An Enhanced Inter-Access Point Protocol for Uniform Intra and Intersubnet Handoffs

Chun-Ting Chou and Kang G. Shin, *Fellow*, *IEEE*

Abstract—An enhanced IEEE 802.11 Inter-Access Point Protocol (IAPP) is proposed to provide a unified solution for both intra and intersubnet handoff processes. The proposed enhancement relies on the access point's (AP's) interoperability with other APs, provided by the IP-based IAPP, so as to enable the *intra and intersubnet link-layer frame buffering-and-forwarding*. This enhancement not only eliminates frame losses during an intrasubnet handoff but, more importantly, realizes loss-free, fast intersubnet handoffs without modifying the IP-mobility protocols such as Mobile IP. Our ns-2-based simulation results show that the intersubnet handoff process is transparent to the mobile host's TCP session. Moreover, the enhanced IAPP supports higher user mobility and achieves a higher TCP throughput—up to 50 percent improvement over the original IAPP.

Index Terms—Mobile IP, IEEE 802.11 wireless LAN, fast/smooth handoff, TCP performance.

◆

1 INTRODUCTION

The explosive growth in demands for wireless services and applications has ignited extensive research into wireless network architectures and protocols. Two key focuses of these research efforts are to: 1) provide a sufficient transmission capacity for bandwidth-demanding applications and 2) provide seamless connections for mobile hosts. Although the transmission capacity has been improved from a kbps range to a Mbps range over the last decade, realization of seamless connections has been hindered by inevitable handoffs resulting from user mobility.

A handoff occurs when a mobile host moves from its current radio access cell/network to a new one. During a handoff, the mobile host cannot send or receive any data since the current connection (i.e., a link between a mobile host and its previous wireless AP) has been torn down and the connection with the new AP has not yet been established. This "blackout" time interval is referred to as *handoff latency* and ranges from hundreds of milliseconds to several seconds, depending on the type of a handoff. In general, there are two types of handoffs: intra and intersubnet handoffs. In an intrasubnet handoff, the APs involved in that handoff reside in the same IP subnet. A mobile host only needs to establish a link-layer connection (with the new AP) without modifying its IP address. Therefore, an intrasubnet handoff is also referred to as a *link-layer* or *layer-2 handoff*. A typical example of intrasubnet handoff occurs when a wireless station moves across between two APs of an IEEE 802.11 wireless LAN [1]. In an intersubnet handoff, the APs involved in that handoff reside in two different IP subnets. A mobile host not only needs to establish a link-layer connection (with the new AP) as in an intrasubnet handoff, but also needs to obtain a new IP address to maintain IP-layer reachability. Therefore, an intersubnet handoff is also referred to as an *IP-layer* or *layer-3 handoff*. Fig. 1 depicts these two types of handoffs and the relation between them.

To facilitate the handoff process, beacons or router advertisements are implemented in a variety of wireless networks. For example, in an IEEE 802.11 wireless LAN, the APs periodically broadcast beacon frames so that a mobile host can use these beacons as an indication of whether or not to initiate an intrasubnet handoff. In an IEEE 802.11 wireless LAN, the beacon interval is 100 milliseconds, incurring a link-layer handoff latency of $100 \sim 450$ milliseconds [2], [3]. For the case of Mobile IP [4], [5], access routers (ARs) periodically broadcast router advertisements which contain the subnet prefix information. A mobile host can then determine if it has moved to a new IP subnet based on the received advertisements and decide whether or not to initiate an intersubnet handoff. The default advertisement interval in Mobile IP is 1 second, incurring an IP-layer handoff latency of up to 3 seconds.

Since a handoff latency of several seconds is unacceptable for most delay-sensitive and loss-sensitive applications (e.g., TCP-based applications), numerous approaches have been proposed to reduce the IP-layer handoff latency. An effective approach is to take advantage of the link-layer handoff process. Since an IP-layer handoff always starts with the establishment of a link-layer connection, an ongoing link-layer handoff is a good indication of a potential IP-layer handoff. By using this link-layer indication, a mobile host can initiate an IP-layer handoff much earlier than waiting for the Mobile IP router advertisements, primarily because of the relatively short link-layer handoff latency.

Many fast-handoff schemes have been developed based on the link-layer handoff indication. For example, the fast handoff protocols designed for Mobile IP networks rely on these link-layer indications to expedite the Mobile IP binding update process [6], [7], [8], [9], [10], [11]. Although these schemes can reduce the IP-layer handoff latency,

---

- *C.-T. Chou is with Philips Research USA, 345 Scarborough Road, Briarcliff Manor, NY 10510. E-mail: chun-ting.chou@philips.com.*
- *K.G. Shin is with the Real-Time Computing Laboratory, The University of Michigan, 1301 Beal Ave., Ann Arbor, MI 48109-2122. E-mail: kgshin@umich.edu.*

Manuscript received 20 July 2004; revised 13 Dec. 2004; accepted 23 Dec. 2004; published online 27 May 2005.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0227-0704.

1536-1233/05/$20.00 © 2005 IEEE Published by the IEEE CS, CASS, ComSoc, IES, & SPS

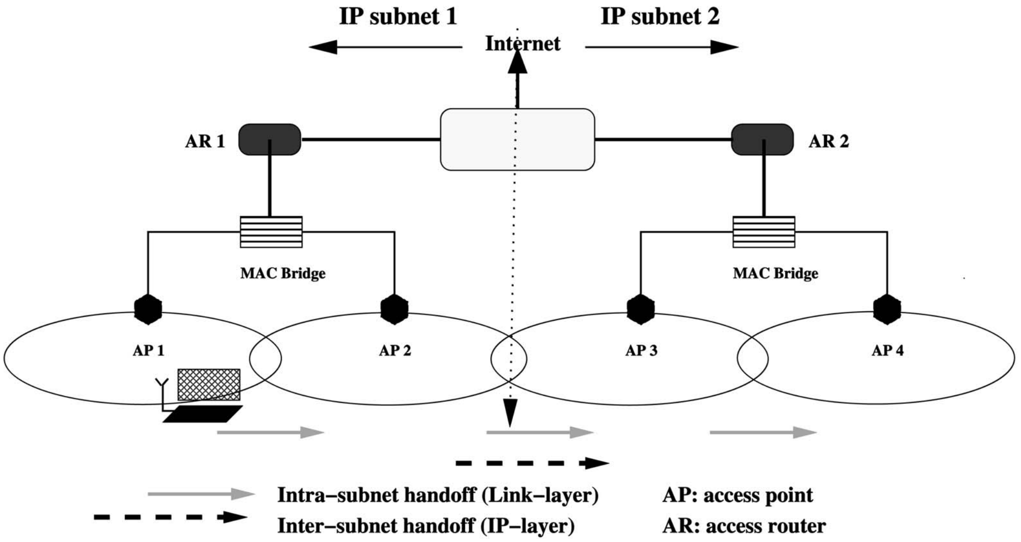


Fig. 1. Intrasubnet (link-layer) and intersubnet (IP-layer) handoffs.

there exist some practical issues that need to be addressed carefully. For example, in the "preregistration" fast handoff schemes [7], [8], it is assumed that link-layer indications are available before a link-layer handoff takes place. Thus, protocol signaling for handoffs between the mobile host and ARs can be completed prior to loss of connectivity. A practical problem with this is that mobile hosts usually cannot switch to different radio "channels" (to locate new APs) while still maintaining the connection with the old AP, unless the mobile host has multiple radio interfaces [12] or the old AR has a priori knowledge of the host's next target AR.

The "postregistration" schemes in [7], [8], [13] seem more practical in the sense that they only require link-layer indications to be available after a link-layer handoff is completed. However, they may cause false alarms since a link-layer handoff is not necessarily followed by an IP-layer handoff. Therefore, the mobile host or the AR must decide whether or not to initiate the fast handoff procedure once it receives these link-layer indications. The easiest approach to this problem is to let the mobile host send out a *Router Solicitation* packet whenever the mobile host receives a link-layer handoff indication. However, the neighbor discovery protocol [14] requires 1) a mobile host to randomly delay the initial *Router Solicitation*[1] and 2) an AR to randomly delay the solicited *Router Advertisement*.[2] These delays can easily add up to significantly degrade the performance of fast intersubnet handoff schemes. Another approach is to include the subnet prefix or "IP identifier" of the AR in the IEEE 802.11 beacon frame [15], so that a mobile host can determine if it has moved to a new IP subnet directly by examining the link-layer indication. The other approaches require the new AR to establish a bidirectional edge tunnel (BET) with the mobile host's old AR [13], [16] or send an unsolicited router advertisement to the mobile host, based on mobile host's IP identifier information provided in the link-layer indication.

There exist other enhancements for the IP-layer handoff which do not use link-layer indications. A handoff-dedicated MAC-layer bridge has been used to facilitate IP-layer handoffs [17]. The problem is that connecting two IP subnets with a MAC bridge make them *link-layer visible to each other* so that the intersubnet handoff problem does not even exist, and it only works for the micromobility scenario. Some IP-layer approaches such as hierarchical mobility agent [18], [20], S-MIP [19], and HAWII [21] have also been proposed to reduce the signaling delay for the micromobility scenario. Some other approaches use a handoff-dedicated entity such as "mailbox" to reduce the signaling delay [22] or use caching agents to send solicited router advertisements more promptly [23]. There also exist many multicast-based approaches, such as [24], [25], [26], to reduce packet losses during an intersubnet handoff. A potential problem is that multicasting wastes more network resources and requires tracking mechanisms or a priori information to determine which ARs the packets should be multicast to.

In this paper, we propose a new handoff scheme based on the IEEE 802.11f standard [27], also known as the Inter-Access Point Protocol (IAPP). IAPP enables the IEEE 802.11 APs to communicate with each other and facilitates context transfer for mobile hosts. By using IAPP, the reassociation/reauthentication process is shortened and so is the link-layer handoff latency. However, IAPP cannot prevent the link-layer frame losses. To remedy this problem, we add a simple but effective functionality— link-layer frame buffering-and-forwarding—to the existing IAPP. Although one can immediately benefit from the resulting loss-free link-layer handoffs thanks to the proposed enhancement, our main goal is to apply this enhanced IAPP to provide a uniform handoff scheme with a low handoff latency and zero data loss for both intra and intersubnet handoffs. The differences between the proposed approach and the existing approaches are summarized as follows:

1. Our approach achieves the same IP-layer handoff latency as those using link-layer indications. Moreover, our scheme prevents frame losses during the

---

[1] To alleviate congestion whih may occur when many hosts start up on a link at the same time.

[2] So a single advertisement can respond to multiple solicitations.

"believed-to-be-short-enough" link-layer handoff. We will show via an implementation on a testbed how link-layer frames get lost during a "fast" link-layer handoff and how these losses degrade the performance of higher-layer applications.

2. Because IAPP is essentially an IP-based protocol, even cross-IP subnet frame-forwarding can be handled by the APs using the IAPP. Therefore, neither multicasting nor IP-layer buffering-and-forwarding is needed for smooth IP-layer handoffs, thus wasting less network resources.
3. Our approach uses a unform procedure—an IAPP-based frame buffering-and-forwarding—for both micro and macromobility scenarios, as compared to the existing IP-layer schemes which use 1) regional/local registration schemes for the micromobility scenario and 2) original Mobile IP for the macro mobility scenario. Therefore, the intelligence of differentiating link and IP-layer handoffs is not needed.
4. Because of items 1, 2, and 3, the Mobile IP remains intact, and no additional IP-layer, handoff-dedicated devices/agents are needed.

The rest of this paper is organized as follows: In Section 2, we show via a testbed the frame-loss problems during a link-layer handoff and discuss the consequence and solutions to this problem. We introduce the current IAPP and present the enhanced IAPP in Section 3. We also show how both the link and IP-layer handoffs benefit from the enhanced IAPP. We present the protocol details and discuss the *ns-2* simulation results in Section 4. Finally, conclusions are drawn and the direction of our future work is discussed in Section 5.

2 FRAME LOSSES IN A LINK-LAYER HANDOFF

Even though the link-layer handoff latency is as low as hundreds of milliseconds, the link-layer handoff itself is still problematic. The problem, as we will show in this section, may degrade the performance of those fast IP-layer handoff schemes using link-layer handoff indications. To show the potential degradation, we establish a testbed and demonstrate how the relatively small link-layer handoff affects the TCP performance. The setup of our testbed is shown in Fig. 2, where AP1 and AP2 run under the Linux operating system and use D-link IEEE 802.11b wireless LAN cards with Prism2 chipset. The wireless station (STA) also runs Linux but uses a Cisco IEEE 802.11b wireless LAN card. Two FTP servers, one local server (FTP server 1) and one remote server (FTP server 2), are both considered in order to study the impact of round-trip time (RTT) on the TCP performance. FTP server 1 runs Linux with finer timer granularity such that the TCP retransmission timeout (RTO) is about 500 msecs (as shown in Fig. 3), while the RTO of the FTP sessions with FTP server 2 is about 2 seconds because of the coarse timer granularity and larger minimal RTO value used in Unix machines [30].

2.1 Scenario I: Small Round-Trip Time

Fig. 3 plots the TCP sequence numbers of the STA's FTP session with FTP server 1 throughout a link-layer handoff. The FTP session is interrupted by unplugging the cable between AP1 and the bridge for about 3 seconds (starting at around the 42nd second) before the STA's handoff in order to obtain the RTO value, which is about 500 msecs in this setting. After the handoff takes place at 45.3 seconds, all TCP segments destined for the STA get lost. Upon completion of the handoff, one can observe that some new TCP segments taking the new route (due to the link-layer update frame) arrive at AP2. These TCP segments are transmitted from the sender's TCP congestion window because the TCP sender receives some acknowledgements right after the handoff. These acknowledgements are those that cannot be sent by the STA before the handoff and are sent via the new AP after the handoff. Due to the losses of some TCP segments during the handoff, the TCP sender times out eventually and the first lost segment is retransmitted (about 500 msecs after it was transmitted for the first time). This result shows that, even though the link-layer handoff latency is small, a TCP retransmission timeout can still be triggered due to link-layer frame losses, thus degrading the throughput.

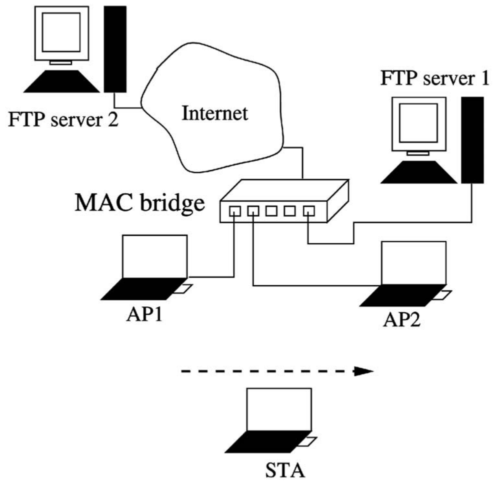


Fig. 2. A testbed of TCP performance during a link-layer handoff.

To remedy the problem shown in Fig. 3, we modify the drivers of the APs' LAN cards in order to support link-layer frame buffering-and-forwarding for the STA [2], [29]. The TCP sequence numbers under this new setting are shown in Fig. 4. One can observe that upon completion of the handoff, all TCP segments buffered at AP1 during the handoff are forwarded to the STA via AP2, and no retransmission timeout occurs. Note that forwarded segments and the TCP segments taking the new route (due to the link-layer update frame) arrive at AP2 interleavingly because of the small RTT in this setting. However, TCP can handle this type of out-of-order delivery without invoking fast retransmit since the number of out-of-order TCP segments is always less than 3 in our experiment.

2.2 Scenario II: Large Round-Trip Time

Fig. 5 shows the TCP sequence numbers of the STA's FTP session with FTP server 2 during a handoff. All the TCP segments arriving at the AP1 during the handoff simply get lost if there is no link-layer frame buffering-and-forwarding. Upon completion of the handoff, some new TCP

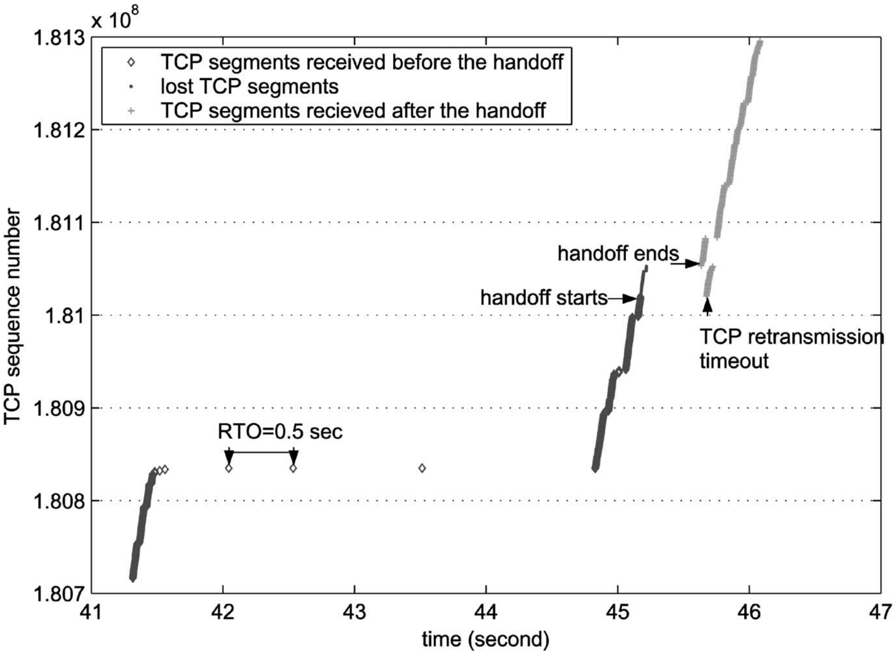


Fig. 3. TCP performance—Scenario I: small RTT without link-layer frame forwarding.

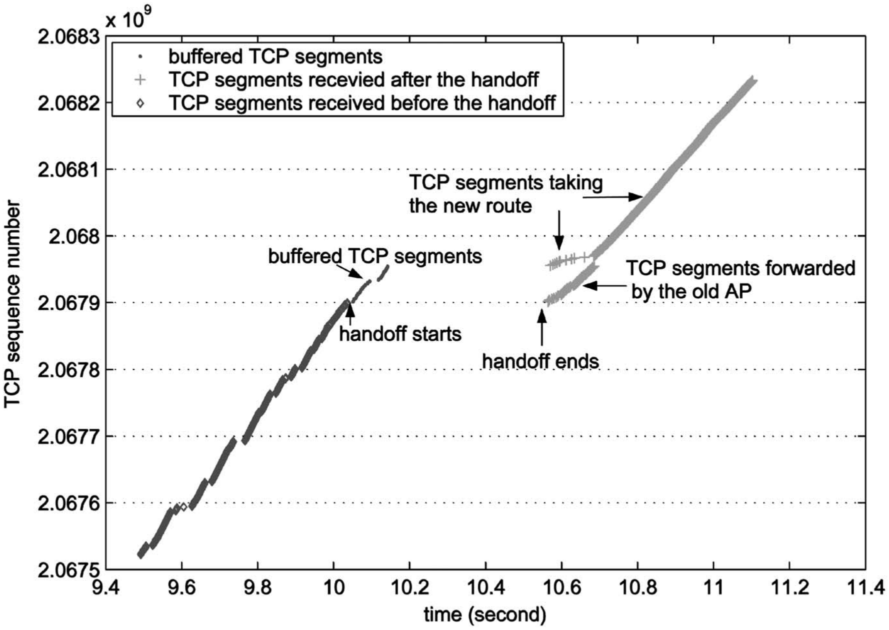


Fig. 4. TCP performance—Scenario I: small RTT with link-layer frame forwarding.

segments arrive at the STA via AP2 as in the previous cases. Unlike the first case in which the RTT is small, no TCP retransmission timeout occurs because of the larger value of RTO and the relatively small link-layer handoff latency. Instead, the out-of-order delivery (i.e., the new TCP segments via the new route) will invoke TCP fast retransmit such that the lost segments are retransmitted at 27.5 second. This undue invocation of fast retransmit again reduces the TCP throughput. Fig. 6 shows the TCP sequence numbers in the case where the APs support link-layer frame buffering-and-forwarding. Upon completion of the handoff, the TCP segments buffered at AP1 are forwarded to AP2. Since the RTT is large in this case, forwarded segments always arrive earlier than those taking the new route and, therefore, no out-of-order delivery occurs. That is, the handoff is completely transparent to the TCP session in this scenario.

The above experiments show that, without link-layer frame buffering-and-forwarding, either the TCP retransmission timeout or fast retransmit will be invoked during a link-layer handoff. This invocation of TCP congestion control unduely reduces the TCP congestion window and, consequently, the throughput. However, if the frame buffering-and-forwarding is applied, the link-layer handoff becomes transparent to the TCP (and upper-layer applications). That is, this link-layer frame buffering-and-forwarding helps an already-fast link-layer handoff become an error-free (or smooth) handoff. *Unfortunately, the above link-layer frame*

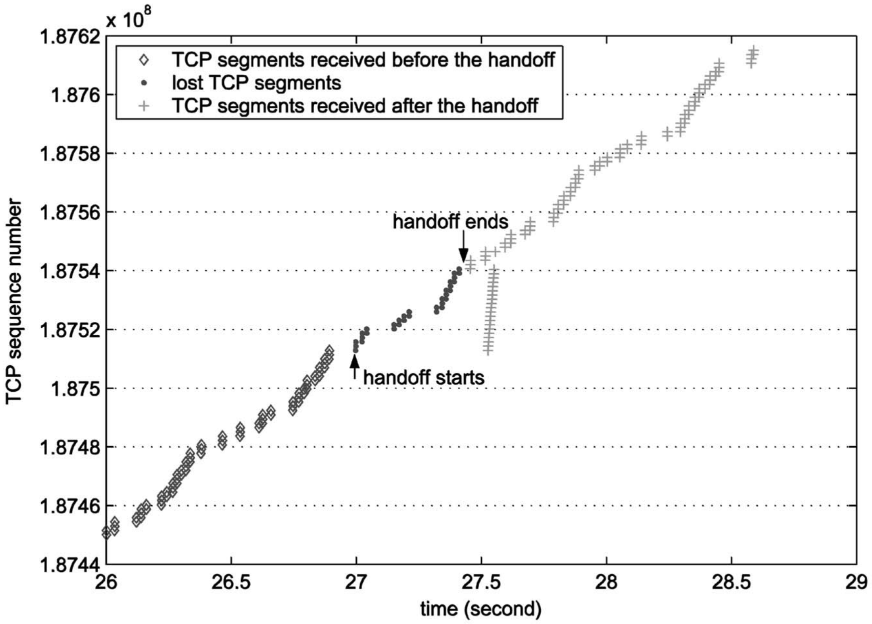


Fig. 5. TCP performance —Scenario II: large RTT without link-layer frame forwarding.

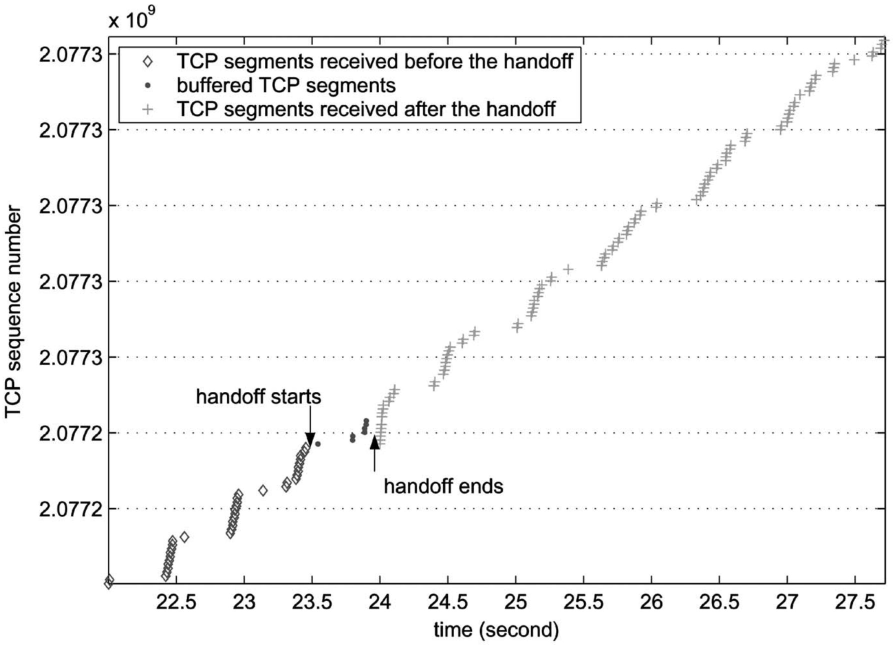


Fig. 6. TCP performance—Scenario II: large RTT with link-layer frame forwarding.

buffering-and-forwarding cannot make the fast IP-layer handoff schemes (which use the link-layer handoff indication) error-free because the APs involved in an IP-layer handoff do not reside in the same LAN segment as in our experiment. However, this problem can be solved by using the (enhanced) IAPP as we describe in the next section.

3 INTER-ACCESS POINT PROTOCOL (IAPP)

In order to better describe IAPP and the proposed enhancement, we first introduce some basic concepts of the IEEE 802.11 network architecture. The basic building block of an IEEE 802.11 network is the "basic service set" (BSS). Within a BSS, wireless stations (STAs) can communicate with each other and access the wired Internet via the STA serving as an AP of the BSS. Instead of being standalone, a BSS may also be a component of an extended form of network that is built with multiple BSSs. This extended form of network is called an "extended service set" (ESS) and the architectural component used to interconnect BSSs (to form an ESS) is the distribution system (DS). The relations among these components are illustrated in Fig. 7.

In a common DS, two STAs which cannot communicate directly with each other via the wireless medium can still communicate, as long as both STAs belong to the same ESS.

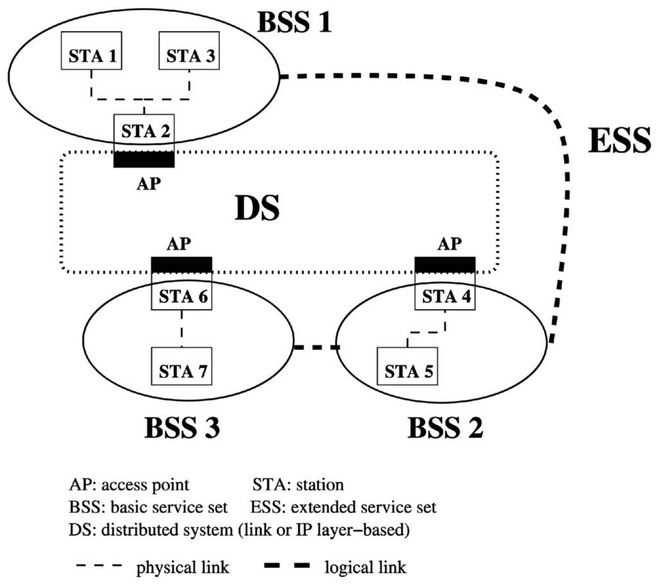


Fig. 7. The IEEE 802.11 wireless network architecture.

That is, an ESS conceptually appears the same to a logical link control layer as a BSS, but with a larger "coverage." The IEEE 802.11 standard does not require the DS to be link-layer-based or network-layer-based as long as the DS can distribute the data frames, using the provided information, to the correct "output" point that corresponds to the desired recipient. The information required by the DS can be obtained from the association-related control frames in the IEEE 802.11 standard.

3.1 Current IAPP

With the basic concepts introduced above, we can now discuss IAPP. Briefly, IAPP is a set of functionalities and a protocol used by an AP to communicate with other APs on a common DS. It is part of a communication system comprising APs, STAs, an arbitrarily connected DS, and Remote Authentication Dial In USER Service (RADIUS) servers [28]. The RADIUS servers provide two functions: 1) mapping the BSS Identification (BSSID) of an AP to its IP address on the DS and 2) distribution of keys to the APs to allow the encryption of the communications between the APs. The functions of the IAPP are to 1) facilitate the creation and maintenance of the ESS, 2) support the mobility of STAs, and 3) enable APs to enforce the requirement of a single association for each STA at a given time.

Among the functions provided by the IAPP, we focus on the IAPP's support for STAs' mobility. The events and frame exchanges followed right after a STA moves away from its current AP are illustrated in Fig. 8. First, the STA starts searching for a new AP by switching to different channels and seeking new beacon frames. If a new AP is located, the STA attempts to reassociate itself with this AP by sending a reassociation request. This request contains the STA's MAC address and the BSSID of the STA's previous AP. Upon receiving this reassociation request, the new AP replies to the STA with a reassociation response using the MAC address obtained in the received reassociation request. The new AP also sends an IAPP MOVE-notify to the old AP via the DS as required by the IAPP. The old AP then responds to the new AP with a MOVE-response which carries the context block for the STA's association.

The IAPP MOVE-notify and MOVE-response are IP packets carried in a TCP session between APs. The IP address of the old AP must be found by mapping the BSSID from the reassociation message to its IP address. This mapping is done using a RADIUS exchange and any standard RADIUS server that supports the CALL CHECK service-type should work.[3] Finally, a link-layer update frame is sent by the new AP so that any local layer-2 devices, such as bridges, switches, and other APs, can update their forwarding tables in order to forward frames to reach the new location of the STA through the correct port.

3.2 The Proposed IAPP Enhancement

With the MOVE-notify and MOVE-response packets, the IAPP can provide context transfer between APs. By using the context information (such as security information of a STA) carried by these packets, an AP can expedite the reauthentication of a mobile host on reassociation, thus reducing the link-layer handoff latency. However, there always exists a time period during which the STA cannot send or receive anything, and frames may get lost as shown in Fig. 8. That is, the problems demonstrated in Section 2 can still occur even though the APs are now able to communicate with each other via IAPP. To fix this problem, we include the technique in Section 2, namely, the link-layer frame buffering-and-forwarding, in the current IAPP. *Unlike the "link-local" frame buffering-and-forwarding in Section 2, the frame buffering-and-forwarding here will enable frame-forwarding not only between the APs in the same subnet but also between the APs in different subnets*. The proposed frame buffering-and-forwarding is illustrated in Fig. 9.

Each buffered link-layer frame (at the old AP) will be carried in our proposed new IAPP packet—called the IAPP **MOVE-forward**—and sent to the new AP, following the MOVE-response packet. Although a single MOVE-response packet may carry all buffered frames, it is more feasible to put the individual buffered frames to several Move-forward packets because 1) the resulting packet size may be too large if there are many buffered frames at the old AP and 2) in-flight data frames (to the old AP) cannot be forwarded to the new AP if they arrive at the old AP after the MOVE-response packet is sent. The format of the proposed MOVE-forward packet follows the general IAPP packet format and is depicted in Fig. 10. The "Command" field in the IAPP packet header identifies the specific function of the packet. For the IAPP MOVE-forward packet, one can choose any integer value between 7 and 255.[4] The "Data" field contains a subfield "MAC Address" which represents the MAC address of the STA initiating the reassociation request. This address can be obtained from the IAPP MOVE-notify

---

[3] It can also be done using locally configured information mapping the BSSID of APs to their IP-address on the DS.

[4] One through six are reserved for IAPP MOVE-notify, MOVE-response, etc.

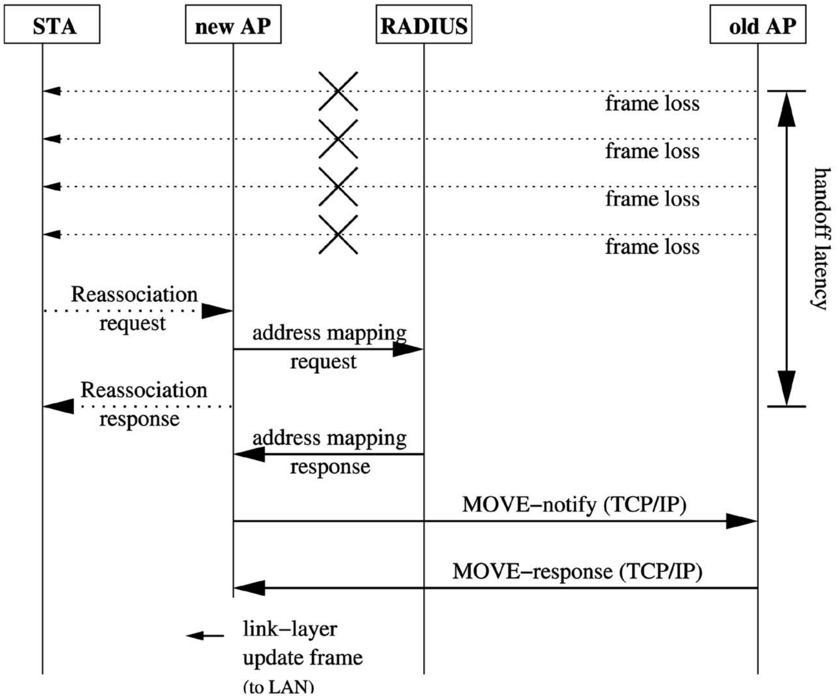


Fig. 8. The IAPP MOVE-notify and MOVE-response packet exchanges during a link-layer handoff.

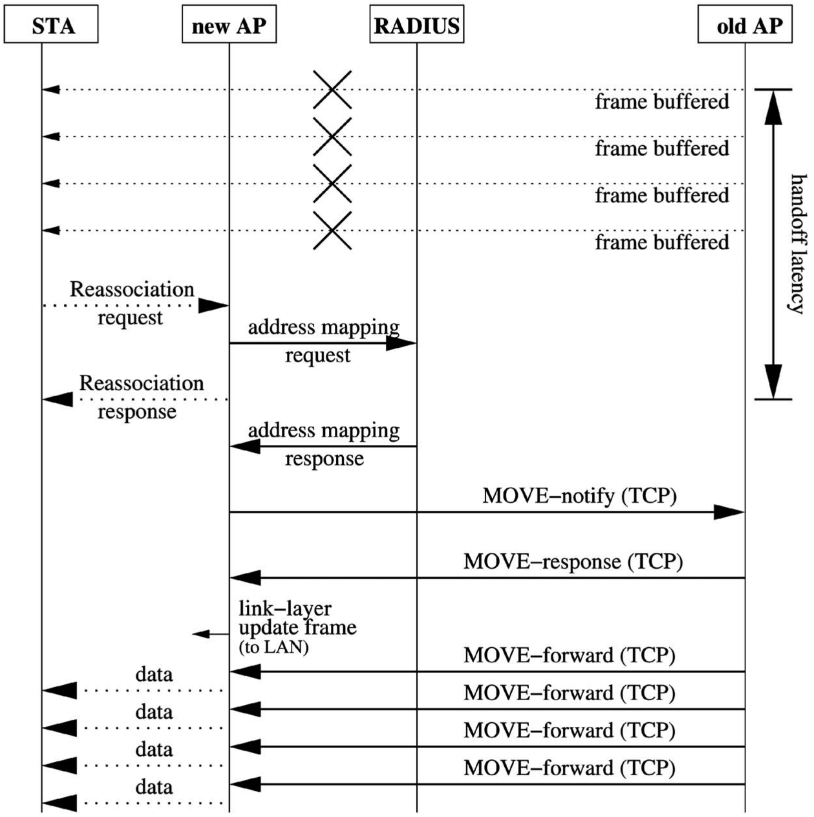


Fig. 9. The enhanced IAPP packet exchanges during a link-layer handoff: MOVE-notify/MOVE-response packets followed by MOVE-forward packets.

packet so the old AP can tell which frames to forward. The address is also used by the new AP to transmit the link-layer frame to its final recipient via the wireless link. The new AP retrieves the forwarded link-layer frame from the "Information" subfield of the "Context Block" in a received MOVE-forward packet. The retrieved link-layer frame is transmitted to the STA once the authentication or security association between the new AP and the STA is completed.

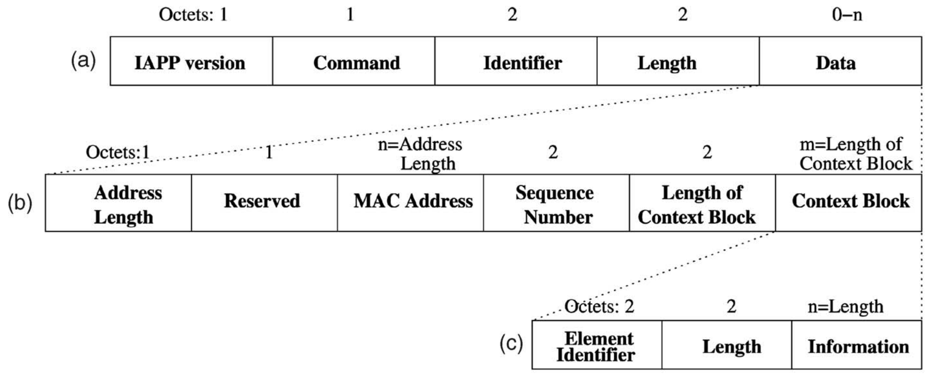


Fig. 10. IAPP MOVE-forward packet format: (a) General IAPP packet format, (b) MOVE-forward DATA field format, and (c) Information element format.

3.3 Improvements by Using the Proposed Enhancement

The enhanced IAPP not only improves the link-layer handoff as described in Section 2, but also the IP-layer handoff as follows:

1. A mobile host can receive forwarded link-layer frames (from the old AP) via the new AP even when this new AP resides in a different IP subnet because the IAPP is an IP-based protocol and the forwarded frames are transmitted via TCP/IP.
2. Because of item 1, if the mobile host moves to a new IP subnet, it can resume receiving data frames (via the IAPP MOVE-forward packets) even before the IP-layer handoff (e.g., the Mobile IP procedure) is initiated. From the mobile host's perspective, the IP-layer handoff latency is reduced to the level of the link-layer handoff latency as in those fast handoff schemes using link-layer handoff indications.
3. The APs function uniformly regardless of the type of handoffs they are involved with because the enhanced IAPP does not require differentiation between a link-layer and an IP-layer handoff for the purpose of frame forwarding. As a result, the intelligence of determining the handoff type in order to initiate a fast IP-layer handoff is no longer needed.
4. Because of items 1, 2, and 3, a fast and smooth IP-layer handoff is achieved "implicitly" (by the enhanced IAPP) without modifying Mobile IP since the mobility agents/access routers are not involved in frame buffering-and-forwarding. That is, a fast IP-layer handoff is achieved without coupling link-layer operations with the Mobile IP operations. Such independence makes the enhanced IAPP applicable to other protocols supporting IP mobility which may emerge in future.
5. The mobile host requires neither multiple radio interfaces nor a priori knowledge of the new AP it may head for, thanks to the "posthandoff" feature in the enhanced IAPP.
6. No additional *over-the-air* signaling is required, unlike other schemes, except the original reassociation frame in the IEEE 802.11 standard. Of course, the frame buffering-and-forwarding requires resources at both end APs and consumes network bandwidth along the path between them. However, the wired network is not the resource bottleneck and such resource requirement should be acceptable in order to achieve smooth handoffs.

3.4 Uniform Link and IP-layer Handoffs

We show via an example how the enhanced IAPP can actually achieve all of the above salient features. Let us consider the scenario shown in Fig. 1 and consider the case when a mobile host moves from AP1 to AP2, and eventually to AP3. As the mobile host is handed off to AP2, it sends a reassociation request to AP2 as required by the IEEE 802.11 standard. Once it receives the reassociation request from the mobile host, AP2 follows the procedure illustrated in Fig. 9: It sends a reassociation response to the mobile host and an IAPP MOVE-notify to AP1. In the meantime, AP1 buffers all link-layer frames destined for the mobile host (signaled by the frame retry count as we will detail later). Upon receiving the IAPP MOVE-notify from AP2, AP1 replies with an IAPP MOVE-response and forwards all buffered frames to AP2. Then, AP2 sends a link-layer update frame to the local subnet and transmits the frames received from AP1 to the mobile host via the wireless link. Since the link-layer update frame "refreshes" the local MAC bridge's forwarding table, the new link-layer frames (from the mobile node's corresponding node) will take the direct route to AP2. Under this scenario, the mobile host will soon receive the router advertisement from AR1 and realize that no IP-layer handoff is necessary.

Next, suppose that the mobile host loses its connection to AP2 at $t = t_1$ and tries to reassociate with AP3 as shown in Fig. 11. The mobile host and AP3 follow exactly the same procedures as above (since it is just a link-layer handoff so far) and the reassociation is completed at $t = t_2$. AP2 also reacts exactly the same as AP1 does during the first handoff. The only difference is that now the forwarded link-layer frames take a longer, cross-subnet path. However, this is perfectly fine since the APs communicate with each other using IAPP via the DS, which is an IP-based distribution system. Until this time instant, the mobile host (more precisely, its Mobile IP entity) has not detected the upcoming IP-layer handoff. It is until the mobile host misses two consecutive router advertisements from the old AR that the Mobile IP entity starts the Mobile IP binding update (at $t = t_4$). In the meantime, the buffered frames

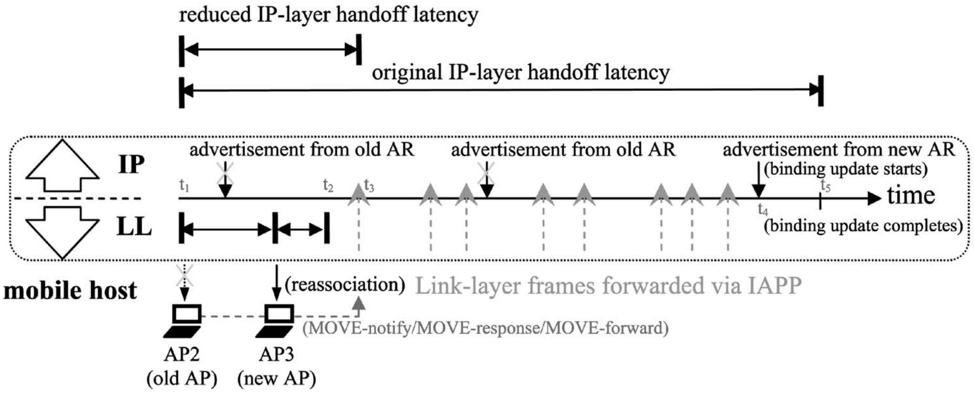


Fig. 11. Smooth and fast IP-layer handoffs by using the enhanced IAPP: 1) IP-layer handoff latency is reduced to the level of link-layer handoff latency and 2) packet losses during an intersubnet handoff are eliminated by the link-layer frame buffering-and-forwarding.

carried in the IAPP MOVE-forward packets start to arrive at the mobile host at $t = t_3$, along the route from AP2, via the MAC bridges and the routers, to AP3 as shown in Fig. 1. Since the mobile host restarts receiving the forwarded frames at $t = t_3$ instead of at $t = t_5$ when the Mobile IP binding update is completed, the IP-layer handoff latency is reduced significantly and is equal to that in the postregistration fast handoff schemes. Most importantly, the APs react uniformly to both intra and intersubnet handoffs, and Mobile IP is left intact.

4 SIMULATION AND EVALUATION

The IAPP and the proposed enhancement are implemented in the *Network Simulator-2* (ns-2) since at present there is no off-the-shelf wireless LAN card supporting the IAPP. In order to support handoffs in IEEE 802.11 wireless LANs, the operations for beaconing, association, and reassociation are also implemented in the ns-2. Without giving too much of the implementation details, we list the essential operations in the AP and the mobile host for supporting the enhanced IAPP. Especially, we describe how the AP gets signaling of frame buffering based on the existing IEEE 802.11 standard.

4.1 Operations of APs

Since an AP works differently depending on whether it is acting as an old AP or a new AP for the mobile host, we separate discussions of the AP's operations accordingly.

4.1.1 Old AP

The most important tasks of an old AP are to 1) buffer the data frames destined for the mobile host once it lost the connection with the mobile host and 2) forward the frames after it is informed by another AP about the mobile's handoff. For frame buffering, an old AP needs some signaling or indications to initiate frame-buffering. Although the IEEE 802.11 standard defines the disassociation procedure between an AP and a mobile host, using disassociation frames as the signaling is not reliable because the disassociation frame may never reach the old AP once the mobile host loses the link-layer connection.[5] In our implementation, we use the frame retry count as the signaling for frame buffering.

In the IEEE 802.11 wireless LAN, a frame can be retransmitted up to *retry count limit* (= 7) times before it is discarded. If the old AP has retransmitted a frame seven times, it is a strong indication that the mobile host may have moved out of its coverage area. Of course, the frame may happen to collide with others, but the probability that a frame collides with others seven consecutive times is extremely small due primarily to the exponential random backoff in the IEEE 802.11 standard. Another possibility of consecutive frame retransmissions is that the mobile host suffers a poor reception due to multipath fading. We handle this situation as follows:

1. An AP buffers any frame which is supposed to be discarded based on the IEEE 802.11 standard (that is, any frame with the retry count exceeding *retry count limit*). The AP also starts a timer which expires 500 msecs after the first frame is buffered.
2. Whenever a frame from the mobile host is received, the AP discards all buffered frames[6] and stops the timer.
3. If the timer expires but the AP does not receive an IAPP Move-notify packet from other APs, the AP discards all buffered frames and stops the timer.
4. If the AP receives an IAPP Move-notify packet regarding a mobile host whose MAC address matches the destination MAC address of a buffered frame, the matched frame is forwarded and the timer is stopped. Moreover, the AP sets a *forwarding flag* associated with the mobile host to TRUE so that in-flight frames destined for the mobile host will also be forwarded once they arrive at the old AP.

By following the above procedure, the old AP can accurately buffer the frames for the mobile host during a link-layer handoff. One should note that all of these operations (in the old AP) are at the MAC layer as required by the IEEE 802.11 standard, except the operations involved with other APs (including MOVE-notify, MOVE-response and MOVE-forward), which are regulated by IAPP and our enhancement.

---

[5] Most existing IEEE 802.11 wireless LANs do not support disassociation between APs and mobile hosts via the wireless link.

[6] For better performance, the AP can send the buffered frames to the mobile host but this is out of the scope of a handoff.

4.1.2 New AP

The new AP follows the procedure as we explained in the previous section. In addition, the new AP will:

1. Set the *forwarding flag* associated with the mobile host to FALSE—if such a flag had been set to TRUE before—once the AP completes the reassociation process of the mobile host. This way, the new AP can stop any frame forwarding that may have been activated for the mobile host when last time the mobile host was handed off from this AP.
2. Check the list of associated mobile hosts for every received MOVE-forward packet. If the MAC address contained in the IAPP header of the MOVE-forward packet matches any one of the mobile hosts in the list, the new AP retrieves the link-layer frame from the received MOVE-forward packet, and transmits it to that MAC address via the wireless link immediately. Otherwise, the new AP discards the received MOVE-forward packet.

4.2 Operation of a Mobile Host

The mobile host follows the normal reassociation procedures defined in the IEEE 802.11 standard during a link-layer handoff. In addition, the mobile host also follows the following procedure:

1. The mobile host buffers any frame which is supposed to be discarded based on the IEEE 802.11 standard (that is, the frame with the retry count exceeding *retry count limit*). The mobile host also starts a timer which expires 500 msecs after the first frame is buffered.
2. Whenever a frame from the current AP is received, the mobile host discards all buffered frames[7] and stops the timer.
3. If the timer expires but the mobile host does not receive any beacon frame from other APs, the mobile host discards all buffered frames and stops the timer.
4. If a new beacon frame is received before the timer expires, the mobile host stops the timer and transmits the buffered frame via the new AP once the reassociation with the new AP is completed.

By following this procedure, the mobile host can prevent any uplink (from the mobile host to the AP) frame loss during a handoff. As a result, both uplink and downlink transmissions are error-free during both intra and intersubnet handoffs.

4.3 Simulation Results

The network topology used throughout the simulation is shown in Fig. 12. All APs in the figure are the IEEE 802.11 APs. AP1 and AP2 reside in an IP subnet and are connected by a MAC bridge, while AP3 and AP4 reside in another IP subnet and are also connected by a MAC bridge. The purpose of using the MAC bridges is to separate the APs in the same IP subnet so that they are in two different "segments." This way, we can capture the effects of link-layer update frame (in the IAPP protocol) on an intrasubnet handoff process. In order to better monitor the mobile host's handoffs, we choose the transmission power and receiving power threshold in a way that the mobile host loses its connection to both APs when it is in the middle of the two APs, which are separated by 40m.

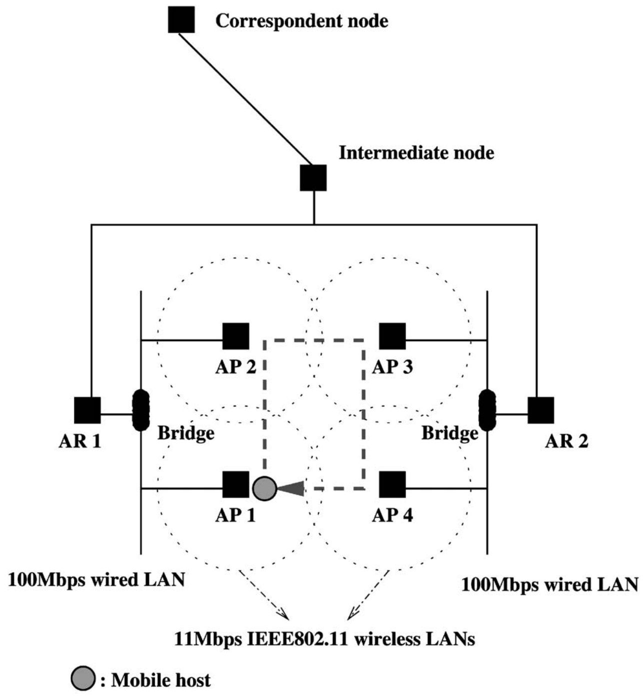


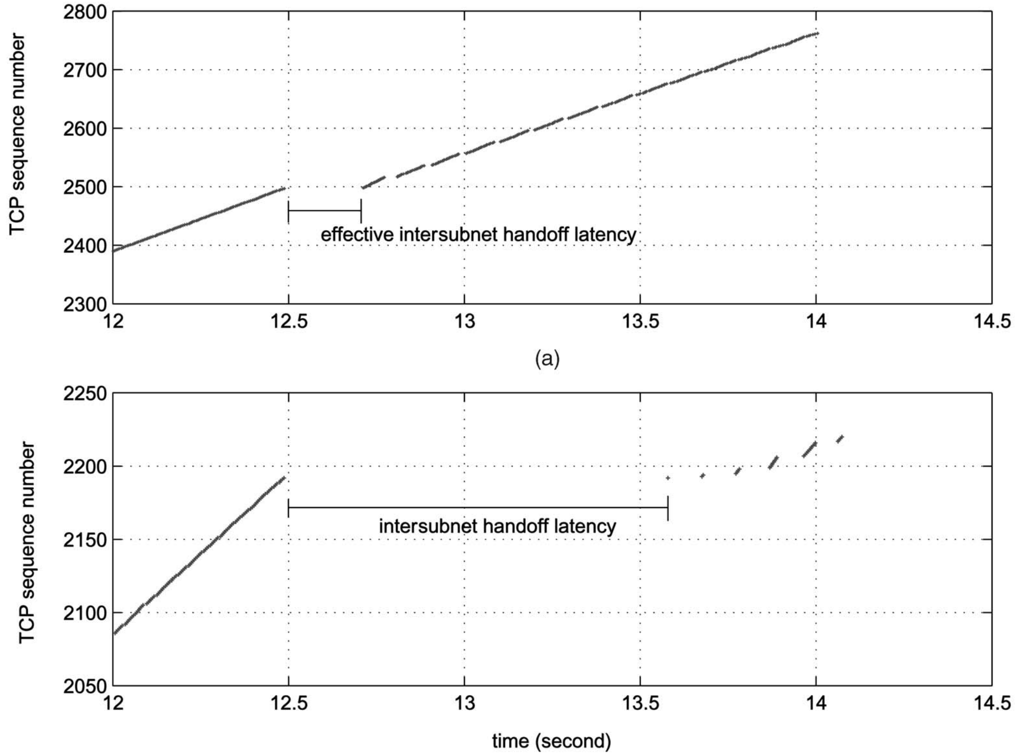
Fig. 12. Network topology in the *ns-2* simulation.

The mobile host in the figure follows a very simple movement pattern. The mobile host starts at AP1 and heads toward AP2 at a fixed speed S. Once reaching AP2, the mobile host turns right and heads toward AP3 with the same speed. The mobile host repeats the same movement after it arrives AP3, then AP4 and eventually AP1. After that, the mobile host starts all over again. This way, the mobile host will experience two intrasubnet handoffs (between AP1 and AP2, and between AP3 and AP4) and two intersubnet handoffs (between AP2 and AP3, and between AP4 and AP1). For each intersubnet handoff, the mobile host has to perform a link-layer handoff (between the APs) and an IP-layer handoff (between the ARs).

In order to initiate a handoff, a mobile host needs to seek a new beacon frame (for a link-layer handoff) or a router advertisement (for an IP-layer handoff) after waiting for some time and still receiving no beacon or advertisement from the current AP or AR. This waiting time is usually chosen to be multiple beacon frame intervals for a link-layer handoff or multiple router advertisement intervals for an IP-layer handoff. Of course, one can choose a waiting time equal to a beacon/advertisement interval to expedite a handoff. However, the mobile host may miss a beacon/advertisement simply because of a transmission error or a collision. Therefore, choosing too small a waiting time may force a mobile host to switch to other radio channels for seeking new beacons/advertisements which may be unnecessary in the

---

[7] For better performance, the mobile host can send the buffered frames to the current AP but this is out of the scope of a handoff.



(a)

(b)

Fig. 13. Reduced IP-layer handoff latency as compared to the original Mobile IP-only scheme.

first place. That is, the beacon/advertisement waiting time creates a trade-off between the handoff latency and accuracy of initiating a handoff process. Since the link-layer handoff latency is relatively small (usually hundreds of milliseconds), we choose the beacon waiting time to be twice of the beacon interval (= 100 msecs) to prevent any "premature" channel switching. For the router advertisement waiting time, we consider the value of a single router advertisement interval (= 1 second) and twice of the interval (= 2 seconds).

Finally, we use the TCP-based application as the traffic source in our simulation. The mobile host and its correspondent node establish a FTP session with an approximated end-to-end throughput of 2.4 Mbps, based on the chosen packet size (= 1,500 bytes), average round-trip time (≈ 100 msecs), and the maximal TCP congestion window size (20). In what follows, we show how the enhanced IAPP improves the performance in terms of handoff latency and overall TCP throughput, and investigate the impacts of user mobility and router-advertisement waiting time on these improvements.

4.3.1 Reduced IP-layer Handoff Latency

Since we have already shown the effects of link-layer frame buffering-and-forwarding on intrasubnet handoffs in Section 3, we now focus on the intersubnet handoff in this section. The trace of TCP sequence numbers (in the mobile host side) under the enhanced IAPP is plotted in Fig. 13a. Here, we only show an intersubnet handoff between AP2 and AP3 around $t = 12$ seconds. At $t = 12.48$ seconds, the mobile host loses its connection with AP2 when it is heading for AP3. However, the mobile host has not detected the situation since it just received a beacon frame from AP2 at $t = 12.4$ seconds and believes it is still connected. It is until $t = 12.62$ seconds that the mobile host starts seeking new beacon frames because the beacon-frame waiting time has expired (200 milliseconds in our simulation). At $t = 12.7$ seconds, the mobile host receives a new beacon frame from AP3 and attempts to reassociate with AP3. After the reassociation is completed, the mobile host starts to receive forwarded TCP segments from AP2 via AP3 (note that it is a batch of 20 segments). It should be noted that at this time point, the mobile host has not discovered that it has moved to a new IP subnet yet. It is until $t = 13.4$ seconds that the mobile host receives a router advertisement from AR2 (via AP3), and then starts the binding update. Once the binding update is completed, the TCP segments will take the new route instead of being forwarded by AP2. In this scenario, the "effective" intrasubnet handoff latency is equal to the link-layer handoff latency, which is around 210 milliseconds in our simulation.

Fig. 13b shows the same scenario as above except that we use the original IAPP. As in the previous case, the link-layer handoff process is completed around $t = 12.7$ seconds. However, without frame buffering-and-forwarding, the mobile host receives nothing from the correspondent node until the TCP segment #2192 times out at $t = 13.52$ seconds (note the exponential increase of TCP congestion window size thereafter). Unfortunately, the TCP retransmission timeout reduces the correspondent node's TCP congestion window size, hence reducing the throughput. We will investigate this issue in the next section. In regard to the

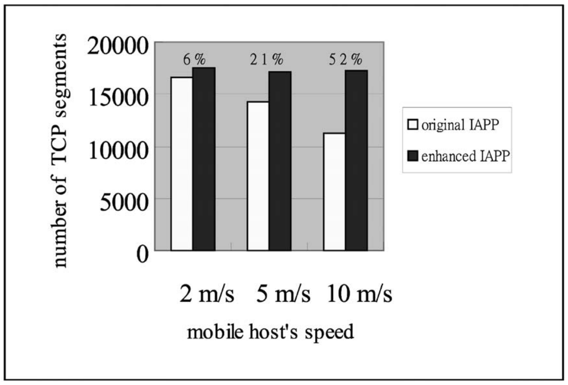


Fig. 14. Throughput improvement made by the enhanced IAPP under different user mobility.

handoff latency, the resulting intersubnet handoff latency is around 1 second, which is 790 milliseconds more than that of using the enhanced IAPP. Of course, the intersubnet handoff latency also depends on the router-advertisement waiting time. So far, we use the minimal waiting time (equal to the router advertisement interval). One can expect an even longer intersubnet handoff latency (without using the enhanced IAPP) if we allow the use of a longer router-advertisement waiting time. We will also discuss this issue in the following simulations.

4.3.2 User Mobility

Based on the mobility pattern described in the beginning of this section, we choose three different speeds for the mobile host: $S = 2m/s$, $S = 5m/s$, and $S = 10m/s$. These three different speeds represent *low-mobility*, *medium-mobility*, and *high-mobility*, respectively. We set the router-advertisement waiting time as a router-advertisement interval, which is the minimal value one can choose. This way, the mobile host is more "agile" in seeking new router advertisements and initiating a handoff process.

Fig. 14 shows the number of TCP segments received by the mobile host in an 85-second time interval (so that a mobile host can visit all APs at a speed of 2 m/s) at different speeds. For each speed, we use the original IAPP and the enhanced IAPP for comparative purposes. As shown in the figure, the mobile host receives more TCP segments at all three speeds if the enhanced IAPP is used. These improvements originate from the fact that neither the TCP fast retransmit nor retransmission timeout is invoked, thanks to the loss-free, much faster handoff process enabled by the enhanced IAPP. In contrast, the TCP fast retransmit may occur during an intrasubnet handoff and the TCP may time out during an intersubnet handoff, if the original IAPP is used.

The improvement percentages (over the original IAPP) are also shown in the figure indicating that the higher the user mobility, the larger the improvement. This is because when the mobile host moves fast, the less time it stays within the coverage of an AP and, therefore, the larger the percentage of time the mobile host spends on a handoff. For example, at a speed of 10 m/s, the number of TCP segments sent by using the enhanced IAPP is 52 percent more than that by using the original IAPP. In fact, the simulation result can be verified as follows: During an intersubnet handoff, a retransmission timeout will occur if the original IAPP is used. As a result, the lost TCP segment will be retransmitted 1 second after the handoff starts (as shown in Fig. 13) with TCP congestion window reset to 1. Then, the window size increases exponentially until it exceeds the new threshold (i.e., 10, given the window size is 20 before a handoff starts) and increases linearly after that until reaching the maximum value (i.e., 20 in our simulation). The evolution of TCP congestion window size is shown in Fig. 15. By computing the area under the curve of the figure, one can calculate the total number of TCP segments sent after an intersubnet handoff. For example, at a speed of 2 m/s, the mobile host's "cell" sojourn time is $T_{sojourn} = \frac{40m}{2m/s} = 20$ secs, giving a total of $(15 + 165 + 3,500) = 3,680$ TCP segments.[8] The number of TCP segments sent after an intrasubnet handoff can be calculated similarly except that the TCP fast-retransmit, instead of TCP time-out, will occur thanks to the link-layer update frame. At a speed of 2 m/s, the total number of TCP segments sent after an intrasubnet handoff is around 3,800, if the original IAPP is used. Compared to the case when the enhanced IAPP is used, the total number of TCP segments sent after a handoff is always $3,960 = 20 * \frac{20-0.2}{0.1}$. This is because the TCP resumes right after the link-layer handoff is completed and, therefore, no reduction of TCP congestion window size occurs. The "theoretical" improvement percentage can be obtained as

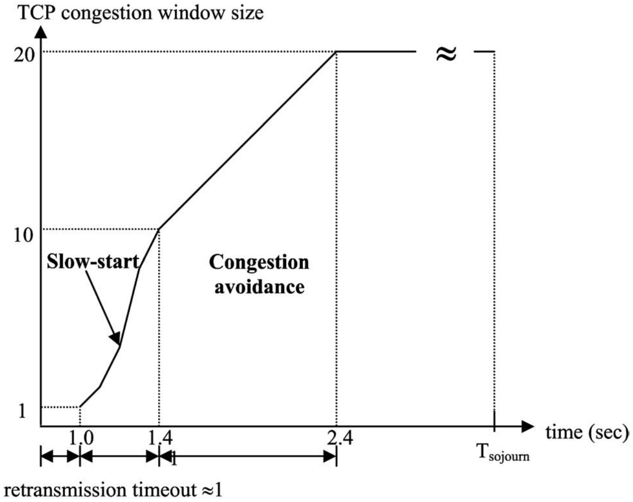


Fig. 15. The evolution of TCP congestion window during an intersubnet handoff if using the original IAPP.

$$\text{improvement} = \frac{3,960 * 4 - (3,680 * 2 + 3,800 * 2)}{3,680 * 2 + 3,800 * 2} = 5.88\%,$$

where we have four handoffs—two intra and two intersubnet handoffs—within an 85-second time period. This result matches the simulation result very well (6 percent in our simulation), and one can use the same approach for the cases of medium and high-mobility as well.

The improvement also depends on the router-advertisement waiting time used by a mobile host. In the simulation, we use the smallest value (= 1 second) given that the router-

---

[8] These three numbers represent the number of TCP segments sent within the three phases shown in Fig. 15.

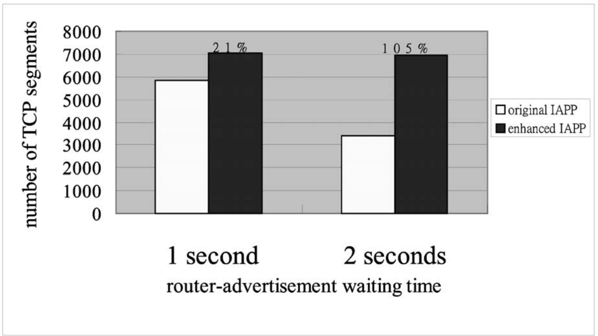


Fig. 16. Throughput improvement made by the enhanced IAPP for different Mobile IP router-advertisement waiting times.

advertisement interval is 1 second as suggested in the Mobile IP standard. One can expect that if a larger waiting time is used, the transmission of a mobile host will stall longer, under the original IAPP, due to the longer interhandoff latency. In contrast, the transmission of a mobile host is not affected by the value of router-advertisement waiting time under the enhanced IAPP as we will show next.

4.3.3 Router-Advertisement Waiting Time

As mentioned earlier, there exists a trade-off between the handoff latency and accuracy of initiating a handoff process. Although choosing a small router-advertisement waiting time can reduce an intersubnet handoff latency, doing so may sometimes invoke movement-detection operations which should not take place at all, hence incurring control overhead. For example, a mobile host may simply miss a router advertisement due to transmission errors. To investigate the impact of this waiting time on the handoff performance, we consider both 1-second and 2-second waiting times. A 2-second waiting time allows a mobile host to miss one router advertisement without trying to initiate an intersubnet handoff. In the original Mobile IP standard, the waiting time should not exceed 3 seconds (that is, allowing a mobile host to miss two consecutive router advertisements).

The number of TCP segments received by the mobile host is shown in Fig. 16 for both waiting times under the original IAPP and the enhanced IAPP. On can observe that the mobile host receives 42 percent ($\approx \frac{60-35}{60}$) less segments if a larger waiting time under the original IAPP is used. This is because the larger waiting time suffices to cause two consecutive TCP retransmission timeouts during an inter-handoff latency. As illustrated in Fig. 17, a TCP segment times out within around 1 second after its first unsuccessful transmission, due to an intersubnet handoff under the original IAPP. The segement is retransmitted, and will still get lost if a 2-second router-advertisement waiting time is used, mainly because it takes up to 2 seconds for a mobile host to perceive the movement and to initiate the Mobile IP binding update. After the second transmission failures, the TCP will double the retransmission timeout. As a result, the second retransmission of the lost TCP segment will not start even though the intersubnet handoff is completed before the retransmission timer expires, which degrades the TCP performance.

However, a TCP retransmission timeout does not occur under the enhanced IAPP because the TCP segments will be forwarded to the mobile host right after a link-layer handoff is completed. Since the link-layer handoff is independent from the Mobile IP operations (i.e., the router advertisements), the TCP performance is not affected by the router-advertisement waiting time as also shown in Fig. 16. Compared to the TCP performance under the original IAPP, the improvements achieved by using the proposed enhancement are 21 percent and 105 percent for the cases of 1-second and 2-second waiting times, respectively, under our simulation setting.

Based on the simulation results, we can conclude that the enhanced IAPP allows the use of a larger router-advertisement waiting without sacrificing the TCP performance or increasing intersubnet handoff latency. In other words, the enhanced IAPP optimizes the aforementioned trade off between the handoff latency and accuracy of initiating a handoff process caused by the router-advertisement waiting time.

5 CONCLUSION

In this paper, we proposed a simple but effective enhancement for IAPP to improve both intrasubnet and intersubnet handoff processes. This enhancement relies on the AP's capability of interoperation with other APs, provided by the IP-based IAPP, and enables intersubnet frame buffering-and-forwarding (between APs) for the mobile host via a distribution system (DS).

The enhanced IAPP reduces the intersubnet handoff latency significantly without requiring any modification to Mobile IP. Unlike other existing schemes which require a Mobile IP entity to process link-layer handoff indications, our enhanced IAPP decouples the Mobile IP operations from the underlying link-layer handoff process. Such decoupling (or independence) makes the enhanced IAPP applicable to other IP-mobility solutions.

We conducted the *ns*-2-based simulation to study the TCP performance under the enhanced IAPP. The simulation results show that the enhanced IAPP supports high user mobility. Unlike other schemes which require user intervention, the enhanced IAPP performs a fast handoff automatically by means of the enhanced IAPP frame

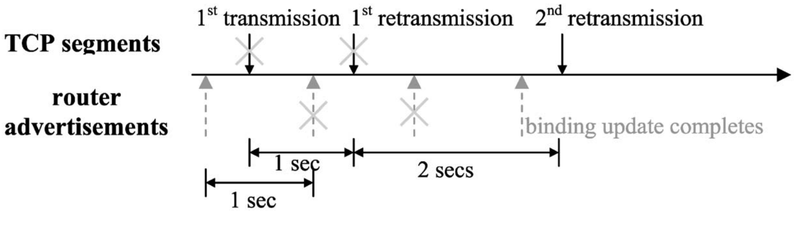


Fig. 17. Consecutive TCP retransmission timeouts if using a 2-second advertisement-waiting time.

buffering-and-forwarding. The results also show that the enhanced IAPP allows the Mobile IP to use a less aggressive movement detection, thus reducing the associated control overhead.

As future work, we would like to study the performance of UDP-based applications such as audio or video streaming under the proposed enhanced IAPP.

ACKNOWLEDGMENTS

The work reported in this paper was supported in part by the US AFOSR under Grant No. F49620-00-1-0327.

REFERENCES

[1] "IEEE 802.11b: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Standard,* 1999.

[2] M. Portoles, "IEEE 802.11 Link-Layer Forwarding for Smooth Handoff," *Proc. 14th IEEE Int'l Symp. Personal, Indoor, and Mobile Radio Comm. (PIMRC '03),* pp. 1420-1424, 2003.

[3] A. Mishra, A Shin, and W. Arbaugh, "An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process," *ACM SIGCOMM Computer Comm. Rev.,* vol. 33, no. 2, Apr. 2003.

[4] C. Perkins et al., "IP Mobility Support," *RFC 2002,* Oct. 1996.

[5] D. Johnson and C. Perkins, "Mobility Support in IPv6," draft-ietf-mobileip-ipv6-17.txt, a work in progress.

[6] J.C. Wu, C. Cheng, and N. Hunag, "Intelligent Handoff for Mobile Wireless Internet," *Mobile Networks and Applications,* no. 6, pp. 67-79, 2001.

[7] K.E. Malki et al., "Low Latency Handoff in Mobile IPv4," draft-ietf-mobileip-lowlatency-handoffs-v4-02.txt, a work in progress.

[8] G. Dommetry et al., "Fast Handovers for Mobile IPv6," draft-ietf-mobile-fast-mipv6-04.txt, a work in progress.

[9] A.E. Yegin et al., "Supporting Optimized Handover for IP Mobility-Requirements for Underlying System," draft-manyfolks-12-mobilereq-02.txt, a work in progress.

[10] IEEE 802.21: A Generalized Model for Link Layer Triggers (an abstract).

[11] J. Choi, "Detecting Network Attachment in IPv6 Problem Statement," draft-jinchoi-dna-dnav6-prob-00.txt, a work in progress.

[12] M. Ohta, "Smooth Handover over IEEE 802.11 Wireless LAN," draft-ohta-smooth-handover-wireless LAN-00.txt, a work in progress.

[13] J. Kempf et al., "Bidirectional Edge Tunnel Handover for IPv6," IETF draft: draft-kempf-beth-ipv6-02.txt, Mar. 2002.

[14] T. Narten, "Neighber Discovery for IPv6," *RFC 2461,* Dec. 1998.

[15] P. Tan, "Recommendations for Achieving Seamless IPv6 Handover in IEEE 802.11 Networks," draft-paultan-seamless-ipv6-handoff-802-00.txt, a work in progress.

[16] Y. Gwon et al., "Fast Handoffs in Wireless LAN Networks Using Mobile Initiated Tunneling Handoff Protocol for IPv4 (MITHv4)," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC '03),* pp. 1248-1253, 2003.

[17] T. Yokota et al., "Link Layer Assisted Mobile IP Fast Handoff Method over Wireless LAn Networks," *Proc. ACM Mobicom '02 Conf.,* 2002.

[18] C.E. Perkins and K. Wang, "Optimized Smooth Handoffs in MobileIP," *Proc. IEEE Int'l Symp. Computers and Comm.,* pp. 340-346, 1999.

[19] R. Hsieh, Z.G. Zhou, and A. Seneviratne, "S-MIP: A Seamless Handoff Architecture for Mobile IP," *Proc. IEEE INFOCOM '03 Conf.,* vol. 3, pp. 1774-1784, 2003.

[20] X. Jiang and I.E. Akyildiz, "A Distributed Dynamic Regional Location Management Scheme for Mobile IP," *Proc. IEEE INFOCOM '02 Conf.,* vol. 2, pp. 1069-1078, 2002.

[21] R. Ramjee et al., "HAWAII: A Domain-Based Approach for Supporting Mobility in Wide-Area Wireless Networks," *IEEE/ACM Trans. Networking,* vol. 10, no. 3, pp. 396-410, June 2002.

[22] J. Cao et al., "Design and Performance Evaluation of an Improved Mobile IP," *Proc. IEEE INFOCOM '04 Conf.,* vol. 1, pp. 319-329, 2004.

[23] S. Sharma, N. Zhu, and T. Chiueh, "Low-Latency Mobile IP Handoff for Infrastructure-Mode Wireless LANs," *IEEE J. Selected Areas in Comm.,* vol. 22, no. 4, pp. 643-652, May 2004.

[24] H. Balakrishnan, S. Seshan, and R.H. Katz, "Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks," *Wireless Networks,* vol. 1, no. 6, pp. 469-481, 1995.

[25] S. Ohzahata, S. Kimura, and Y. Ebihara, "A Proposal of Seamless Handoff Method for Celluar Internet Environments," *IEICE Trans. Comm.,* vol. E84-B, no. 4, pp. 752-759, Apr. 2001.

[26] E. Shim, H.-y. Wei, Y. Chang, and R.D. Gitlin, "Low Latency Handoff for Wireless IP QoS with NeighborCasting," *Proc. IEEE Int'l Conf. Comm. (ICC '02),* vol. 5, pp. 3245-3249. 2002.

[27] "IEEE 802.11f: Recommended Practice for Multi-Vender Access Point Interoperability via an Inter-Access Point Protocol Access Dostribution Systems Supporting IEEE 802.11 Operation," *IEEE Standard 802.11f/D1,* Jan. 2002 (draft).

[28] C. Rigney et al., "Remote Authentication Dial In User Service (RADIUS)," *RFC 2058,* Jan. 1997.

[29] J. Malinen, "Host AP Driver for Intersit Prism2/2.5/3," http://hostap.epitest.fi/, 2005.

[30] P. Sarolahti and A. Kuznetsov, "Congestion Control in Linux TCP," *Proc. 2002 USENIX Ann. Technical Conf.,* June 2002.

Chun-Ting Chou received both the BS and MS degrees in electrical engineering from National Taiwan University, Taipei, Taiwan in 1995 and 1997, respectively. He received the PhD degree in electrical engineering and computer science from the University of Michigan, Ann Arbor, in 2004. He is currently a senior member research staff at Philips Research USA, Briarcliff Manor, New York. His current research interests include quality of service (QoS) in wireless and mobile networks, user mobility management, MAC protocol in MBOA Ultra-Wide Band (UWB), and spectrum agile and cooperative communication.

Kang G. Shin received the BS degree in electronics engineering from Seoul National University, Seoul, Korea, in 1970 and both the MS and PhD degrees in electrical engineering from Cornell University, Ithaca, New York in 1976 and 1978, respectively. He is the Kevin and Nancy O'Connor Professor of Computer Science and founding firector of the Real-Time Computing Laboratory in the Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, Michigan. His current research focuses on QoS-sensitive networking and computing as well as on embedded real-time OS, middleware, and applications, all with emphasis on timeliness and dependability. He has supervised the completion of 51 PhD theses and authored/coauthored around 600 technical papers and numerous book chapters in the areas of distributed real-time computing and control, computer networking, fault-tolerant computing, and intelligent manufacturing. He has coauthored (jointly with C.M. Krishna) a textbook *Real-Time Systems* (McGraw Hill, 1997). He has received a number of best paper awards and has also coauthored papers with his students which received the best student paper awards. He has also received several institutional awards. From 1978 to 1982, he was on the faculty of Rensselaer Polytechnic Institute, Troy, New York. He has held visiting positions at several universities and in industry. He also chaired the Computer Science and Engineering Division, EECS Department, The University of Michigan for three years beginning January 1991. He is fellow of the IEEE and the ACM, and member of the Korean Academy of Engineering, and is currently or has served as general chair and program chair for several conferences and symposiums. He was guest editor of the 1987 August special issue of the *IEEE Transactions on Computers* on real-time systems, an editor of the *IEEE Transactions on Parallel and Distributed Computing*, and an area editor of the *International Journal of Time-Critical Computing Systems*, *Computer Networks*, and *ACM Transactions on Embedded Systems*. He was a distinguished visitor of the IEEE Computer Society.