# A Seamless and Proactive End-to-End Mobility Solution for Roaming Across Heterogeneous Wireless Networks

Chuanxiong Guo, *Member, IEEE*, Zihua Guo, *Member, IEEE*, Qian Zhang, *Member, IEEE*, and Wenwu Zhu, *Senior Member, IEEE*

***Abstract*—Roaming across heterogeneous wireless networks such as wireless wide area network (WWAN) and wireless local area network (WLAN) poses considerable challenges, as it is usually difficult to maintain the existing connections and guarantee the necessary quality-of-service. This paper proposes a novel seamless and proactive end-to-end mobility management system, which can maintain the connections based on the end-to-end principle by incorporating an intelligent network status detection mechanism. The proposed system consists of two components, connection manager (CM) and virtual connectivity (VC). The CM, by using novel media access control-layer and physical-layer sensing techniques, can obtain accurate network condition, while at the same time reducing the unnecessary handoff and ping-pong effect. The VC can make mobility transparent to applications without additional network-layer infrastructure support using a local connection translation, and can handle mobility well in the network address translator and simultaneous movement cases using a subscription/notification service. The proposed system enjoys several unique advantages: 1) capable of reacting to roaming events proactively and accurately; 2) maintaining the connection's continuity with small handoff delay; and 3) being a unified end-to-end approach for both IPv4 and IPv6 networks. We have built a prototype system and performed experiments to demonstrate the advantages of the proposed system.**

***Index Terms*—Mobility management, network condition detection, wireless wide area network (WWAN)/wireless local area network (WLAN) handoff.**

## I. Introduction

With rapid development and deployment of wireless technologies, future wireless Internet is expected to consist of different types of wireless networks with different access bandwidth and coverage range. For instance, wireless local area networks (WLANs) can provide high-speed Internet access at limited places (i.e., hot spots), whereas cellular networks can offer universal network access but with limited access rate. A natural trend is, therefore, to utilize high-bandwidth provided by WLANs such as IEEE 802.11 and the universal coverage of the wireless wide area networks (WWANs) such as General Packet Radio Service (GPRS)/Universal Mobile Telecommunications System (UMTS) networks. It, therefore, requires an efficient support for switching between different wireless networks. We refer to such a switch procedure as *internetwork roaming* or *vertical handoff*. One of desired features for such heterogeneous wireless networks is, therefore, how to support seamless and proactive global roaming. Here, by "seamless," we mean that upper layer applications should be unaware of the roaming events and the related roaming procedure. By "proactive," we mean that accurate access network conditions and user preferences should be obtained in advance for accurate roaming decision. The seamless and proactive features can enable mobile users to move freely across heterogeneous wireless networks with the needed quality-of-service (QoS) requirement.

There has been several works focusing on vertical handoff support. In [1], Gustafsson *et al.* proposed the "always best connected (ABC)" concept that offered connectivity over multiple-access technologies to enhance user experience. This work provided the framework of ABC; but mobility management was not discussed. In [2]–[4], several experimental systems were reported, in which handoff decision and connection maintenance issues were treated separately and the core of such systems is the mobility management scheme, which can maintain users' connections after a vertical handoff occurred. However, in these systems, proactive network condition detection and seamless mobility management were not addressed. We proposed a mobility management scheme supporting WWAN and WLAN roaming in [5]; however, technical details such as network bandwidth and delay estimation, as well as end-to-end mobility management were not fully addressed.

In order to achieve seamless and proactive roaming, two key issues have to be addressed: network condition detection for vertical handoff decision and connection maintenance. Moreover, the network condition detection scheme and the connection maintenance scheme need to be tightly integrated to form a mobility system for roaming across heterogeneous wireless networks. To achieve proactive vertical handoff, the network condition should always be obtainable by an appropriate handoff metric. In heterogeneous wireless network environment, this is challenging as there does not exist comparable signal strength at the physical-layer to be utilized as vertical handoff decision metrics due to the different physical techniques. This is further complicated by the fact that different wireless access technologies offer different QoS parameters such as available access bandwidth and access delay, which could be difficult to obtain. To achieve seamless mobility management, small handoff delay is critical and the mobility event should be made transparent to applications. Since minimized handoff delay can be

Manuscript received June 7, 2003; revised December 16, 2003.

The authors are with the Wireless and Networking Group, Microsoft Research Asia, Beijing 100080, China (e-mail: t-chuguo@microsoft.com; zhguo@microsoft.com; qianz@microsoft.com; wwzhu@microsoft.com).

Digital Object Identifier 10.1109/JSAC.2004.826921

achieved if the mobility management procedure is performed from end-to-end between two communication peers, thus a natural question arises is whether it is possible to design a mobility management scheme that resolves the mobility issue by end hosts themselves using the end-to-end principle (i.e., keep the network as it is, enhance the end systems instead).

To address the aforementioned problems, this paper proposes a novel end-to-end mobility management system for roaming across heterogeneous wireless networks. In this system, a connection manager (CM) is presented to intelligently detect the availability of different networks and the network conditions for vertical handoff decision. In CM, we propose a new QoS-based handoff metric including available bandwidth and access delay by using WLAN media access control (MAC) layer sensing, which can help to make accurate handoff decision and significantly reduce the wrong-decision probability during roaming from WWAN to WLAN. Moreover, in order to timely detect the unavailability of WLAN when roaming from WLAN to WWAN, a fast Fourier transform-based (FFT) signal decay detection scheme is introduced to reduce the ping-pong effect and the unnecessary handoff probability significantly.

To maintain connections' continuity by utilizing the network condition information provided by CM, a virtual connectivity (VC) is proposed. VC is based on the end-to-end principle and is a unified and complete end-to-end mobility scheme to maintain connections' continuity. Besides the basic end-to-end mobility management procedures, in VC, we introduce a local connection translation (LCT) which maintains mapping relationship between the original connection information and the current connection information for each active connection, thus making mobility transparent to applications. Moreover, an application layer subscription/notification (S/N) service is introduced to support mobility in case that network address translator (NAT) [6] and/or simultaneous movement are involved.

The rest of this paper is organized as follows. Section II reviews related works. Section III presents CM, including how to perform MAC-layer sensing and how to accurately detect the signal decay. In Section IV, VC is proposed as a complete and unified end-to-end mobility management scheme by introducing LCT and S/N. A prototype system is implemented and experiments are performed in Section V. Section VI concludes this paper.

## II. Related Work

In this section, we review related work on network condition detection and mobility management. Existing horizontal handoff, which deals with switching between base stations (BSs) [7], uses handoff metrics such as call blocking and call dropping rate to evaluate the performance. These metrics, however, are not applicable to vertical handoff. In the case of vertical handoff, a handoff decision algorithm faces the following challenges. First, signal strength comparison-based handoff decision algorithm used in today's horizontal handoff of cellular system cannot be applied to vertical handoff, since the roaming networks are asymmetric and there are no comparable signal strengths associated with them. Second, new metrics such as network available bandwidth, access delay, and packet error rate differ greatly in different wireless networks and are difficult to obtain compared with the physical-layer parameters such as received signal strength (RSS) and signal-to-interference ratio (SIR).

On existing vertical-handoff work, in [8], a neural network-based approach was proposed to detect the received signal decay and make handoff decision. Such an algorithm, however, is rather complex due to the complicated neural network topology and, hence, is difficult to implement in practice. In [9] a roaming scheme that considered the relative bandwidth of WLAN and GPRS was proposed, but no technical details on how to obtain the bandwidth were provided. In [4], a roaming scheme that considered only the signal strength was proposed. More detailed performance description of [4] can be found in [10]. In [11], a detailed vertical handoff signaling procedure was presented, but no details on the handoff decision algorithm were given. To the best of our knowledge, there is no reported work on how to perform MAC-layer QoS sensing in addition to physical-layer sensing to obtain the wireless network conditions for vertical handoff decision.

Once the network conditions are sensed and the vertical handoff is triggered, the next key issue is how to maintain the continuity of the on-going connections using a mobility management scheme. There are two types of mobility schemes: network infrastructure-based and end-to-end-based. Mobile IP (MIP) [12], [13] is the widely studied network-layer mobility management scheme, in which packets from and to the mobile host are tunneled through a home agent (HA) at its home network so that the corresponding node that communicates with the mobile host is unaware of the mobility of the mobile host. Based on MIP, in order to improve the routing performance and resolve the associated scalability problem, several extensions have been proposed [14]–[16]. All of these solutions significantly rely on the network infrastructure, in particular, HA. It has been noticed that the home and foreign agents could become bottlenecks since they need to handle the tunneled packets for a possibly large number of mobile hosts. As a network infrastructure-based approach, MIP may face deployment issue, since it needs HAs (and foreign agents for MIPv4) to be added into the network.

On the other hand, the end-to-end approaches can provide small handoff delay and can handle mobility without additional support from the network elements. Session initiation protocol (SIP) extensions are proposed to extend the protocol to support host mobility [17], [18] from end-to-end, alleviating some of the shortcomings associated with MIP and its route optimization variants. However, this approach can only be applied to applications that use SIP. TCP-R [19] and Migrate [20], which are end-to-end schemes dealing with mobility for TCP, introduce new states and options to handle the continuation of TCP connections. Though there are a few reported end-to-end mobility approaches [17]–[21], there still lacks a unified end-to-end mobility solution for various applications and works well under various mobility scenarios. Specifically, the following issues, namely, transparent to applications, work under NAT, and work under simultaneous movement, need to be further studied and addressed (see Section IV for detail discussion).

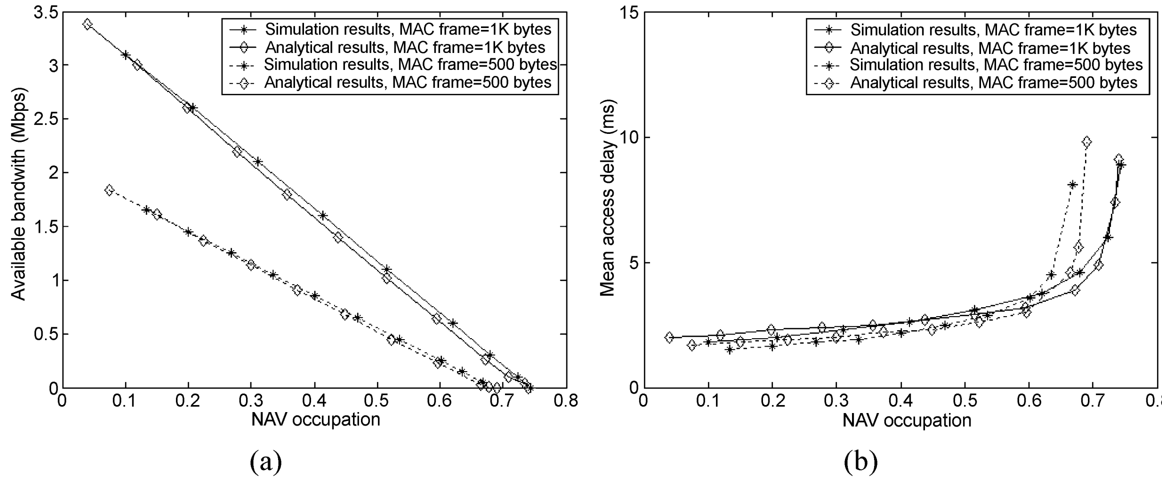



Fig. 1. Network performance under different NAV occupation. (a) Available bandwidth. (b) Mean access delay.

## III. Connection Manager for Roaming Detection

The primary functionality of CM is to intelligently sense and detect the network conditions, such as network type, signal strength, bandwidth, and delay. Motivated by the fact that the handoff in CM between WLAN and WWAN is vertical and asymmetrical, we introduce new handoff metrics for roaming between WLAN and WWAN. Specifically, from WWAN to WLAN roaming, we present MAC-layer sensing by taking into account the QoS of WLAN system for better connectivity; from WLAN to WWAN roaming, we propose an FFT decay detection algorithm for timely detecting unavailability of WLAN.

### A. Roaming From WWAN to WLAN

For roaming from WWAN to WLAN, existing approach is largely based on physical-layer sensing, e.g., in [10], where the objective is to guarantee the availability of the stable WLAN signal. To this end, the WLAN air interface will be periodically turned on to scan the beacons. Once the user moves into WLAN, a valid WLAN service set identifier (SSID) will be detected. Furthermore, in order to assure that the WLAN signal is stable enough, it is required that the received signal strength index (RSSI) be larger than a preset threshold. In addition to existing physical-layer sensing, we propose MAC-layer sensing technique considering not only the availability of WLAN but also the MAC-layer QoS conditions of WLAN, such as available bandwidth and access delay, to aid the handoff decision for better connectivity.

*1) MAC-Layer Sensing:* The goal of MAC-layer sensing is to detect the WLAN network condition, such as available bandwidth and mean MAC-layer access delay, which are used by CM as handoff metrics. If the WLAN network condition is better than that of the WWAN, the user will switch to WLAN by selecting the best available access point (AP). Otherwise, the user will stay with WWAN. As a result, the MAC-layer sensing can help users to choose the best AP and can reduce the wrong decision probability compared with the existing approach, where the roaming decision to WLAN is always performed as long as WLAN is detected to be available. To obtain the available bandwidth and access delay in MAC layer is not trivial since in MAC layer, we cannot obtain these parameters directly. Instead, we can only obtain some indirect information, e.g., by listening to and collecting the network allocation vector (NAV) in MAC layer. NAV is the main scheme used in IEEE 802.11 WLAN [22] to avoid collision by setting a busy duration on hearing frame transmissions from other mobile hosts. Thus, the NAV can well reflect the channel busy status.

To demonstrate the effectiveness of the bandwidth and delay estimation based on NAV, we perform both analysis and simulation. We consider the IEEE 802.11 WLAN system in which the MAC protocol is based on carrier sense multiple access with collission avoidance (CSMA/CA), in particular, under the distributed coordination function (DCF) mode [22]. In performance analysis, we derive how the NAV can be mapped to the available bandwidth and access delay (see the Appendix for details). In the simulation, we use ns-2 [23] with the random constant bit rate (CBR) traffic model and request-to-send/clear-to-send (RTS/CTS) enabled. We record the NAV values under various traffic conditions. We find that the simulation results are insensitive to the user number when the user number is within 6 to 30. In the following simulations, ten users are used. Fig. 1 illustrates the results obtained from both simulation and analysis on the estimated available bandwidth and access delay versus observed NAV. From Fig. 1(a), it can be observed that the system available bandwidth decreases almost linearly with the NAV occupation. From Fig. 1(b), we can see that the access delay explodes when the NAV reaches around 0.7 that indicates the system saturation point.

In CM, we observe the NAV for $T$ s and calculate the average NAV in this observation window. If the average NAV is smaller than a threshold $\delta_1$, roaming into WLAN is granted; otherwise, the user stays in WWAN. To measure the performance of the proposed MAC-layer sensing, we define the wrong-decision probability that is the sum of the following two components.

- *Unnecessary handoff probability*: Under a given input traffic $r$ and NAV observation window size $T$, the unnecessary handoff probability is defined as the probability of roaming into WLAN under the condition that the

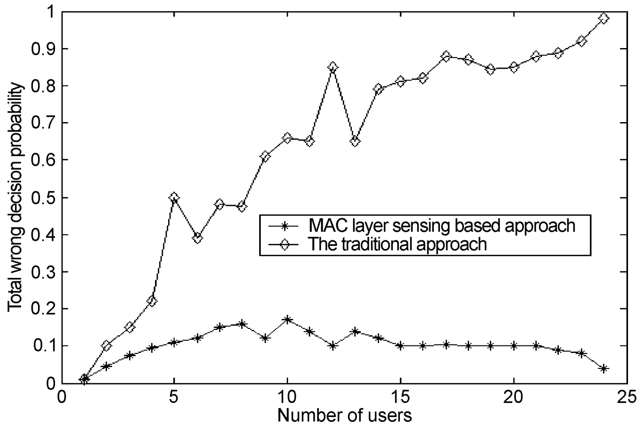



Fig. 2. Comparison of wrong-decision probabilities between our MAC-layer sensing approach and the traditional approach.

observed NAV values are smaller than $\delta_1$ but in the following $D$ s they are not all smaller than $\delta_2$.

- *Missing probability*: Under a given input traffic $r$ and NAV observation window size $T$, the missing probability is defined as the probability of not roaming into WLAN under the condition that the observed NAV values are larger than $\delta_1$ but in the following $D$ s they are all smaller than $\delta_2$.

It can be easily seen that the larger the NAV threshold $\delta_1$ is, the smaller the missing probability will be, but the larger the unnecessary handoff probability will be. To demonstrate the effectiveness of our MAC-layer sensing, we use the total wrong-decision probability to compare our MAC-layer sensing approach with the traditional approach where the decision is made once the WLAN is physically detected [10] using simulation. In the simulation, the NSWEB traffic model is used [24] with 1–24 users, $T$ is set to 1 s and $D$ is set to 3 s, and the NAV decision threshold $\delta_1 = 0.6$ and $\delta_2 = 0.64$. Fig. 2 shows the comparison results. The case with larger $T$ is also simulated and similar results are observed. From Fig. 2, it can be seen that the wrong decision probability with MAC sensing is much lower than the existing approach which only uses physical-layer sensing.

### *B. Roaming From WLAN to WWAN*

The goal of roaming from WLAN to WWAN is to switch to WWAN before the WLAN link breaks, meanwhile, to stay in WLAN as long as possible due to less cost and better QoS. However, because of fading effect, the WLAN RSSI may fluctuate severely (more than 10 dB usually). Thus, the key problem is how to timely detect the unavailability and the decay of the WLAN signal. To this end, we present an FFT-based approach for accurately detecting the unavailability of WLAN.

*1) FFT-Based Decay Detection:* To detect the signal decay, we propose an FFT-based decay detection approach based on the following proposition.

*Proposition:* The fundamental term of the FFT of a statistically decreasing sequence with length $N$ ($N$ is even) always has a negative imaginary part. That is

$$E\left[X(1) = \sum_{n=0}^{N-1} x(n)\sin\left(-\frac{2\pi n}{N}\right)\right] < 0. \tag{1}$$

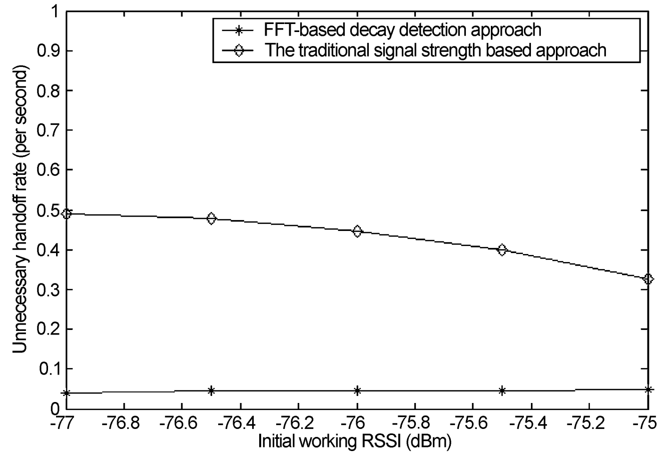



Fig. 3. Comparison of unnecessary handoff rate between our proposed FFT-based decay detection approach and the traditional signal strength-based approach.

*Proof:* Recall the imaginary part of the fundamental term of the FFT of $x(\mathrm{n})$ is given by

$$X(1) = \sum_{n=0}^{N-1} x(n)\sin\left(-\frac{2\pi n}{N}\right). \tag{2}$$

Note that $\sin(-(2\pi(N-n)/N)) = -\sin(-(2\pi n/N))$, then

$$\begin{aligned} E\left[X(1)\right] &= E\left[\sum_{n=0}^{N-1} x(n)\sin\left(-\frac{2\pi n}{N}\right)\right] \\ &= E\left[\sum_{n=0}^{\frac{N}{2}-1}\left(x(n)-x\left(\frac{N}{2}+n\right)\right)\sin\left(-\frac{2\pi n}{N}\right)\right]. \end{aligned} \tag{3}$$

Because $\sin(-(2\pi n/N))$ is always negative when $0 \leq n < N/2$, we obtain

$$E\left[X(1)\right] < 0.$$

■

If we regard $\sin(-(2\pi n/N))$ as a linear filter applied to the sequence $x(n)$, it is obvious that $X(1)$ is the most smooth metric because $\sin(-(2\pi n/N))$ is the filter with the least high-frequency component. This will reduce the variation of $X(1)$ even $x(n)$ may vary severely. Based on the above proposition, we can set a threshold for $X(1)/N$. If it is smaller than this threshold, the signal is considered to be decaying.

To demonstrate the effectiveness of the FFT-based decay detection on WLAN unavailability over the traditional signal strength-based detection [10], we perform the simulation and the results shown in Fig. 3. In this simulation, for the traditional signal strength-based detection approach, once the mean of the RSSI drops below a threshold, the signal is assumed to be decaying. However, for our approach, only when the mean RSSI is below the threshold and that the FFT threshold is also below a preset value, the signal is considered decaying. In the simulation, a Rician fading channel (Rician fading factor $= 2$ dB) with Doppler frequency shift being 5 Hz is assumed. The RSSI threshold is set to $-76$ dBm and the RSSI sampling interval is 100 ms, and the FFT threshold $X(1)/N$ is set to $-0.6$. When the user moves out of the WLAN area gradually (with speed of 0.7 m/s), the number of unnecessary handoff triggered is recorded for each approach. Since we find out in this case, the missing probability which is actually the handoff failure probability is negligible for both methods, we only compare the unnecessary handoff rate versus initial working RSSI, as shown in Fig. 3. From Fig. 3, it can be seen that the unnecessary handoff rate with our proposed FFT-based decay detection approach is much smaller than that with the traditional signal strength-based approach.

## IV. Virtual Connectivity for Mobility Management

This section addresses how to maintain continuity of the on-going connections. Due to the movement of mobile hosts, the IP addresses of the communication peers may change, resulting in connection's break if no mobility support is provided.

There are two important issues for mobility management. First, how to locate a mobile host. Second, how to maintain the continuity of the on-going connections. For the first issue, there are several possible ways to address it. For instance, by using dynamic updates of DNS [25], each mobile host updates its record when its IP address changes. We also note that peer-to-peer (P2P) technology may be very suitable for this purpose, since P2P systems provide that "given a key, it returns the IP address of the node which owns the key" [26] in a distributed, dynamic environment like the Internet. For the second issue, as mentioned in Section I, the end-to-end mobility management schemes have good properties such as independent of additional network infrastructure and being able to achieve small handoff delay. However, the following important issues have not been well addressed by the existing end-to-end schemes.

1) Make mobility transparent to application. By transparency to application, we mean that the upper layer applications should be unaware of the IP address and port number change caused by the mobility events.
2) Work under NAT. NAT is very popular due to lack of global IP addresses. Since the connection setup procedure is unidirectional under NAT, when a mobile host on the public side moves, it will not be able to notify its communication peer on the private side.
3) Work under simultaneous movement. The connection update information cannot be sent to the right receiver since both of the hosts have moved away in simultaneous movement case. As more and more wireless devices connect to the Internet, simultaneous movement will happen more frequently.

We propose VC to address the aforementioned issues as follows. First, VC introduces end-to-end operations for two communication peers to exchange mobility messages directly. Second, VC introduces a LCT, which provides transparency to applications by maintaining the original connection information unchanged no matter how the hosts move during the connection life span. Third, an application level subscription/notification (S/N) service is introduced to support mobility in the cases of simultaneous movement or when NAT is used. VC resides between IP layer and transport layer, thus, it works for both IPv4 and IPv6 networks in a natural way. VC is, therefore, a unified and complete end-system-based mobility management scheme.

### A. Basic End-to-End Operations

In VC, two types of operations, namely peer negotiation and connection maintenance, are performed per connection. Two peers first use peer negotiation to agree on items that will be needed for secure and accurate mobility management before mobility events happen, and then use connection maintenance procedure to maintain the connection continuity when mobility events occur.

*1) Peer Negotiation:* The two peers exchange the following information during peer negotiation.

- *Shared Secret and Connection Identification*. In order to protect the connection from being hijacked by malicious attackers, the connection maintenance should be performed securely. We use Diffie–Hellman key agreement protocol [27] to create a shared secret between the two peers. The shared secret is used to secure the forthcoming connection maintenance operations. Based on the shared secret and the connection information, the peers create a local $cid$ to represent the connection. The procedure on how to create $cid$ will be described later in Section IV-B.
- *Original IP and Port Number*. In peer negotiation, the host also notifies its original address and port number to its peer. The receiver then can infer whether its peer is behind NAT box by comparing the original address and port number with the ones in the received packet.
- *Capability and Preference*. When both end hosts have explicit information about their peers, they can make more efficient decisions. For instance, if a host is publicly addressed and does not move, the basic connection maintenance procedure will be enough and the S/N service will not be needed. The end hosts can also exchange their certificates, such that a simpler connection maintenance procedure can be selected if they trust each other. The peers can negotiate with each other on which security method to use, whether it has multihome capability, etc. In case that S/N service is needed, the peers need to exchange and negotiate the preferred S/N server, so that they can subscribe the address change events of each other via the S/N service.

*2) Connection Maintenance:* VC provides the following primitives to maintain the connection's continuity when the IP addresses of the communicating peers change. Note that the basic operations are performed per connection instead of per host, since both IP address and port number for a connection may change if NAT is involved.

- *Connection update (CU)*: When a host's address is changed, it sends CU to its peer directly to update its new connection information for each on-going connection. When a connection update request (CUR) is received for a connection, a CU should also be sent out for that specific connection.
- *Connection update ACK (CUA)*: It is an acknowledgment message for a received CU. CUA is the response to CU when the receiver trusts that the peer is really at its claimed current address carried in the previous CU.
- *Connection update challenge (CUC)*: In case that a peer needs to check whether the mobile node is reachable at its claimed current address, the peer sends out a challenge

message CUC to the mobile node's claimed current address. If the mobile node can receive the CUC and gives the right response, the peer is sure that the mobile node is really at its claimed current address.

- *Connection update challenge response (CCR)*: CCR is the response message for CUC. CU, CUC, and CCR compose a three-way handshake to ensure that the mobile node is reachable at its claimed current address (return routability test), so that a malicious mobile node cannot redirect traffic to innocent nodes.
- *Connection update request (CUR)*: A host uses CUR to requests its peer to send a CU either its current binding is close to timeout or being notified of its peer address change.

The connection maintenance procedure, thus, has two forms: CU/CUA or CU/CUC/CCR. If both nodes trust each other, CU/CUA can be used; otherwise, they should use CU/CUC/CCR. The procedure is negotiated in the peer negotiation period. The format of these messages is as follows:

$$opcode\ cid\ conn\_info \quad [challenge/response]\ seq\_no\ signature,$$

where $opcode$ is the type of the message. $cid$ is the connection identification used to identify the connection this message belongs to (how to compute $cid$ is given in Section IV-B). $conn\_info$ is used to attach the current IP address and port number of the host, so that the peer can judge whether the host is behind NAT. $challenge/response$ only appear in CUC and CCR. $seq\_no$ is the sequence number, which increment upon each update, the sequence numbers for CU/CUA, CU/CUC/CCR, and CUR/CU should be the same. $signature$ is used to sign the message.

Specifically, we place the peer negotiation and connection maintenance messages above the IP layer. For TCP connection, new TCP options are introduced; for UDP packet, these messages are put between UDP and the IP headers of the packet. This allows VC to be independent of the network layer and thus can be a unified approach applied for both IPv4 and IPv6 networks.

### B. Local Connection Translation

One key observation from this design is that connection's continuity can be maintained without a globally unique home address. VC achieves this by introducing a $cid$ to uniquely identify a connection between two end hosts. $cid$ is not globally unique, it is only unique between the two communicating peers. $cid$ is generated during peer negotiation as follows:

$$cid = HASH(orig\_src\_addr, orig\_src\_port, orig\_dst\_addr, orig\_dst\_port, protocol, seq, shared\_key).$$

If one mobile host detects that the newly created $cid$ is identical to an existing $cid$, this results a hash collision and it needs to notify its peer. Then, both hosts will increase the $seq$ by one and recompute the $cid$ again. In this paper, we choose the length of $cid$ to be 64 bits, and the hash function to be SHA1, the $cid$ collision probability is, thus, very small. Unlike HIP [28], which introduces a statistically $global$ unique host identity to solve mobility problem, in VC, by introducing a $local$ $cid$ between two end hosts, we can decouple logical connection and IP address without introducing yet another global ID space.

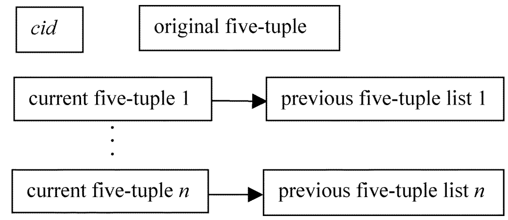



Fig. 4. LCT.

In order to address the application transparency, we propose to use an LCT table based on $cid$. The basic idea is to maintain a mapping relationship between the original connection information (i.e., $\langle \mathrm{src\_addr},\ \mathrm{src\_port},\ \mathrm{dst\_addr},\ \mathrm{dst\_port},\ \mathrm{protocol}\rangle$ five-tuple of a connection) and the current connection information for each active connection via $cid$, as illustrated in Fig. 4. The original connection information does not change during the connection life time, whereas the current connection information changes each time the node or the peer changes IP address. The upper layer applications only see the original connection information. Therefore, mobility is made transparent to the upper layer applications.

For packet transmissions, when application sends a packet, VC looks up LCT and updates the connection information, and then delivers the packet to the lower layer. For packet receptions, when VC receives a packet from outside network, it also looks up the LCT table and substitutes the current connection information of the packet to the original packet information, and then delivers the packet to the upper layer.

Since LCT needs to manipulate the connection information, it is placed in the TCP/IP kernel stack. The LCT item is created when $cid$ is negotiated, and is updated when the source and/or destination hosts move to new network access points or new network interface is added. After a connection migration (i.e., the five-tuple of the connection is updated), VC requires the migrated connection to carry $cid$ in each packet. In this way, the original connection information can be identified quickly from the LCT table by using $cid$ as index.

With the abstraction of $cid$ and LCT, VC can achieve multihome support naturally. As illustrated in Fig. 4, one $cid$ can map to multiple current five-tuples, where each current five-tuple represents a network interface. It thus can utilize LCT for a single logical connection to utilize several physical links at the same time. This feature is vital for VC to achieve seamless mobility management. By the proactive handoff signal from CM, after VC handoffs the on-going connections to the new network, it can still receive packets from the previous network. Hence, the connections will not lose those packets that are still in the previous pipe (see Section V-B1 for an experiment).

VC can also handle the address ambiguity problem caused by lacking global unique IP address using LCT. Suppose mobile host $A$ at location LA is communicating with a correspondent node $C$ using UDP, the five-tuple is $\langle \mathrm{IP}_A,\ \mathrm{port}_A,\ \mathrm{IP}_C,\ \mathrm{port}_C,\ \mathrm{udp}\rangle$. $A$ then moves to location LB and gets a new IP address $\mathrm{IP}_B$. It consequently updates its new information to $C$, and due to LCT, the connection is maintained transparently to the application running on $C$. Suppose a mobile host $B$ moves to location LA and gets the IP address $\mathrm{IP}_A$ (the previous address of $A$), and uses the same five-tuple to communicate with $C$. Under this circumstance, the application at $C$ will not be able to distinguish these two connections. There are two solutions to this problem using LCT: 1) if $B$ is VC enabled, $C$ can inform $B$ that a collision has been detected, then VC at $B$ can map $\langle \mathrm{IP}_A,\ \mathrm{port}_A,\ \mathrm{IP}_C,\ \mathrm{port}_C, \mathrm{udp}\rangle$ to $\langle \mathrm{IP}_A,\ \mathrm{port}'_A,\ \mathrm{IP}_C,\ \mathrm{port}_C,\ \mathrm{udp}\rangle$ locally so that the new port does not collide at $C$ and 2) if $B$ is not VC enabled, $C$ can translate the new five-tuple from $\langle \mathrm{IP}_A,\ \mathrm{port}_A,\ \mathrm{IP}_C,\ \mathrm{port}_C,\ \mathrm{udp}\rangle$ from $B$ to a new one, say $\langle \mathrm{IP}'_A,\ \mathrm{port}_A,\ \mathrm{IP}_C,\ \mathrm{port}_C,\ \mathrm{udp}\rangle$. Here, $\mathrm{IP}'_{\mathrm{A}}$ can be chosen from the private IP address space defined in RFC1598 [29]. By using this method, VC can avoid using different port number to do LCT translation, hence avoid polluting the port space.

In the above description of VC, we use "five-tuple" and "connection" interchangeably. However, if we treat every new five-tuple, such as UDP-based DNS queries or short-lived Web browsing, as connections that need mobility support, the computational overhead may be unbearable. In the following, we present a heuristic to deduce whether a connection needs VC support to alleviate the implementation cost. The heuristic works as follows.

1) Only if a five-tuple is active for a period of time, say, $>$ *tthresh* seconds, and the bytes transmitted $>$ *bthresh* bytes, can it be treated by VC as a connection that needs mobility support.
2) When a five-tuple is deduced to be a connection that needs mobility support, only the sender can begin the VC negotiation.

We note that this heuristic can filter the short-lived five-tuples efficiently, considering the fact that most of the five-tuples only live for a short period of time or transmit only small amount of traffic, i.e., the size of the connection is heavy-tailed [30]. As to the traces in [31], when setting *bthresh* to 10 KB and *tthresh* to 1 s, more than 95% of flows can be filtered by this heuristic. We note that this heuristic is very useful for Web browsing, which is dominated by short-lived TCP connections, since it is certainly not necessary to add mobility support for every short-lived TCP connections.

### C. *Subscription/Notification Service*

The S/N service is proposed to address the mobility problem (i.e., mobility under NAT or simultaneous movement) that cannot be resolved by the existing end-to-end approaches. The S/N service consists of two parts: an S/N server implementing the subscription/notification functionalities and an S/N client located at each end host exchanging information with the S/N server using an S/N protocol. The idea of using S/N service is as follows. A node can subscribe the address changes of mobile node(s) via the S/N service. Once a mobile node changes its IP address, it will inform its S/N server. Then, the S/N server will *directly* notify the address change information of the mobile node to the peers, which have subscribed this event. Once the peers know the new address of the mobile node, the otherwise broken connection can be resumed by using the basic end-to-end connection maintenance procedures. The advantage of using S/N service is that the address change information can be actively "pushed" to the peers timely. Compared with the methods that a node "polls" the current address of the peer when it finds out that its peer is not reachable (by timeout), S/N service can provide shorter handoff delay. We assume S/N server is publicly addressable and is not mobile.

We next discuss how S/N service solves the NAT and simultaneous issues. We assume hosts $A$ and $B$ are communicating with each other. As to the NAT issue described previously in Section II, both $A$ (the node with private address) and $B$ (the node with public address) are connected to an S/N server (they could connect to different S/N servers separately, we assume they are connected to a single server for simplicity), and $A$ subscribes to $B$'s IP address changes via the S/N server. If $B$ moves to a new network access point, it will inform the S/N server of its new address. Then, the S/N server notifies $A$ of the new address of $B$. After that, $A$ can resume the connection with $B$ by issuing the connection maintenance procedure.

As to the simultaneous movement issue, both $A$ and $B$ are connected to an S/N server, $A$ and $B$ subscribe to the IP address changes of each other. When $A$ and $B$ move simultaneously, they inform the S/N server their new IP addresses, and then the S/N notifies $A$ and $B$ of the new addresses of $B$ and $A$, respectively. Consequently, $A$ and $B$ can resume the connection by exchanging CU/CUA.

From the above description, the functionalities of S/N server can be summarized as follows: 1) detects IP address change of mobile clients; 2) accepts subscriptions; and 3) delivers notifications.

An S/N protocol is provided for the end host and the S/N server to communicate with each other. The details of the protocol are straightforward, and here we only provide a brief description. The end host uses the protocol to register/un-register itself, to subscribe/unsubscribe the hosts that it would like to know the address change information. The S/N server uses the protocol to notify the address change information of the registered hosts to the subscribers. The messages between the end host and the S/N server are secured by the shared key between them. This shared key may be assigned offline or created using Diffie–Hellman key agreement protocol online.

In the design of S/N service, in order to reduce the traffic between S/N server and mobile hosts, the following guidelines are proposed.

1) Only publicly addressed host can be subscribed. This is because private address is not globally unique, hence, even if a mobile host knows the private IP address of its peer, it still cannot contact the peer due to the separation of NAT.
2) S/N server may delay the notification message to the subscribers under the following case: suppose host $B$ which is publicly addressed, subscribes the address change of host $A$. When S/N server detects the address change of $A$, if it does not detect the address change of $B$ in the next $T$ time interval, S/N can safely cancel the notification to $B$ since $A$ should be able to update its connection information to $B$ directly under this case. The rationale behind the delayed notification is based on the fact that

simultaneous movement is a low probability event. However, $T$ should be carefully chosen so that the additional delay is bearable when simultaneous movement happens.

The combination of NAT and simultaneous movement can produce very complicated mobility cases which can be all handled by the proposed S/N service (except the case that both peers are behind NAT, where connection cannot be established from end-to-end). See Section V-B2 for an experiment on how the proposed S/N service handles NAT and simultaneous movement. Other cases can be handled similarly.

We note that S/N service is provided at the application layer. S/N differs from HA of MIP in that S/N moves part of the mobility functionality from within the network core to the end system, S/N therefore is not a network layer component and is deployed at the end systems, which makes it easier to deploy. Moreover, S/N only involves in signaling, not packet forwarding, and is invoked only under cases that cannot be handled by the basic end-to-end connection maintenance procedure (i.e., NAT or simultaneous movement). S/N is, thus, unlikely a congestion point.

### D. Handoff Delay Analysis of VC

Small handoff delay is important to achieve seamless roaming user experience. By using the end-to-end principle and the "push" method in the S/N service, VC achieves small handoff delay. We next analyze the handoff delay of VC and compare with that of MIPv6. For fair comparison, we choose CU/CUC/CCR as the connection maintenance procedure for VC, as both of them include return routability test. In what follows, we assume mobile node $A$ and $B$ are communicating with each other, and for single node movement case, we assume node $A$ changes its address, and the handoff delays can be derived, respectively

$$\begin{aligned} D_{n1}^{\mathrm{VC}} &= \mathrm{RTT}_{A-B} + T_{A->B} \\ D_{n1}^{\mathrm{MIPv6}} &= \mathrm{RTT}_{A-\mathrm{HA}_A} + \max \\ &\quad \times (\mathrm{RTT}_{A-B}, \mathrm{RTT}_{A-\mathrm{HA}_A} + \mathrm{RTT}_{\mathrm{HA}_A-B}) \\ &\quad + \mathrm{RTT}_{A-B} \end{aligned}$$

where $D_{n1}$ denotes the handoff delay for single node movement case, $\mathrm{RTT}_{x-y}$ is the round-trip time between nodes $x$ and $y$, and $T_{A->B}$ is the delay from $A$ to $B$, respectively. Since there is no HA in VC, the handoff delay of VC is at least less than one RTT smaller than that of MIPv6 in single node movement case.

For simultaneous movement case, the handoff delays are

$$\begin{aligned} D_{n2}^{\mathrm{VC}} &= \min\left(T_{A->B}^{\mathrm{S/N}}, T_{B->A}^{\mathrm{S/N}}\right) + D_{n1}^{\mathrm{VC}} \\ D_{n2}^{\mathrm{MIPv6}} &= \max_{\substack{X=A,Y=B \\ \text{or } X=B,Y=A}} (\mathrm{RTT}_{X-\mathrm{HA}_X} \\ &\quad + \max(\mathrm{RTT}_{X-\mathrm{HA}_X} + \mathrm{RTT}_{\mathrm{HA}_X-\mathrm{HA}_Y} \\ &\quad + \mathrm{RTT}_{\mathrm{HA}_Y-Y}, \mathrm{RTT}_{X-\mathrm{HA}_Y} + \mathrm{RTT}_{\mathrm{HA}_Y-Y}) \\ &\quad + \mathrm{RTT}_{X-\mathrm{HA}_Y} + \mathrm{RTT}_{\mathrm{HA}_Y-Y}) \end{aligned}$$

where $D_{n2}$ denotes the handoff delay for simultaneous movement case, and $T_{A->B}^{\mathrm{S/N}}$ and $T_{B->A}^{\mathrm{S/N}}$ are delays from $A$ to $B$ and $B$ to $A$ via S/N, respectively. $D_{n2}^{\mathrm{VC}}$ is calculated as follows. Since S/N pushes the address change information to the two peers simultaneously, the delay caused by S/N is, thus, the minimum value of $T_{A->B}^{\mathrm{S/N}}$ and $T_{B->A}^{\mathrm{S/N}}$ plus the delay caused by the basic end-to-end operations. However, for MIPv6, under simultaneous movement, a mobile node needs to perform the mobility management procedure with both HAs. Furthermore, handoff procedure is completed only after the two separate binding cache maintenance procedures are done. $D_{n2}^{\mathrm{MIPv6}}$ is, thus, the maximum value of the two procedures.

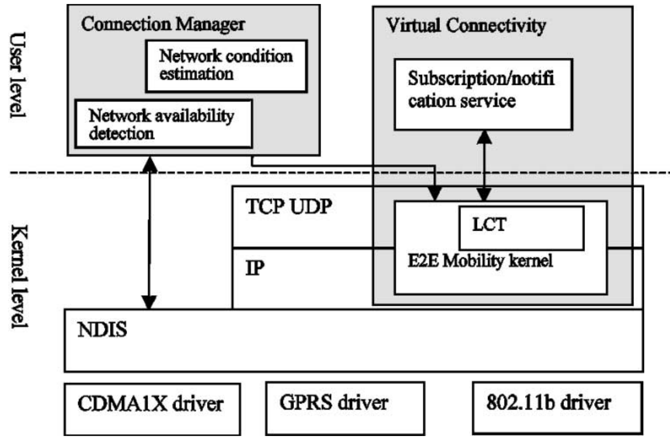



Fig. 5. Implementation architecture of the proposed mobility management system.

From the above analysis, we can conclude that in both cases, the handoff delay of VC is much smaller than that of MIPv6. This property is an important feature to enable seamless roaming with smaller handoff delay.

## V. System Implementation and Experiments

### A. System Implementation

To demonstrate the effectiveness of the proposed mobility system, we have implemented a prototype system on the Windows 2000 operating system based on the architecture illustrated in Fig. 5.

In this architecture, CM runs as a background service at user level. CM senses all the wireless network interfaces via the network driver interface specification (NDIS) device interface, which in turn manipulates all the network devices. CM performs the network availability detection (physical layer sensing) and network condition estimation (MAC-layer sensing) based on the methods introduced in Section III. After that, it can get information such as WLAN_WEAK and WLAN_AVAILABLE, available bandwidth, mean access delay. CM then decides whether it needs to switch the physical access network based on the information gathered. After vertical handoff decision is made, CM then triggers VC to perform connection maintenance functionalities.

The implementation of VC in this prototype system is composed of the following two parts: end-to-end mobility kernel part and S/N service part. The mobility kernel part is located at kernel level. It implements the peer negotiation, the connection maintenance, and the LCT table maintenance. This part is located within the TCP/IP protocol stack and is tightly integrated with the TCP/IP code to achieve high efficiency. For instance, the LCT data structure is added into the TCP control block (*TCB*) structure of TCP and the *SrcObj* structure of UDP [32]. Considerable efforts have been taken to minimize the

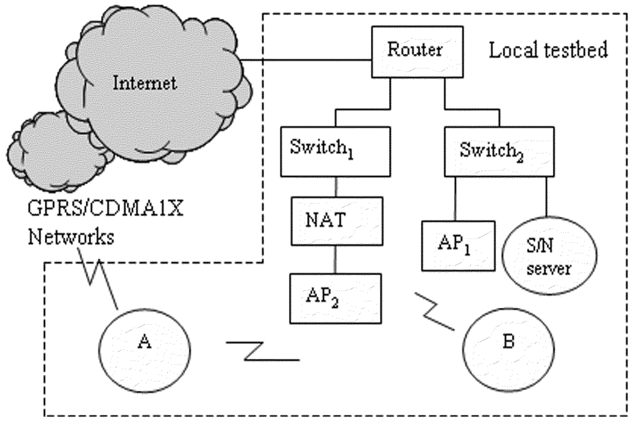



Fig. 6. Experiment testbed.

overhead introduced by VC. The S/N service part is located at user level. As we have described in Section IV-C, S/N service is composed of two parts. A centralized S/N server is implemented to realize the subscription/notification functionalities, and a S/N client is located at each end host. The S/N client and S/N server use the S/N protocol to exchange mobility related information. An interface is designed for the mobility kernel to exchange information with CM and the S/N service. CM can trigger the mobility kernel to perform mobility management procedures and the S/N service can exchange subscription/notification messages with the mobility kernel via this interface. The protocol stack we use in this implementation is the IPv6 stack from Microsoft Research [32].

### B. Experiments

The experimental testbed is setup, as depicted in Fig. 6. Mobile hosts $A$ and $B$ have multiple methods to access the network. They can connect to the Internet via the China Mobile GPRS network (with a maximum 40 kb/s downlink rate) using a GPRS Aircard 750 card or via the China Unicom CDMA1X network (with a maximum 153.6 kb/s) using a ZTE MC310 PC Card. $A$ and $B$ can also access the local WLAN network via $\mathrm{AP}_1$ or $\mathrm{AP}_2$ using a Lucent WaveLAN 802.11b card ($\mathrm{AP}_1$ and $\mathrm{AP}_2$ work under two noninterfering channels and belong to the same distribution system), and they can connect to the local network via the conventional Ethernet. Both $A$ and $B$ install the prototype system (i.e., CM and VC enabled). In all the experiments that follow, the last hop wireless link is the only bottleneck.

In the following experiments, GPRS/CDMA WWAN is assumed always available and 802.11b WLAN is assumed to be available only at certain hot spots.

Based on the implementation of the prototype system, we perform the following experiments to show that the proposed system can indeed achieve end-to-end mobility proactively and seamlessly. Only basic CU/CUA connection maintenance procedure is used for simplicity in the experiments. CU/CUC/CCR procedure works approximately the same, except that its handoff time is one RTT longer.

*1) Seamless Roaming and Proactive Network Selection:* We first show how seamless roaming is achieved. In this experiment, host $B$ is connected to the local network via Ethernet. At first, mobile host $A$ is communicating with $B$ via GPRS, it then moves from GPRS to WLAN; after some time, it moves back to GPRS network. There is a TCP connection between $A$ and $B$, transferring data packet from $B$ to $A$.

The sequence numbers of the data packets received by $A$ during the two handoff periods are illustrated in Fig. 7(a) and (b). It can be seen from Fig. 7 that the connection is maintained successfully in both directional movements. More specifically, in the case of roaming from GRPS to WLAN, at time 39.22 s CM triggers VC to migrate to the WLAN network, and new packets via WLAN are received at $A$ at time 39.46 s. However, there are still nine packets with 5796 bytes in the previous GPRS pipe. Due to the design of VC and since GPRS is always on, these data in the previous pipe will not be lost and will still trigger back acknowledgments (ACKs), which is important to avoid a time out. The data in the GPRS pipe will be drained at time 40.55 s. Therefore, under this circumstance, VC can achieve zero disruption time[1] and very small handoff delay (the time interval between the time CU is sent out and the time CUA is received, i.e., $39.40 - 39.22 = 0.18$ s). Note that seamless mobility is achieved since the disruption time is 0.

In the case of roaming from WLAN to GPRS, we can see that at time 463.5 s CM triggers $A$ to send CU to $B$ via GPRS, and $A$ receives CUA at time 464.3 s. In this experiment, some data packets from $B$ to $A$ are lost after $A$ detaches from WLAN, hence, TCP will time out and slow start; the first data packet transmitted via GPRS is at time 465.6 s. In this case, the handoff time is $464.3 - 463.5 = 0.8$ s, and the disruption time is approximately 2.1 s, which is much smaller than that in the system presented in [10] ($>10$ s for TCP). Note that in this case when bandwidth changes from high to low, the disruption time is related to the upper layer transport protocol.

We then use the following example to show how proactive network selection is achieved. The setup of this experiment is similar. Host $A$ moves from GPRS to WLAN hot spots. However, there are more than one access points (namely, $\mathrm{AP}_1$ and $\mathrm{AP}_2$ in this experiment) that $A$ can associate with. With the help of CM, $A$ can choose the best network to enter. Here, by best, we mean highest available bandwidth.

In the traditional scheme without MAC-layer sensing, host $A$ will pick up an AP randomly. However, in our proposed system, $A$ can proactively measure the available bandwidths of $\mathrm{AP}_1$ and $\mathrm{AP}_2$, and choose the AP with higher available bandwidth.

We then measure the throughput in each approach with and without available bandwidth estimation. Fig. 8 shows the sequence numbers of the data packets captured at node $A$, the available bandwidths of $\mathrm{AP}_1$ and $\mathrm{AP}_2$ are about 2.02 and 3.93 Mb/s, respectively. Hence, the proposed system can choose the AP with better network condition (in this case, $\mathrm{AP}_2$) to associate with, whereas the traditional approach may choose the network with the lower available bandwidth. We can see that the connections in both cases are maintained seamlessly due to the functionality of VC.

*2) Connection Maintenance Under NAT and Simultaneous Movement:* In this experiment, as depicted in Fig. 9, there is a

[1]In this paper, we define the disruption time as $\min(0, T_c - T_p)$, where $T_c$ is the time when the first packet received from the new network interface, and $T_p$ is the time when the last packet received from the previous network interface.

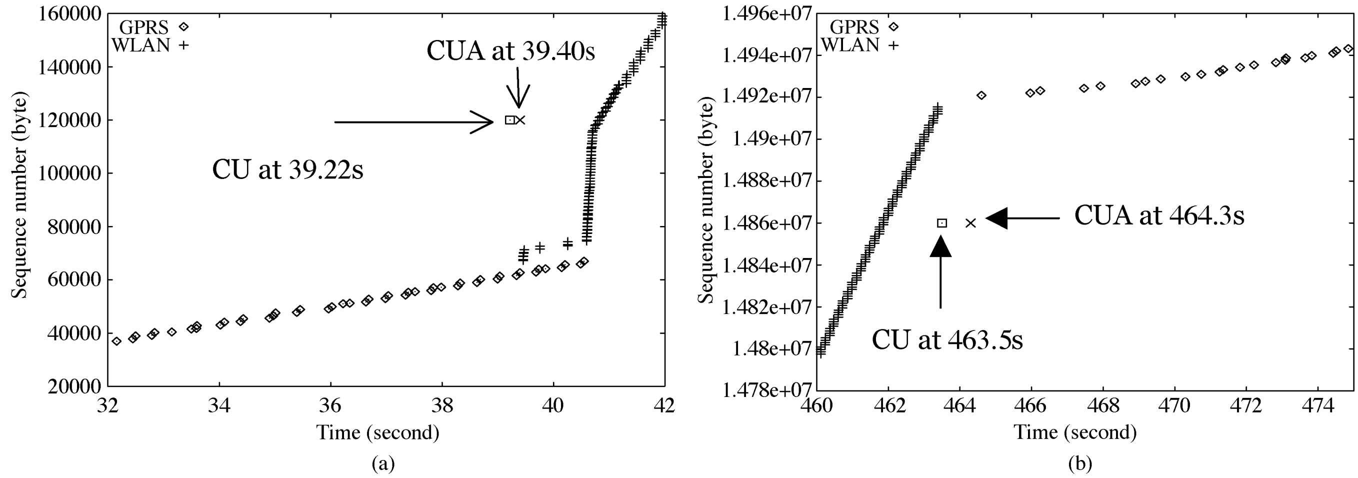



Fig. 7. TCP sequence number measured at node $A$. (a) Handoff from GPRS to WLAN. (b) Handoff from WLAN to GPRS.

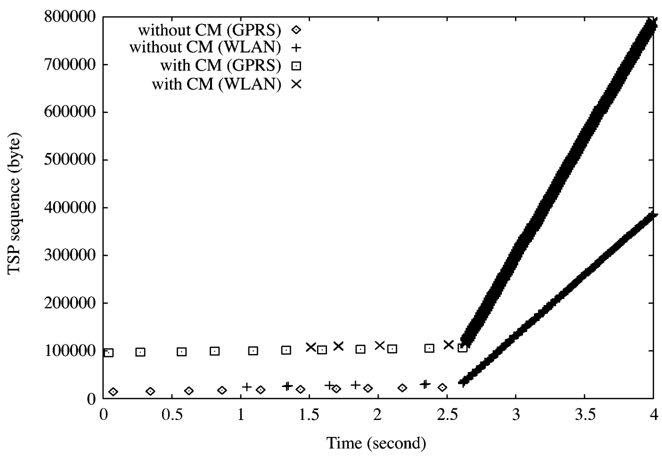



Fig. 8. TCP sequence number measured at host $A$ with and without CM.

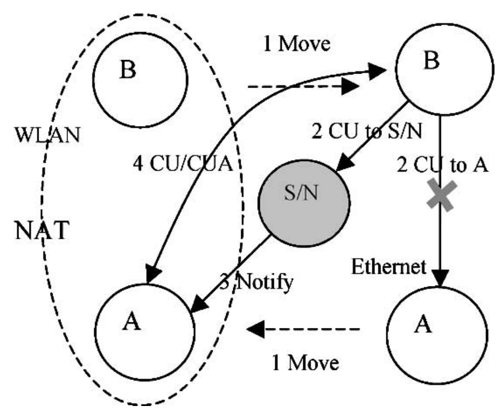



Fig. 9. Mobility management under simultaneous movement and NAT.

duplex UDP connection between $A$ and $B$, representing a bidirectional voice communication. The rate of the UDP connection is 32 Kb/s and the packets are equally 250 bytes long. At the beginning, mobile host $A$ is behind NAT and accesses the network via $\mathrm{AP}_2$, mobile host $B$ is connected to the network through Ethernet. After some time, $B$ changes from WLAN to Ethernet by plugging the Ethernet cable in and $A$ moves from Ethernet to WLAN by unplugging the Ethernet cable simultaneously. We use WiNE [33], a network emulator for windows platform, to add 100–ms delay for the Ethernet interfaces of $A$ and $B$.

The mobility management procedure works as follows.

1) After simultaneous movement, $A$ and $B$ send CU messages to the S/N server to update the connections with S/N, and $B$ also sends CU to $A$'s previous location to update the connection with $A$, however, this message will be lost since $A$ has moved away ($A$ does not send CU to $B$ since $B$'s previous address is private).
2) S/N detects that both $A$ and $B$ have moved.
3) It notifies $A$ of $B$'s new IP address via a notification message (but it does not notify $B$ of $A$'s new IP address since $A$'s new address is a private address.
4) $A$ issues CU message to $B$'s new IP address and $B$ replies a CUA message to $A$'s new IP address, after that, the connection between $A$ and $B$ is kept.

Fig. 10 illustrates the traffic trace received at host $A$ and $B$. Since in this experiment, $A$ and $B$ moves simultaneously, hosts $A$ and $B$ cannot update the connection information directly ($B$ moves at time 27.84 s and $A$ moves at time 27.92 s. The CU from $B$ to $A$ will be lost since 27.92 s $-$ 27.84 s $<$ 100 ms). From the traces, we can see that this bidirectional connection is maintained successfully. More specifically, for host $A$, after it moves at time 27.92 s, it cannot receive packets from $B$ anymore. Host $A$ receives a notification message from the S/N server at time 27.97 s and it sends out a CU to $B$ to update its new connection information using the new IP address of $B$ immediately. After $A$ receives the CUA from $B$'s new location at time 28.17 s, it begins to receive packets from $B$ again. The situation is a little different for host $B$. Since, in this case, $A$ changes to private address, the S/N server will not notify $A$'s new address to $B$. $B$ moves at time 27.84 s and waits for the CU message from $A$, and receives the CU message from $A$ at time 28.07 s, after that, it begins to receive packets from $A$. In this case, as depicted in Fig. 10, the handoff delays for host $A$ and $B$ are about 250 and 230 ms, respectively. This experiment, therefore, demonstrates that VC can support simultaneous movement efficiently.

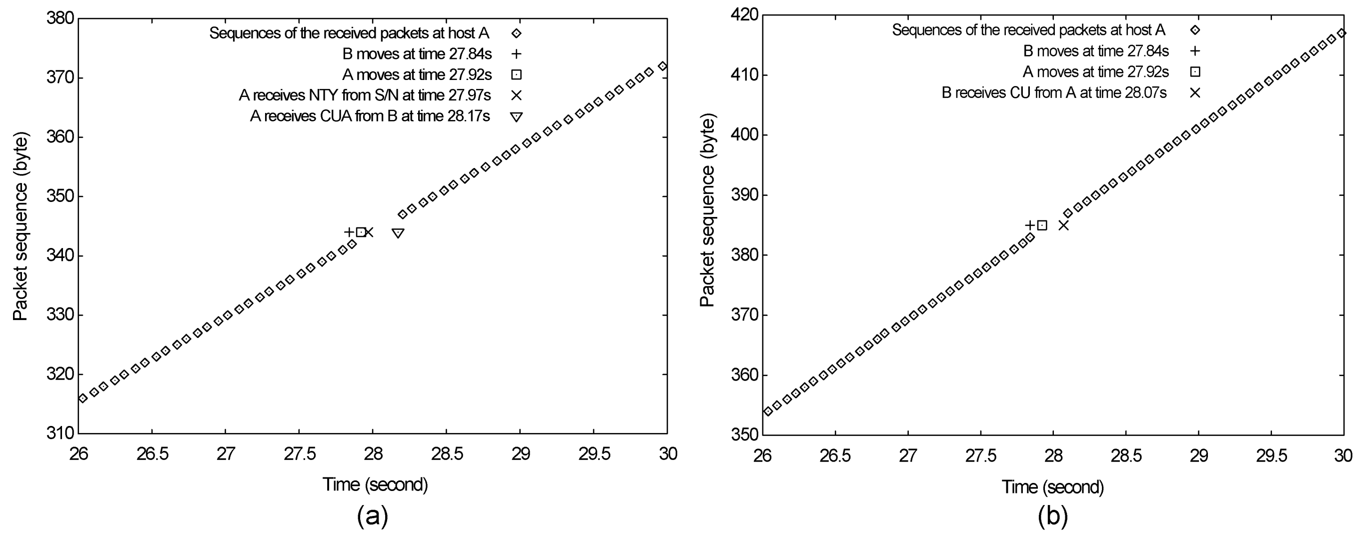



Fig. 10. (a) Sequences of the received packets at host $A$. (b) Sequences of the received packets at host $B$.

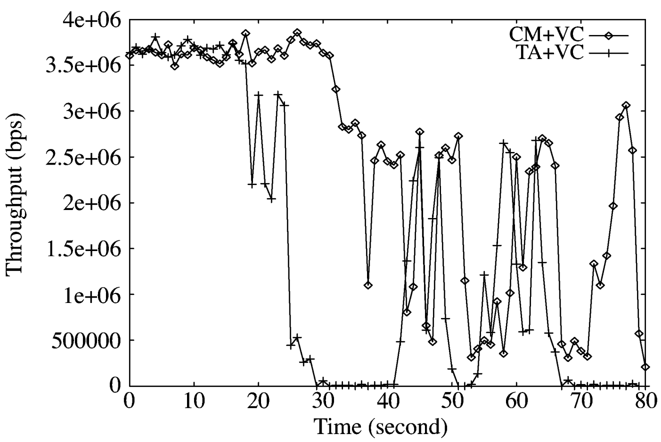



Fig. 11. TCP throughputs under our system and the traditional signal strength-based handoff plus VC.

*3) Fewer Ping-Pong Switches and Higher Throughput:* The setup of this experiment is the same as that of experiment 1. In this experiment, mobile host $A$ roams around the edge of the WLAN, and then moves away from WLAN to CDMA1X. Due to the signal fading of WLAN, $A$ may switch between CDMA1X and WLAN and between $\mathrm{AP}_1$ and $\mathrm{AP}_2$ within the same WLAN (link layer handoff, which is transparent to the network layer) frequently if traditional handoff scheme is used. However, with help from CM, the ping-pong effect can be minimized; hence much higher TCP throughput is achieved.

The TCP throughputs (averaged in 1 s) under the traditional signal strength-based handoff plus VC approach ($\mathrm{TA}+\mathrm{VC}$ for abbreviation) and our mobility system are depicted in Fig. 11. During the testing period, TCP connection is maintained successfully under both cases owing to VC. From Fig. 11, we observe that by combining CM and VC, much higher throughput can be obtained since CM can notify the handoff decision more accurately. In this experiment, TA triggers five vertical handoffs (three for WLAN to CDMA1X and two for CDMA1X to WLAN), while CM triggers no vertical handoff expect the one when $A$ finally steps out of WLAN. The average TCP throughputs are 2.5 and 1.46 Mb/s under $\mathrm{CM}+\mathrm{VC}$ and $\mathrm{TA}+\mathrm{VC}$, respectively. Note that sometimes the transient throughput may also be low even in WLAN due to the low RSSI and unknown interference, but still much larger than the one in CDMA1X, which achieves only tens of kilobits per second.

The above three experiments conclusively demonstrate that, by integrating CM and VC, the proposed mobility system can successfully provide mobility support for heterogeneous wireless networks. It reacts to the roaming events not only proactively but also more accurately as compared with traditional approaches. The connections are maintained seamlessly from end-to-end under different mobility cases and the overall performance is significantly improved.

## VI. Conclusion and Discussions

In this paper, a novel end-to-end mobility management system is proposed for seamless and proactive roaming across heterogeneous wireless networks. The proposed system integrates a *connection manager* that intelligently detects the condition of the wireless networks and a *virtual connectivity*-based mobility management scheme that maintains connection's continuity using the end-to-end principle. The proposed system reacts to roaming events proactively and accurately, maintains the connections' continuity seamlessly, and is a unified end-to-end mobility management system for both IPv4 and IPv6.

More specifically, in CM, MAC-layer sensing is proposed in addition to the traditional physical-layer sensing to obtain network parameters such as available bandwidth and access delay when roaming from WWAN to WLAN, and an FFT decay detection algorithm is proposed to significantly reduce the unnecessary handoff rate and ping-pong effect when roaming from WLAN to WWAN. In VC, end-to-end connection maintenance mechanism is introduced to make it independent of additional network layer infrastructure. VC consists of an LCT to make mobility transparent to upper layer applications and an S/N service to successfully handle mobility under NAT and simultaneous movement.

We have built a prototype system and have performed a series of experiments, using roaming among GPRS/CDMA1X, 802.11b, and Ethernet as examples, to verify and demonstrate the properties of the proposed system. The experimental results demonstrate that the proposed system achieves seamless and proactive end-to-end mobility when mobile host roams across heterogeneous wireless networks.

We are currently working to extend CM to include wireless personal area networks (WPAN), WCDMA, IEEE 802.11a/g into this prototype system. We are also investigating how to use P2P networks to decentralize the S/N service, since the "lookup" service provided by the P2P networks [26] can be extended to the "subscription/notification" semantic. In this way, a distributed, self-organized S/N can be achieved without a central point.

## APPENDIX

In this Appendix, we derive the available bandwidth and access delay from the NAV for the IEEE 802.11 WLAN with DCF mode [22].

### A. Available Bandwidth From NAV

According to the equivalent *p-persistent* model [34], the probability with $i$ time slots between two successive transmissions can be given by the geometric distribution with parameter $\alpha$ as follows:

$$(1-\alpha)\alpha^{i-1}, \quad i = 1, 2, \ldots, \infty. \tag{A-1}$$

This can also be derived by assuming that the user number is large enough and the traffic collectively follows Poisson process [35], [36].

Thus, the mean time interval between two successive transmissions $\tau$ is

$$\tau = \sum_{i=1}^{\infty}(1-\alpha)\alpha^{i-1}(i\delta) \tag{A-2}$$

where $\delta$ is the duration of a time slot. From (A-2), $\alpha$ can then be derived as

$$\alpha = 1 - \frac{\delta}{\tau}. \tag{A-3}$$

With this model, the probability of simultaneous transmissions from $k$ users at a time slot will decrease exponentially as $k$ increases. Since the collision probability caused by simultaneous transmission of $k$ users $(k > 2)$ is usually quite small, we approximate that the collisions are caused by two simultaneous transmissions. Since the channel busy occupation is caused by either a successful transmission or a collision, we can derived the mean frame rate $\lambda$ (frames/second) as

$$\lambda = \frac{b}{T_s + \frac{(N-1)T_{\text{col}}}{2}} \tag{A-4}$$

where $b$ is the channel busy time per second, $N$ is the average number of trials of a transmission, $T_s$ is the time length of a successful transmission which is the sum of RTS, CTS, DATA, and ACK, and $T_{\text{col}}$ is the length of a collision. Then, the average interval between two successive transmissions $\tau$ can be obtained by dividing the idle time with the number of transmission trials. That is

$$\tau = \frac{1-b}{\lambda + \frac{(N-1)\lambda}{2}} = \frac{(1-b)\left[T_s + \frac{(N-1)T_{\text{col}}}{2}\right]}{b + \frac{(N-1)b}{2}}. \tag{A-5}$$

From the desired user point of view, at the boundary of each time slot, a transmission from other users is inserted with probability $p_1 = 1-\alpha$. Accordingly, the probability with no transmission inserted is $p_0 = \alpha$. As a result, the probability of the successful transmission of a frame for the $j$th trial is $p_0 p_1^{j-1}$. Thus, $N$ is obtained as follows:

$$\begin{aligned} N &= p_0 + 2p_0p_1 + 3p_0p_1^2 + \ldots + Sp_0p_1^{S-1} \\ &= \sum_{j=1}^{S} jp_0p_1^{j-1} \\ &= \frac{1-(1-\alpha)^S}{\alpha} - S(1-\alpha)^S \end{aligned} \tag{A-6}$$

where $S$ is the maximum number of trials before a frame is discarded.

It can be derived that the NAV has the following relationship with $N$ and $\lambda$:

$$\text{NAV} = \lambda T_n + \frac{(N-1)\lambda T_{n,c}}{2} \tag{A-7}$$

where $T_n$ is the NAV duration for a successful frame transmission and $T_{n,c}$ is the NAV duration for a collision. By combining (A-3)–(A-7), we can solve the five unknowns, $\tau$, $\alpha$, $N$, $\lambda$, and $b$ for a given NAV.

Then, the available bandwidth can be obtained from NAV as follows:

$$\text{BW} = B_0 - \lambda L = B_0 - L\frac{\text{NAV}}{T_n + \frac{T_{n,c}(N-1)}{2}} \tag{A-8}$$

where $L$ is the mean frame size and $B_0$ the total system bandwidth which can be obtained in the following section.

### B. Delay From NAV

In this section, we derive the relationship between NAV and access delay. The access delay is defined as the time spent from the time when a MAC frame becomes the head of a MAC layer queue to the time when this frame is successful transmitted.

Assume that in the current system, the desired user generates a new frame and wants to send it through the WLAN. According to IEEE 802.11 WLAN MAC, a contention window $w$ will be uniformly generated within the range of $[0, \text{CW}_{\min}]$ after sending a MAC frame, where $\text{CW}_{\min}$ is the initial contention window range defined in DCF mode. Thus, the access delay $t$ is given by

$$t = t_0 + t_f + t_{\text{send}} \tag{A-9}$$

where $t_0$ denotes the time interval when this new generated frame should wait until the channel is idle, $t_f$ denotes the time needed to reduce the contention window counter to zero, and $t_{\text{send}}$ denotes the time needed to send the frame after the counter

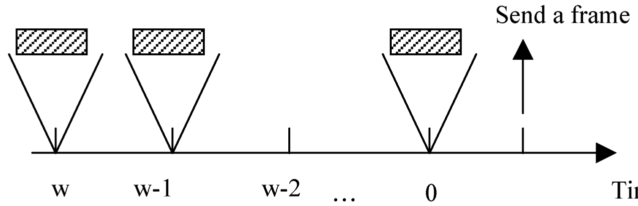



Fig. 12. Reduction of contention window.

reduces to zero. It can be seen that the mean of $t_0$ can be expressed as

$$E[t_0] = \frac{b\left(p_c T_{\mathrm{col}} + (1-p_c)T_s\right)}{2} \tag{A-10}$$

where the collision probability $p_c$ denoting the number of the collisions among all the trials to send a frame, can be given by

$$p_c = \frac{N-1}{N}. \tag{A-11}$$

Next, we calculate the mean values of $t_f$ and $t_{\mathrm{send}}$ in the following sections, respectively.

*1) Computation of $t_f$:* Note that the contention window $w$ is reduced by one after each idle time slot. However, the reduction will be suspended due to the transmission from other users. In each time slot, a transmission from other users is inserted with probability $p_1 = 1 - \alpha$. Accordingly, the probability with no transmission inserted is $p_0 = \alpha$. In Fig. 12, we plot the procedure of reducing the contention window $w$.

From Fig. 12, we can see that the time needed to reduce the contention window $w$ to zero is given by

$$\begin{aligned} u(w) &= \binom{w}{0} p_0^w [w\delta] + \binom{w}{1} p_0^{w-1} p_1 \\ &\quad \times [w\delta + p_c T_{\mathrm{col}} + (1-p_c)T_s] + \ldots \\ &\quad + \binom{w}{w} p_1^w \left[w\delta + w\left(p_c T_{\mathrm{col}} + (1-p_c)T_s\right)\right] \\ &= w\delta + \sum_{i=0}^{w} \binom{w}{i} p_0^{w-i} p_1^i \times \left(i\left(p_c T_{\mathrm{col}} + (1-p_c)T_s\right)\right). \end{aligned} \tag{A-12}$$

Define $v(D)$ as the mean of $u(w)$, where $w$ is uniformly distributed in $[0, D]$, then

$$v(D) = E\left[u(w)\right] = \frac{1}{D+1} \sum_{w=0}^{D} u(w). \tag{A-13}$$

In DCF mode, a contention window $w$, which is uniformly distributed in $[0, CW_{\min}]$, is generated after previous MAC frame transmission. When a new frame arrives, it will be transmitted immediately when the channel is sensed idle [22]; otherwise, the average remaining time can be approximated to be $v(\mathrm{CW}_{\min})/2$. Thus, the mean of $t_f$ can be obtained as

$$E[t_f] = b * \frac{v(\mathrm{CW}_{\min})}{2}. \tag{A-14}$$

*2) Computation of $t_{\mathrm{send}}$:* When the backoff counter is reduced to zero, the frame is sent. Obviously, the probability that this frame collides with other frames is exactly the same as the probability that a transmission will be inserted during the counter reduction, which is $1-\alpha$. However, in the case of collision, $T_{\mathrm{col}}$ will be wasted and the user will double the contention window and regenerate a backoff counter $w \in [0, 2\mathrm{CW}_{\min}]$. Note that in IEEE 802.11 DCF, the new CW range is $2\mathrm{CW}_{\min}+1$ instead of $2\mathrm{CW}_{\min}$. Here, for simplicity, we ignore the subtle difference. Then, another round of counter reduction plus the trial of sending frame starts. Thus, the mean of $t_{\mathrm{send}}$ is given by

$$\begin{aligned} E(t_{\mathrm{send}}) &= E\left(t_{\mathrm{send}}^{(1)}\right) \\ &= p_0 T_s + p_1 \left[T_{\mathrm{col}} + v(2\mathrm{CW}_{\min}) + E\left(t_{\mathrm{send}}^{(2)}\right)\right] \end{aligned} \tag{A-15}$$

where

$$E\left(t_{\mathrm{send}}^{(2)}\right) = p_0 T_s + p_1 \left[T_{\mathrm{col}} + v(4\mathrm{CW}_{\min}) + E\left(t_{\mathrm{send}}^{(3)}\right)\right]. \tag{A-16}$$

It can be seen that this is a recursive structure and $E[t_{\mathrm{send}}]$ can be written as

$$\begin{aligned} E(t_{\mathrm{send}}) &= \sum_{j=1}^{S-1} p_1^j v(2^j \mathrm{CW}_{\min}) + p_0 T_s \sum_{j=1}^{S-1} p_1^{j-1} \\ &\quad + T_{\mathrm{col}} \sum_{j=1}^{S-1} p_1^j + p_1^{S-1} E\left(t_{\mathrm{send}}^{(S)}\right). \end{aligned} \tag{A-17}$$

In the above equations, in order to guarantee that the transmission time does not explode and the system does not overflow, we need to assure that the following series converges:

$$p_1^j v(2^j CW_{\min}), \quad j = 0, 1, 2 \ldots. \tag{A-18}$$

Here, the convergence of above series means that the probability of successfully sending the frame after one or two collisions should be large enough though the first trial to send the frame may fail. Based on the convergence requirement of above series, we can obtain the maximum frame arrival rate $\lambda_0$ allowed by the system. Then, the total system bandwidth $B_0$ can be calculated as $B_0 = \lambda_0 L$.

Finally, by combining (A-10), (A-14), and (A-17), the mean of the access delay $t$ is given by

$$\begin{aligned} E[t] &= E[t_0] + E[t_f] + E[t_{\mathrm{send}}] \\ &= \frac{b\left(p_c T_{\mathrm{col}} + (1-p_c)T_s\right)}{2} + b * \frac{v(\mathrm{CW}_{\min})}{2} \\ &\quad + p_0 T_s \sum_{j=1}^{S-1} p_1^{j-1} + T_{\mathrm{col}} \sum_{j=1}^{S-1} p_1^j + \sum_{j=1}^{S-1} p_1^j v(2^j \mathrm{CW}_{\min}). \end{aligned} \tag{A-19}$$

Note that $P_1^{S-1} E(t_{\mathrm{send}}^{(S)})$ is omitted from (A-19) since the series in (A-18) converges and $p_1^{S-1} \to 0$, when $S$ is sufficiently large. Since all the required parameters in the above equation, $p_0, p_1$, and $p_c$, can be obtained from NAV with the derivation in the previous section, the mean access delay $E[t]$ can be derived from NAV accordingly.


## Acknowledgment

The authors thank Dr. Y. Yao, Dr. K. Tan, and Mr. H. Liao for their valuable discussions. We also would like to thank our visiting students J. Yuan and J. Li for performing some related experiments. Lastly we would like to thank Prof. B. Li from Hong Kong University of Science and Technology for reviewing this manuscript.

## REFERENCES

[1] E. Gustafsson and A. Jonsson, "Always best connected," *IEEE Wireless Commun.*, vol. 10, pp. 49–55, Feb. 2003.

[2] A. Hills and D. Johnson, "A wireless data network infrastructure at Carnegie Mellon University," *IEEE Pers. Commun.*, vol. 3, pp. 56–63, Feb. 1996.

[3] K. Lai, M. Roussopoulos, D. Tang, X. Zhao, and M. Baker, "Experiences with a mobile testbed," presented at the *WWCA*, Tsukuba, Japan, Mar. 1998.

[4] A. Sanmateu, F. Paint, L. Morand, S. Tessier, P. Fouquart, A. Sollund, and E. Bustos, "Seamless mobility across IP networks using Mobile IP," *Comput. Networks*, vol. 40, pp. 181–190, 2002.

[5] Q. Zhang, C. Guo, Z. Guo, and W. Zhu, "Efficient mobility management for vertical handoff between WWAN and WLAN," *IEEE Commun. Mag.*, vol. 41, pp. 102–108, Nov. 2003.

[6] K. Egevang and P. Francis, "The IP network address translator (NAT)," IETF, RFC1631.

[7] G. Pollini, "Trends in handover design," *IEEE Commun. Mag.*, vol. 34, pp. 82–90, Mar. 1996.

[8] K. Pahlavan, P. Krishnamurthy, A. Hatami, M. Ylianttila, J. Makela, R. Pichna, and J. Vallstrom, "Handoff in hybrid mobile data networks," *IEEE Pers. Commun.*, vol. 7, pp. 34–47, Apr. 2000.

[9] M. Ylianttila, M. Pande, J. Makela, and P. Mahonen, "Optimization scheme for mobile users performing vertical handoffs between IEEE 802.11 and GPRS/EDGE networks," in *Proc. IEEE GLOBECOM*, vol. 6, Nov. 2001, pp. 3439–3443.

[10] EURESCOM Project P1013-FIT-MIP. First steps toward UMTS: Mobile IP services, a European testbed. [Online]. Available: http://www.eurescom.de/public/projects/P1000-series/p1013/default.asp

[11] J. McNair, I. Akyildiz, and M. Bender, "An inter-system handoff technique for the IMT-2000 system," in *Proc. IEEE INFOCOM*, vol. 1, Mar. 2000, pp. 208–216.

[12] C. Perkins, Ed., "IP mobility support for IPv4," IETF, RFC 3344, [Online]. Available: http://www.ietf.org/rfc/rfc3344.txt.

[13] C. Perkins *et al.*, "Mobility support in IPv6," IETF Draft, [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-mobileip-ipv6-24.txt, work in progress., June 2003.

[14] A. Valko, "Cellular IP: A new approach to Internet host mobility," *Comput. Commun. Rev.*, vol. 29, no. 1, pp. 50–65, Jan. 1999.

[15] R. Ramjee *et al.*, "HAWAII: A domain-based approach for supporting mobility in wide-area wireless networks," in *Proc. IEEE ICNP*, Nov. 1999, pp. 283–292.

[16] H. Soliman, C. Castelluccia, K. El-Malki, and L. Bellier, "Hierarchical mobile IPv6 mobility management (HMIPv6)," IETF Draft, work in progress, June 2003.

[17] F. Vakil *et al.*, "Supporting mobility for TCP with SIP," IETF Draft, work in progress, July 2001.

[18] T. Kwon *et al.*, "Mobility management for VoIP service: Mobile IP vs. SIP," *IEEE Commun. Mag.*, vol. 9, pp. 66–76, Oct. 2002.

[19] D. Funato, K. Yasuda, and H. Tokuda, "TCP-R: TCP mobility support for continuous operation," in *Proc. IEEE ICNP*, Atlanta, GA, Oct. 1997, pp. 229–236.

[20] A. C. Snoeren and H. Balakrishnan, "An end-to-end approach to host mobility," in *Proc. MOBICOM*, Boston, MA, 2000, pp. 155–166.

[21] P. Nikander *et al.*, "Homeless mobile IPv6," IETF Draft, work in progress, Feb. 2001.

[22] *IEEE 802.11WG Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specification*, IEEE standard, 1999.

[23] The Network Simulator—ns-2. [Online] http://www.isi.edu/nsnam/ns/

[24] NSWEB Home. [Online] http://www.net.uni-sb.de/~jw/nsweb/

[25] P. Vixie, Ed., "Dynamic updates in the domain name system (DNS UPDATE)," IETF, RFC2136, Apr. 1997.

[26] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for Internet applications," in *Proc. SIGCOMM*, San Diego, CA, 2001, pp. 149–160.

[27] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Inform. Theory*, vol. IT–22, pp. 644–654, 1976.

[28] R. Moskowitz, "Host identity payload architecture," IETF Draft, work in progress, Feb. 2001.

[29] Y. Rekhter, B. Moskowitz, D. Karrenberg, and G. de Groot, "Address allocation for private Internets," IETF, RFC1598.

[30] C. Fraleigh, S. Moon, C. Diot, B. Lyles, and F. Tobagi, "Packet-level traffic measurement from a Tier-1 IP backbone," Sprint ATL Tech. Rep. TR01-ATL-110 101, Burlingame, CA, Nov. 2001.

[31] NLANR. [Online] http://www.nlanr.net

[32] R. Draves, A. Mankin, and B. Zill, "Implementing IPv6 for windows NT," presented at the *2nd USENIX Windows NT Symp.*, Seattle, WA, Aug. 1998.

[33] *Wireless/Wired Wide Area Network Emulator (WiNE) User Manual Version 3.0*, Microsoft Research Asia, Beijing, China, Jan. 6, 2003.

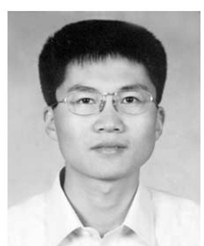
[34] F. Cali, M. Conti, and E. Gregori, "Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit," *IEEE/ACM Trans. Networking*, vol. 8, pp. 785–799, Dec. 2000.

[35] L. Kleinrock and F. Tobagi, "Packet switching in radio channels: part I," *IEEE Trans. Commun*, vol. 23, pp. 1400–1416, Dec. 1975.

[36] A. Zahedi and K. Pahlavan, "Capacity of a wireless LAN with voice and data services," *IEEE Trans. Commun.*, vol. 48, pp. 1160–1170, July 2000.

**Chuanxiong Guo** (S'99–M'01) received the Ph.D. degree in communications and information systems from the Institute of Communications Engineering, Nanjing, China, in 2000.

In 2001, he spent the year to run a research project sponsored by the National High Technology Research and Development Program of China (the 863 Program). He is now at Microsoft Research Asia, Beijing, China, as a Researcher. His current research interests include multiservice support for wired and wireless networks, mobility management, and networking support in operating systems.



**Zihua Guo** (M'01) received the B.S. and M.S. degrees (with distinction) from the University of Science and Technology of China, Anhui, in 1995 and 1998, respectively, and the Ph.D. degree from the Hong Kong University of Science and Technology, Kowloon, in August 2001.

His research interests include wireless communications and networking, multimedia communications, signal processing, etc. He is now with Microsoft Research Asia, Beijing, China.

**Qian Zhang** (M'00) received the B.S., M.S., and Ph.D. degrees in computer science from Wuhan University, Wuhan, China, in 1994, 1996, and 1999, respectively.

She joined Microsoft Research Asia, Beijing, China, in July 1999, as an Associate Researcher in the Internet Media Group and is a Researcher and project lead of the Wireless and Networking Group. She has published about 80 refereed papers in international leading journals and key conferences in the areas of wireless/Internet multimedia networking, wireless communications and networking, and overlay networking. She is the inventor of more than 12 pending patents. Her current research interest includes multimedia delivery over wireless Internet, next-generation wireless networks, P2P network/ad hoc network. Currently, she is participating in many activities with the IETF Robust Header Compression (ROHC) WG Group for TCP/IP header compression.

**Wenwu Zhu** (S'92–M'97–SM'01) received the B.E. and M.E. degrees from the National University of Science and Technology, Hefei, China, in 1985 and 1988, respectively, the M.S. degree from Illinois Institute of Technology, Chicago, in 1993, and the Ph.D. degree from Polytechnic University, Brooklyn, NY, in 1996, all in electrical engineering.

From August 1988 to December 1990, he was with the Graduate School, University of Science and Technology of China (USTC), and the Institute of Electronics, Chinese Academy of Sciences, Beijing, China. He joined Microsoft Research, Beijing, in 1999, as a Researcher in the Internet Media Group. He is currently a Research Manager of the Wireless and Networking Group. Prior to his current post, he was with Bell Laboratories, Lucent Technologies, Whippany, Holmdel, and Murray Hill, NJ, as a Member of Technical Staff during 1996–1999. While he was with Bell Laboratories, he performed research and development in the areas of Internet video, video conferencing, and video streaming over IP networks. He has published over 150 refereed papers in international leading journals and key conferences in the areas of wireless/Internet video delivery, wireless/Internet multimedia communications and networking, and multimedia signal processing, and has contributed to the IETF ROHC WG draft on robust TCP/IP header compression over wireless links. He is the inventor of more than 12 pending patents. His current research interest is in the area of wireless/Internet multimedia delivery and multimedia networking.

Dr. Zhu received the Best Paper Award in the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY in 2001. He is a Member of Eta Kappa Nu, the Visual Signal Processing and Communication Technical Committee, and the Multimedia System and Application Technical Committee of the IEEE Circuits and Systems Society. He is also a Member of the Multimedia Communication Technical Committee of the IEEE Communications Society. He has served as Guest Editor for Special Issues on Streaming Video and Wireless Video in the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. He is also a Guest Editor for the Special Issue on Advanced Video Coding and Delivery in the PROCEEDINGS OF THE IEEE. Currently, he serves as an Associate Editor for the IEEE TRANSACTIONS ON MOBILE COMPUTING and the IEEE TRANSACTIONS ON MULTIMEDIA. He is now Chairman of the IEEE Circuits and System Society Beijing Chapter.