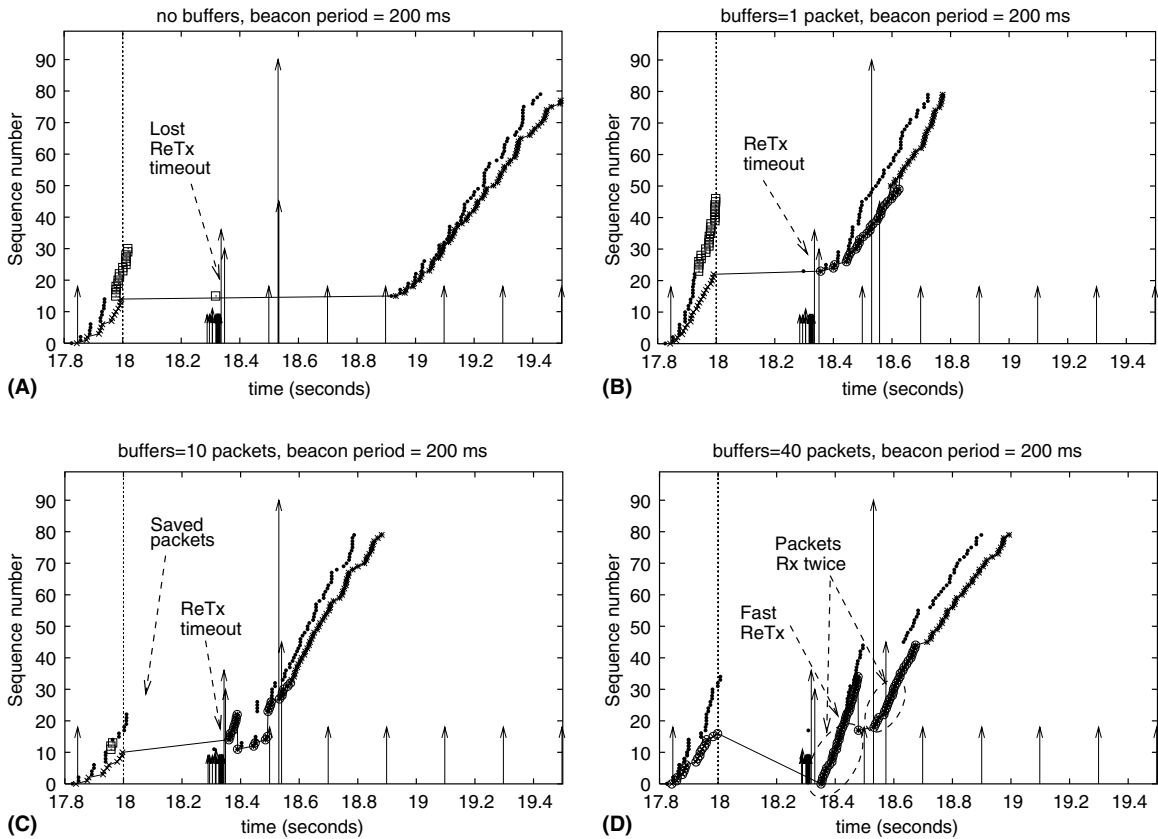


Fig. 17. TCP trace with a beacon period of 200 ms without buffering (A), using a buffer of 1 packet (B) using a buffer of 10 packets (C), and using a buffer of 40 packets (D).

retransmission of the same packet. Traces of Fig. 17B and C show that using a buffer size  $\geq 1$  allows the MN to receive the first retransmission, thus, avoiding to wait for the second timeout. Furthermore, these traces show that the transmission time needed to finish the TCP transmission is nearly the same in both cases, although nine more packets are lost using a buffer of 1 packet (Fig. 17B), than using a buffer of 10 packets (Fig. 17C). The reason is that TCP has already initiated a slow-start phase after the retransmission time-out, thus, it takes approximately the same time to open the window in both cases.

Finally, Fig. 15 shows that having a large buffer may not be always convenient. For instance, this figure shows that a buffer of 20 packets in our scenario leads to a better goodput for all handoff latencies than using a buffer of 40 packets. Fig.

17D explains this result. This figure shows that many of the 40 packets forwarded to AR2 have already reached the MN. This produces false fast-retransmits and the retransmission of packets already received by the MN, thus, reducing the performance.

## 6. Analytical model

### 6.1. Model description

In this section we present an analytical approach to model the handover protocol and to evaluate its performance. The reference network used is shown in Fig. 18. All routers are modeled as M/M/1 queues, resulting in an exponentially distributed response time of a packet traversing a

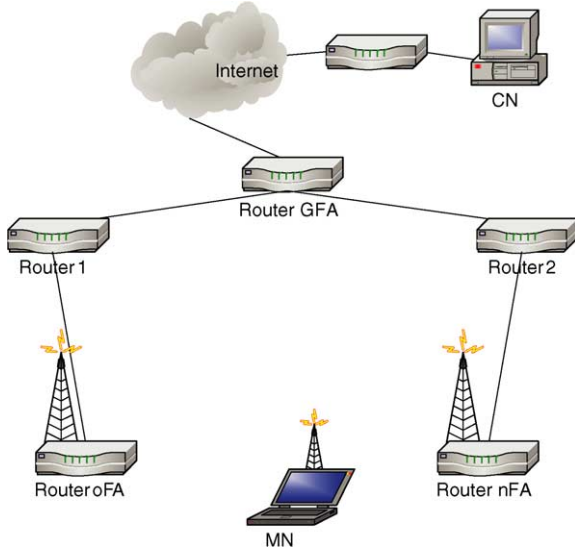


Fig. 18. Reference network.

router. The Foreign Agents (FAs) are assumed to be collocated with the layer 2 access points and also function as routers.

In order to describe packet traversal from a Correspondent Node (CN) to a Mobile Node (MN) during a handover, we need to define several time instants. Let  $t_0$  be the point in time where the handoff starts and let  $D_{LD}$ , when the layer 2 connection is severed with the old Foreign Agent (oFA), coincide with  $t_0$ . At time instant  $D_{LU}$  the layer 2 handoff will be finished and a connection will be established with the new Foreign Agent (nFA). Packet forwarding from the oFA to the nFA starts when the Handoff Request (HRqst) arrives at the oFA at time  $t_1$ .

Now consider a constant bit rate (CBR) stream of UDP packets sent from the CN to the MN every  $T = 10$  ms. Neglecting the potential jitter introduced before a packet reaches the Gateway Foreign Agent (GFA), the first packet arrives at the GFA at time  $t_0 - 100$  ms. Any given packet from the stream can then (1) be forwarded directly to the MN if it arrives at the oFA before  $D_{LD}$ , (2) be dropped at the oFA due to buffer overflow if it arrives after  $D_{LD}$  and before  $t_1$ , (3) be buffered and forwarded to the nFA at  $t_1$  or (4) be forwarded directly to the nFA if it arrives after  $t_1$  at the oFA.

We can express above cases in a more formal manner: let  $t_{GFA}^k$  be the time instant when the  $k$ th packet from the stream arrives at the GFA and then define the following four classes:

1.  $t_{GFA}^k + X_2 + link\_delays < D_{LD}$ ,
2.  $(t_{GFA}^k + X_2 + link\_delays > D_{LD})$  AND  $(t_{GFA}^k + X_2 + link\_delays < t_1)$  AND  $(t_{GFA}^k + BT + X_2' + link\_delays < t_1)$ ,
3.  $(t_{GFA}^k + X_2 + link\_delays > D_{LD})$  AND  $(t_{GFA}^k + X_2 + link\_delays < t_1)$  AND  $(t_{GFA}^k + BT + X_2' + link\_delays > t_1)$ ,
4.  $t_{GFA}^k + X_2 + link\_delays > t_1$

with  $t_1 = D_{LU} + Y_4 + link\_delays$  and  $BT = BS \times T$ , where  $BS$  is the size of the buffer in the oFA. By modeling the classes this way, we label a given packet  $k$  as lost due to buffer overflow when the  $k + BS$ th packet also needs to be buffered, i.e. arriving at oFA before  $t_1$ .  $X_2$  and  $X_2'$  are sums of two exponential variables and  $Y_4$  is the sum of four exponential variables, which are the response times of the routers a packet traverses.

Since these classes are exhaustive and mutually exclusive, we can for example calculate

$$P({}_{GFA}E_{MN}^k > t) = \sum_{i=1}^4 P({}_{GFA}E_{MN}^k > t \cap k \in class(i))$$

with  ${}_{GFA}E_{MN}^k$  the end-to-end delay of the  $k$ th packet between the GFA and the MN. Note that

$$P({}_{GFA}E_{MN}^k > t \cap k \in class(i)) = P(k \in class(i)),$$

if the  $k$ th packet is dropped when it ends up in  $class(i)$ .

## 6.2. Numerical results

Now consider the network depicted in Fig. 18. Each router has a load of 0.8 and a service rate of 1 packet/ms, yielding an exponentially distributed response time with rate 0.2 packets/ms. Each link introduces an additional fixed delay of 5 ms. The UDP stream consists of 30 packets, of which the first one arrives at the GFA at time  $t_0 - 100$  ms.

Fig. 19 shows the expected packet distribution among the different packet classes, for different buffer sizes at the oFA. Classes 1 and 4 are not affected by a change in buffer capacity, as  $D_{LD}$  and

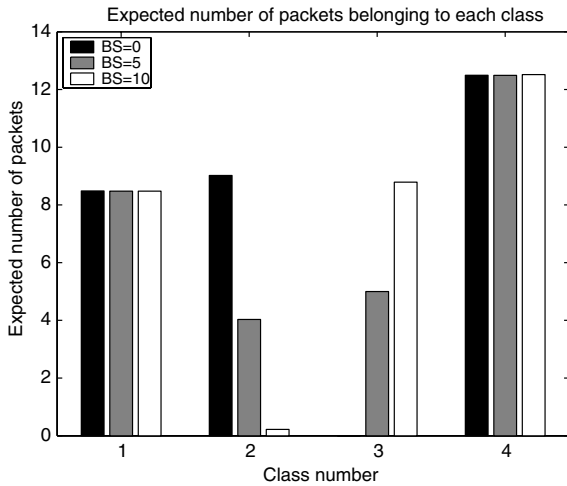


Fig. 19. Packet distribution per class.

$D_{LU}$  are fixed at 0 and 50 ms, respectively. Consequently, the expected value of  $t_1$  is not changed either. Classes 2 and 3 are variable, as fewer packets are lost at the oFA and more packets are forwarded to the nFA as BS increases. We can also look at the delay distribution for each individual packet in the stream. Fig. 20 shows the probability that a given packet's end-to-end delay is larger than a certain time  $t$ . Packets 1–6 are directly delivered to the MN, while packets 7–9 have a certain probability of being buffered and are possibly dropped at the oFA due to buffer overflow

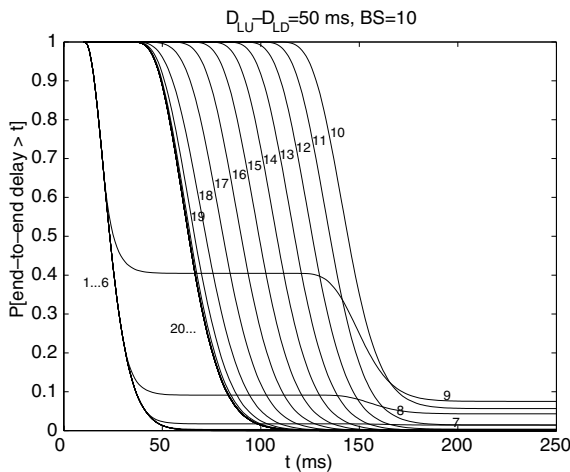


Fig. 20. Individual packet delays.

(curves do not tend to zero). Packets 10–19 are expected to be buffered before being forwarded and all subsequent traffic (20–30) is forwarded to the nFA immediately, subjected to a higher end-to-end delay than the first few packets.

Fig. 21 shows the loss probability due to buffer overflow at the oFA for every packet in the stream. A buffer capacity of 10 packets leads to packet loss probabilities well below 10%. Finally, Fig. 22 clearly illustrates the dependence of expected buffering requirements on the length of the interval  $[D_{LD}, D_{LU}]$ .

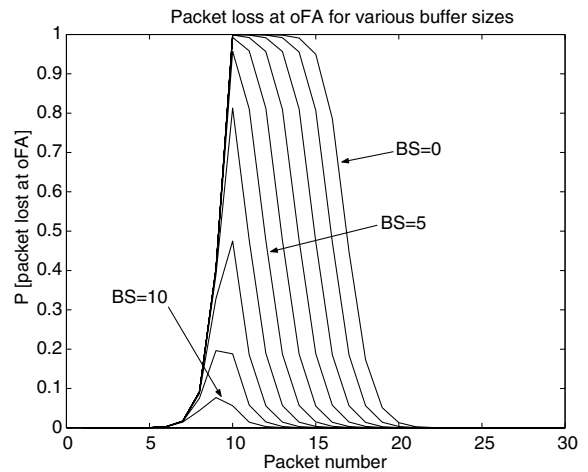


Fig. 21. Loss probability at oFA.

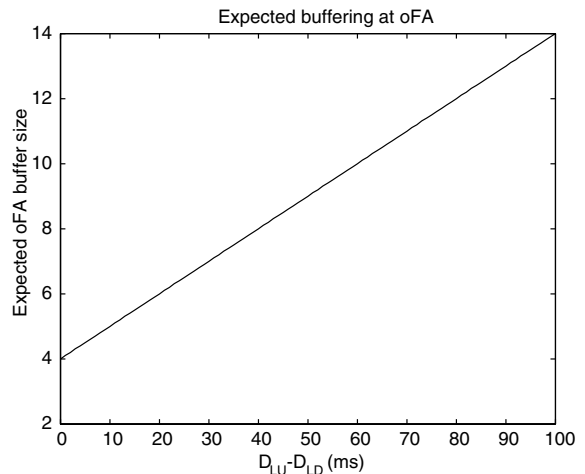


Fig. 22. Expected buffer size.

## 7. Conclusions

This paper proposes a handoff scheme in a wireless access network where IEEE802.11 is used as link layer protocol and Mobile IP as network layer protocol. The scheme uses triggers available from IEEE802.11, together with packet buffering in the old Access Point and packet forwarding from the old to the new Access Point in order to provide a smooth handoff. The proposed scheme has been implemented on a Linux based testbed and it has been analysed by means of an ns simulation and an analytical model. The experiments and the evaluations show that the dimensioning of the buffer is crucial: a small buffer size may introduce packet loss while overdimensioning the buffer may lead to duplicated packets. Experiments on the testbed show that when using a streaming application, such as streamed RTP video, the image quality is minimally affected when the number of lost packets or duplicated packets is small. However, a high number of duplicate packets leads to an increase of the jitter, which requires a large playout buffer at the receiver side. The results of the experiments and evaluations for TCP traffic show a significant increase of the goodput when using buffers and forwarding compared to standard Mobile IP. Furthermore, when the handoff latency increases, the TCP goodput tends to keep the same value, independent from the buffer size.

## References

- [1] H. Velayos, G. Karlsson, Techniques to reduce IEEE802.11b MAC layer handover time, Technical Report KTH/IMIT/LCN/R-03/02, KTH, Royal Institute of Technology, Stockholm, Sweden, April 2003.
- [2] C. Perkins, RFC 2002: IP Mobility Support, October 1996.
- [3] E. Gustafsson, A. Jonsson, C.E. Perkins, Mobile IPv4 regional registration, Internet draft: draft-ietf-mobileip-reg-tunnel-08.txt, November 2003.
- [4] K. El Malki, et al., Low latency handoffs in Mobile IPv4, Internet draft: draft-ietf-monileip-lowlatency-handoffs-v4-04.txt, 2002.
- [5] C. Blondia, O. Casals, L. Cerdà, N. Van den Wijngaert, G. Willems, P. De Cleyn, Low latency handoff mechanisms and their implementation in an IEEE802.11 Network, in: J. Charzinski, R. Lehnert, P. Tran-Gia (Eds.), Proceedings of the ITC 18: Providing Quality of Service in Heterogeneous Environments, vol. 5b, Elsevier, Berlin, 2003, pp. 971–980, cost279ref03004.
- [6] IEEE, ANSI/IEEE Std 802.11, 1999 Edition (1999).
- [7] S. Deering, RFC 1256: ICMP router discovery messages, 1991.
- [8] Host AP driver for Intersil Prism2/2.5/3, Available from <URL <http://hostap.epitest.fi>>.
- [9] The netfilter/iptables project: the Linux 2.4.x/2.5.x firewalling subsystem, Available from <URL <http://www.netfilter.org>>.
- [10] ns with Micromobility Support. Available from <URL <http://comet.ctr.columbia.edu/micromobility>>.



**Peter De Cleyn** obtained his Masters in Computer Science from the University of Antwerp (Belgium) in 2000. He is currently a Ph.D. student at this university, where his main research interests, as member of the research group “Performance Analysis of Telecommunication Systems” (PATS), are related to IP micromobility, IEEE802.11 wireless networks, Linux emulations and implementations for telecommunication problems and performance analysis. He has published some papers in international journals and conference proceedings.



**Nik Van den Wijngaert** obtained his Masters in Mathematics and Computer Science from the University of Antwerp in 2001, where he since then has been pursuing his Ph.D. in the PATS research group. His main interests are protocol modeling methodologies, seamless mobility and performance analysis. He has published a number of papers in international journals and conference proceedings.



**Llorenç Cerdà** obtained the engineering degree in telecommunications in 1993 from Polytechnic University of Catalonia (UPC). He joined the Computer Architecture Department of UPC in 1994, where he received the Ph.D. degree in 2000 and currently he is Assistant Professor. His areas of interest include TCP, IP micromobility, wireless 802.11 networks, C++ programming, simulation, performance evaluation, queueing networks. He has participated in several EU funded research projects (ACTS, IST, COST), research projects in collaboration with Nokia (TESI), and the startup SouthWing.



**Chris Blondia** obtained his Ph.D. in Mathematics from the University of Ghent (Belgium) in 1982. Between 1986 and 1991 he was a researcher at the Philips Research Laboratory and from 1991 till 1994 he was an associate professor at the Computer Science Department of the University of Nijmegen (The Netherlands). In 1995 he joined the Department of Mathematics and Computer Science of the University of Antwerp, where he is currently a professor, head of the research group “Performance Analysis of Telecom-

munication Systems” (PATS) and head of the Department. His main research interests are related to both theoretical methods for stochastic modeling (in particular queueing systems) and to the design, the modeling and performance evaluation of telecommunication systems, in particular related to traffic management in broadband networks, IP mobility management, medium access control for wireless and wired access networks, etc. He has published a substantial number of papers in international journals and conference proceedings. He is editor of the “Journal of Network and Computer Applications” and member of IFIP W.G. 6.3 on “Performance of Computer Networks”. He is and has been involved in several European research programs (RACE, ACTS, IST and COST).