Modeling Key Caching for Mobile IP Authentication, Authorization, and Accounting (AAA) Services

Phone Lin, *Senior Member, IEEE*, Shin-Ming Cheng, *Member, IEEE*, and Wanjiun Liao, *Senior Member, IEEE*

***Abstract*—The Mobile IP *Authentication, Authorization, and Accounting* (AAA) framework architecture is designed to protect signaling messages from eavesdropping by malicious attackers. The message exchanges for AAA incur heavy signaling overhead and long network access latency for mobility service. To solve this problem, the most typical approach is to adopt a key caching mechanism so that the authentication can locally be done and so that the signaling overhead can be significantly reduced. However, in the literature, very little work has conducted a thorough analytical study on the proposed key caching scheme. As a result, the statistical behavior of these schemes cannot be well justified. In this paper, we develop an analytical model that describes the key caching behavior in Mobile IP networks. The accuracy of this model is validated by simulations. Based on the performance analysis, we then propose an adaptive algorithm that dynamically adjusts key cache size so that the signaling overhead can be minimized.**

***Index Terms*—Analytical model, authentication, authorization, and accounting (AAA), key caching, mobile IP.**

I. Introduction

THE MOBILE IP protocol [1] has widely been adopted for user mobility management in future all-Internet-Protocol networks. To guarantee secure access to Mobile IP networks, the working group Internet Engineering Task Force defines the Mobile IP *Authentication, Authorization, and Accounting* (AAA) infrastructure [2] in RFC 2977. Fig. 1 shows the AAA framework in Mobile IP. The *home agent* [HA; see Fig. 1(a)] in the home network and the *foreign agent* [FA; see Fig. 1(b)] in the foreign network are responsible for mobility management for a *mobile node* [MN; see Fig. 1(c)]. The registration procedure starts before packets are delivered to an MN, i.e., signals

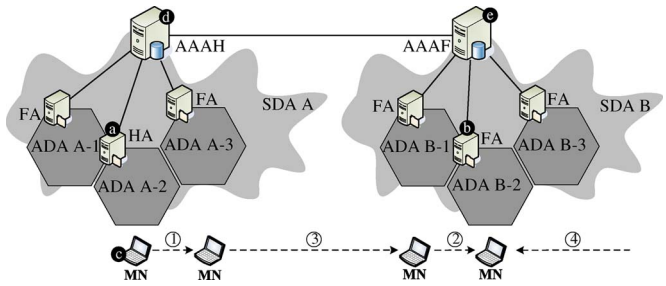

Fig. 1. Mobile IP AAA framework architecture.

are exchanged among the MN, FA, and HA to establish routing tables for future packet delivery to the MN. The details of the registration procedure are given in [1].

All security attacks (including the eavesdropping, replay, and man-in-middle attacks [3]) may be activated during the registration procedure. The Remote Authentication Dial-In User Service (RADIUS) [4] Protocol and its successor, i.e., the Diameter [5] Protocol, are designed to provide centralized management of access to networks based on the AAA concept. The RFC 4004 standard [6] proposed a Diameter application built based on the Mobile IP AAA framework to support the dynamic configuration of the security information. In the rest of this paper, we adopt the Diameter application in the Mobile IP AAA framework, where the AAA servers are introduced to provide AAA services. The AAA servers located in the home network and the foreign network are denoted by AAAH [see Fig. 1(d)] and AAAF [see Fig. 1(e)], respectively. The Diameter Protocol [5] is operated between the AAAH and the AAAF, between the HA and the AAAH, and between the FA and the AAAF to support secured message delivery among these nodes. When the network is initially configured, AAAH creates the authentication credential for each MN. Before an MN gains the network service, both the MN and the AAAH perform an authentication procedure whose details are given in Section II. When the MN moves from the home network to the foreign network, the MN must be authenticated by the AAAH, where the authentication credential of the MN is delivered from the AAAH to the AAAF.

The message exchanges for AAA incur heavy signaling overhead and long network access latency. Caching the authentication information (which is also known as key caching and enables the authentication to be locally done, i.e., the AAAF without being involved in the AAAH) is the most typical approach to reduce such signaling overhead in the literature [7]–[15]. However, most of the previous studies do not provide performance analysis for their proposed schemes, and

Manuscript received October 15, 2007; revised March 24, 2008 and July 1, 2008. Current version published August 14, 2009. The work of P. Lin was supported in part by the National Science Council (NSC), Taiwan, under Contract NSC97-2219-E-007-007, Contract NSC97-3114-E-002-004, Contract NSC96-2628-E-002-002-MY2, Contract NSC95-2221-E-002-091-MY3, Telcordia Applied Research Center Taiwan Company, Telecommunication Laboratories, Chunghwa Telecom Co., Ltd., and Excellent Research Projects of National Taiwan University under Contract 97R0062-05. The work of S.-M. Cheng was supported in part by the NSC, Taiwan, under Contract NSC-97-2219-E-002-018. The work of W. Liao was supported in part by the NSC, Taiwan, under Grant 97-2221-E-002-135-MY2. The review of this paper was coordinated by Dr. L. Chen.

P. Lin is with the Department of Computer Science and Information Engineering, the Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei 106, Taiwan (e-mail: plin@csie.ntu.edu.tw).

S.-M. Cheng and W. Liao are with the Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan (e-mail: smcheng@cc.ee.ntu.edu.tw; wjliao@cc.ee.ntu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2009.2015766

the advantages of the proposed schemes cannot be well justified. There are some other hierarchical approaches (such as MIPv4 Regional Registration (MIP-RR) [16] and Hierarchical Mobile IPv6 (HMIP) [17]) proposed to reduce signaling overhead for Mobile IP AAA. These hierarchical approaches introduce new network entities, which are not defined in the standard Mobile IP AAA architecture. The performance evaluation for these hierarchical approaches will be considered in our future work. This work focuses only on the impacts of the caching mechanism in the standard Mobile IP AAA architecture.

In this paper, we develop a general analytical model to study the key caching behavior in the AAA framework of Mobile IP networks. Specifically, we model the expected number of session keys required for an MN within the service area of an AAA server, and consequently, we can determine the expected key caching size for each MN. The accuracy of our model is validated by simulations. The results show that our model can describe the operation of key caching very well. Based on our performance study, we then propose an adaptive algorithm to dynamically adjust the cache size so that the signaling overhead can be minimized.

The rest of this paper is organized as follows: Section II describes the authentication procedure. In Section III, we present how the key caching mechanism is deployed in the Mobile IP AAA framework. In Section IV, we propose a general analytical model for the caching mechanism with variable key cache sizes. Section V studies the performance via simulations for key caching. Based on the performance study, we propose an adaptive cache-size selection algorithm to dynamically adjust the cache size in Section VI. Finally, we conclude this work in Section VII.

II. Mobile IP Authentication Procedure

In this section, we illustrate the Mobile IP authentication procedure. Through the Diameter Protocol, Mobile Security Associations (MSAs) are pre-set up between the AAAH and the AAAF, between the HA and the AAAH, and between the FA and the AAAF. The MSA between the MN and the AAAH is set up when the mobile user subscribes to the service. The MSA supports the mutual authentication for a message delivery between two network nodes. Each MSA consists of a hash algorithm, a shared session key, and an agreement on the security parameter index (SPI). The hash algorithm is used to compute keyed hashes over messages. The shared session key is the secret for the hash algorithm. The SPI indicates the type of hash algorithm and the secret and is the identifier of the MSA. More details on the MSA are presented in [1]. Let k_{x-y} denote the shared session key of network nodes x and y. To enable authentication of a message (sent from node x to node y), node x appends an authenticator to this message and then sends it to node y. Node y checks the authenticator by taking the following three actions: 1) looks up the MSA based on the SPI; 2) recomputes the keyed hash by using the shared session key k_{x-y}; and 3) verifies whether the recomputed result is equal to the content in the received authenticator. A validation timer is maintained for the shared session key k_{x-y} to prevent the shared key from exposure by malicious crackers. When the validation timer of a shared session key expires, a new session key is regenerated. The purposes of the Mobile IP AAA authentication procedure include the following: 1) to identify and authenticate an MN; 2) to update the MN's correspondent IP address in the HA; 3) to authorize an MN to use the services in the foreign network; and 4) to distribute the shared session keys $k_{\mathrm{MN-FA}}$, $k_{\mathrm{MN-HA}}$, and $k_{\mathrm{FA-HA}}$.

To simplify our description, we denote the service area of an FA (or an HA) and the service area of an AAA server as "ADA" and "SDA," respectively. One SDA may cover multiple ADAs. Suppose that an MN moves from ADA x to ADA y. As shown in Fig. 1, four cases are considered for the MN movement.

- Case 1) Intra-AAAH movement: ADA x and ADA y are within the same SDA served by the AAAH, e.g., the MN moves from ADA A-2 to ADA A-3 [see Fig. 1 (1)].
- Case 2) Intra-AAAF movement: ADA x and ADA y are within the same SDA served by the AAAF, e.g., the MN moves from ADA B-1 to ADA B-2 [see Fig. 1 (2)].
- Case 3) Inter-AAAH and AAAF movement: ADA x and ADA y are within the SDA (served by the AAAH) and the SDA (served by the AAAF), respectively, e.g., the MN moves from ADA A-3 to ADA B-1 [see Fig. 1 (3)].
- Case 4) Inter-AAAF movement: ADA x and ADA y are within different SDAs served by different AAAFs [see Fig. 1 (4)].

As an example, Fig. 2 shows the message flow for the Mobile IP AAA authentication procedure for Case 3). The Mobile IP AAA authentication procedure for Case 3) includes eight steps, the details of which are given here.

- Step 1) When an MN roams from the home network to the foreign network, it sends a Registration Request message to the FA, which contains the *network authentication identity* (NAI). A NAI consists of two parts, i.e., a user part and a realm part, and is in the form of "user@realm." The user part indicates the MN's identity, and the realm part stores the network identity of the MN's home domain.
- Step 2) Upon receipt of the Registration Request message, the FA updates its visitor list (containing the NAIs of all the MNs residing in the ADA served by the FA). Suppose that the FA is served by the AAAF. Then, the FA encapsulates the Registration Request message in the AA-Mobile-Node-Request, which is an AAAF Diameter message.
- Step 3) Upon receiving the AA-Mobile-Node-Request message, the AAAF detects that the requested MN is not in its SDA by checking the realm part of the MN's NAI. Then, the AAAF forwards the AA-Mobile-Node-Request message to the MN's AAAH according to the realm part in the NAI.
- Step 4) The AAAH checks the AA-Mobile-Node-Request message to determine whether the MN is a legal user based on the MSA between the MN and the AAAH.

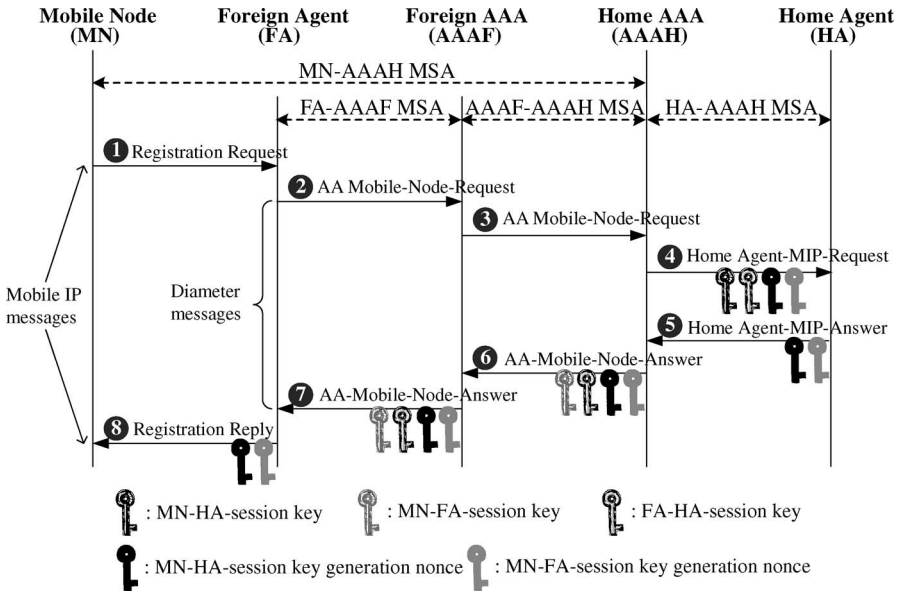

Fig. 2. Message flow for the Mobile IP AAA authentication procedure.

Then, the AAAH generates three session keys (i.e., $k_{\mathrm{MN-FA}}$, $k_{\mathrm{MN-HA}}$, and $k_{\mathrm{FA-HA}}$) and two nonces[1] (i.e., $n_{\mathrm{MN-FA}}$ and $n_{\mathrm{MN-HA}}$) for this MN.

The AAAH sends the HA a Diameter message Home-Agent-MIP-Request, which contains two session keys (i.e., $k_{\mathrm{MN-HA}}$ and $k_{\mathrm{FA-HA}}$), two nonces (i.e., $n_{\mathrm{MN-FA}}$ and $n_{\mathrm{MN-HA}}$), and the Registration Request message.

Step 5) Upon receipt of the Home-Agent-MIP-Request message, the HA extracts the Registration Request message from this message. The two session keys (i.e., $k_{\mathrm{MN-HA}}$ and $k_{\mathrm{FA-HA}}$) are used to ensure legal delivery for the signalings between the MN and the HA, and between the FA and the HA, respectively. The HA generates a Mobile IP message, i.e., Registration Reply, to encapsulate the two nonces (i.e., $n_{\mathrm{MN-FA}}$ and $n_{\mathrm{MN-HA}}$) in this message. Then, the HA sends the Registration Reply message carried in a Diameter message, i.e., Home-Agent-MIP-Answer, to the AAAH.

Step 6) and 7) Upon receiving Home-Agent-MIP-Answer, the AAAH generates a Diameter message, i.e., AA-Mobile-Node-Answer, which encapsulates the Registration Reply message and the two session keys (i.e., $k_{\mathrm{MN-FA}}$ and $k_{\mathrm{FA-HA}}$) obtained at step 4). Then, the AAAH sends the AA-Mobile-Node-Answer message to the AAAF. The AAAF forwards the received AA-Mobile-Node-Answer message to the FA.

Step 8) Upon receipt of the AA-Mobile-Node-Answer message, the FA retrieves the two session keys (i.e., $k_{\mathrm{MN-FA}}$ and $k_{\mathrm{FA-HA}}$) from this message. The two keys are for the secured message delivery between the MN and the FA and between the FA and the HA, respectively. Then, the FA sends the MN the Registration Reply message, which contains two nonces $n_{\mathrm{MN-FA}}$ and $n_{\mathrm{MN-HA}}$.

After receiving Registration Reply, the MN uses two nonces $n_{\mathrm{MN-FA}}$ and $n_{\mathrm{MN-HA}}$ to derive two session keys $k_{\mathrm{MN-FA}}$ and $k_{\mathrm{MN-HA}}$. The two session keys are used to secure the message delivery between the MN and the FA and between the MN and the HA, respectively. The MN starts a validation timer for the two session keys. When the validation timer expires or the MN moves to another ADA, the Mobile IP AAA authentication procedure is executed to get a new session key set.

After the execution of the Mobile IP AAA authentication procedure, the signaling exchanges for the Mobile IP protocol among the MN, FA, and HA are secured. For Case 1) (i.e., intra-AAAH movement), steps 1), 3), 4), 5), 6), and 8) are executed. For Cases 2) and 4) (i.e., intra-AAAF and inter-AAAF movements), steps 1)–8) are executed.

We note that, in this procedure, when the validation timer of a session key set expires or the MN moves to another ADA, the Mobile IP AAA authentication procedure should be executed to get a new session key set from AAAH. This procedure incurs extra signaling overhead to the Mobile IP network. The key caching mechanism (i.e., the cache session key sets at the AAAH and HA), which enables the MN to be authenticated through the AAAF without involving the AAAH, can be applied to reduce the signaling overhead. In the rest of this paper, the authentication key caching mechanism is abbreviated as the AKC mechanism. In the next section, we show how the AKC mechanism operates in the Mobile IP AAA framework by slightly modifying the authentication procedure.

[1]Nonces $n_{\mathrm{MN-FA}}$ and $n_{\mathrm{MN-HA}}$ are used to generate session keys $k_{\mathrm{MN-FA}}$ and $k_{\mathrm{MN-HA}}$, respectively. An MN will get these two nonces in the Registration Reply message and derive the corresponding session key by the shared session key in the MN-AAAH MSA (18).

III. AKC Mechanism

We discuss the operation of the AKC mechanism in two cases.

Case 1) *The MN moves into a new SDA, or all cached session key sets are run out.* Steps 4)–7) in the authentication procedure are modified as follows: At step 4), the AAAH generates K session key sets and K nonce sets $(k_{\mathrm{MN-FA},1}, k_{\mathrm{MN-HA},1}, k_{\mathrm{FA-HA},1})(n_{\mathrm{MN-FA},1}, n_{\mathrm{MN-HA},1})$, $(k_{\mathrm{MN-FA},2}, k_{\mathrm{MN-HA},2}, k_{\mathrm{FA-HA},2})(n_{\mathrm{MN-FA},2}, n_{\mathrm{MN-HA},2}), \ldots, (k_{\mathrm{MN-FA,\ K}}, k_{\mathrm{MN-HA,K}}, k_{\mathrm{FA-HA,K}})\ (n_{\mathrm{MN-FA,K}}, n_{\mathrm{MN-HA,K}})$ for the MN. Then, the AAAH sends the HA a Home-Agent-MIP-Request message containing K two-key sets and two-nonce sets $(k_{\mathrm{MN-HA},1},\ k_{\mathrm{FA-HA},1})\ (n_{\mathrm{MN-FA},1},\ n_{\mathrm{MN-HA},1})$, $(k_{\mathrm{MN-HA},2}, k_{\mathrm{FA-HA},2})(n_{\mathrm{MN-FA},2}, n_{\mathrm{MN-HA},2}), \ldots, (k_{\mathrm{MN-HA,K}},\ k_{\mathrm{FA-HA,K}})\ (n_{\mathrm{MN-FA,K}}, n_{\mathrm{MN-HA,K}})$. At step 5), upon receipt of the Home-Agent-MIP-Request message from the AAAH, the HA caches the K two-key sets and encapsulates K two-nonce sets in the Registration Reply message. At step 6), the AAAH sends the AAAF the AA-Mobile-Node-Answer message containing the Registration Reply message and K two-key sets $(k_{\mathrm{MN-FA},1}, k_{\mathrm{FA-HA},1}), (k_{\mathrm{MN-FA},2}, k_{\mathrm{FA-HA},2}), \ldots, (k_{\mathrm{MF-HA,K}}, k_{\mathrm{FA-HA,K}})$. Note that, to enable the authentication to be locally done in the AAAF, there should be an MSA existing between the AAAF and the MN. This MSA can be established by sending all related information (which is known as the local authenticator [14]) for the MSA (between the AAAH and the MN) to the AAAF at step 6). At step 7), upon receipt of an AA-Mobile-Node-Answer message, the AAAF caches K two-key sets and K two-nonce sets.

Case 2) *If there are valid cached session key sets, the MN moves from one ADA to another belonging to the same SDA, or the validation timer of the currently used session key set expires*. When the validation timer of the currently used session key set expires, only steps 1), 2), 7), and 8) are performed for the local authentication through the AAAF. At the end of step 2), after the AAAF receives an AA-Mobile-Node-Request message, the AAAF authenticates the MN by using the MSA between the MN and the AAAF. At step 7), the AAAF replies to the FA with an AA-Mobile-Node-Answer message containing a cached two-key set. When the MN moves from one ADA to another belonging to the same SDA, in addition to steps 1), 2), 7), and 8), the FA and the HA exchange two Mobile IP messages, i.e., Registration Request and Registration Reply, between steps 7) and 8) to update the MN's location information in the HA.

Note that, at step 6) of Case 1), the transmission of the local authenticator is protected by the MSA between the AAAH and the AAAF, which guarantees information secrecy and data integrity. The replay attack can be defeated [13]. The implementation of the transmission of the local authenticator over the MSA is out of the scope of this paper and has been well treated in [13] and [14]. Moreover, AKC does not modify the execution flow of the authentication procedure in the MN and does not introduce any extra memory and computation overhead in the MN.

IV. Analytical Model for Key Caching With Cache Size K

In this section, we propose an analytical model for AKC with cache size K. The notation used in this paper is summarized as follows:

$C(K)$	function for estimating the total bandwidth consumption for AKC with cache size K when an MN resides in an SDA;
$C_{r,i}$	counter in an MN for counting the number of occurrences of event i when the MN resides in an SDA;
$c_{f,i}$ $(c_{s,i})$	bandwidth consumption for event i if the key cache is empty (nonempty);
$E[N]$	expected number of N for AKC with key cache size K;
$f_A(t_A)$	density function for t_A;
$f_A^*(s)$	Laplace transform of $f_A(t_A)$;
$f_a(t_{a,i})$	density function for $t_{a,i}$;
$f_a^*(s)$	Laplace transform of $f_a(t_a)$;
$f_l(t_l)$	density function for t_l;
$f_v(t_v)$	density function for t_v;
K	number of session key sets in a key cache;
$K(j)$	K value selected by the automatic K-selection mechanism when the MN resides in the jth SDA;
k_{x-y}	shared session key between two network nodes x and y;
$L(K)$	function for measuring the average handoff latency for AKC with cache size K when an MN resides in an SDA;
N	number of key-set retrievals from AAAH when the MN resides in an SDA;
M_i	number of occurrences of event i when the MN resides in an SDA;
$\Pr[N=n]$	probability that, with cache size K, there are n key-set retrievals from the AAAH to the AAAF when the MN resides in an SDA;
$r_a(\tau_a)$	density function for τ_a;
$S(r)$	total number of states for an r-layer SDA random walk;
t_A	period when an MN resides in an SDA;
$t_{a,i}$	residence time for an MN at ADA i;
$t_{f,i}$ (t_{s_i})	handoff latency for event i if the key cache is empty (nonempty);
t_l	lifetime of a session key set;
t_v	length of the validation time of a session key set;
v_a	variance of Gamma-distributed ADA residence time;
α	shape parameter of Gamma-distributed ADA residence time;
β	size of a Mobile IP message;

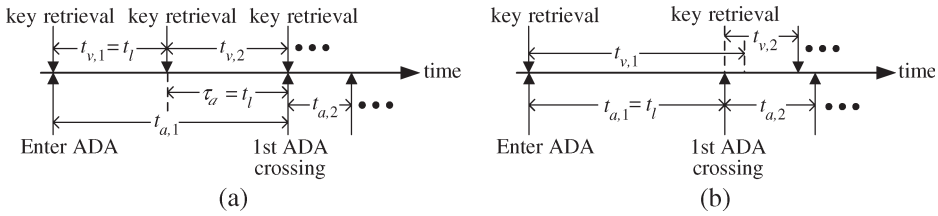

Fig. 3. Timing diagram for t_v, t_a, and t_l. (a) $t_v < t_a$ and $t_v \geq \tau_a$. (b) $t_v \geq t_a$.

γ	size of a Diameter message;
δ	size of a session key;
$1/\eta_a$	expected ADA residence time;
$\Theta(n, K, \tau)$	probability that, by using AKC with cache size K, the MN gets new session key sets from the AAAH n times for a specific period of τ;
$1/\mu_v$	expected validation time of a session key set;
τ_a	time period between the time when the MN gets a session key set and the time when the MN moves to the next ADA.

Suppose that the validation time of a session key set t_v is exponentially distributed with the density function, i.e.,

$$f_v(t_v) = \mu_v e^{-\mu_v t_v} \tag{1}$$

and mean $E[t_v] = (1/\mu_v)$. Let $t_{a,i}$ be the residence time for an MN at ADA i. $t_{a,i}$ are assumed to be exponential independent identically distributed random variables with density function

$$f_a(t_{a,i}) = \eta_a e^{-\eta_a t_{a,i}} \tag{2}$$

and mean $E[t_{a,i}] = (1/\eta_a)$. Note that the exponential assumption may not approximate the ADA residence time very well. However, later in Section V, we will show via simulations that the performance of AKC is independent of the distribution of the ADA residence time, and our exponential distribution assumption can be justified. For a homogeneous Mobile IP network, we have, for $i \neq j$

$$f_a(t_{a,i}) = f_a(t_{a,j}) = f_a(t_a). \tag{3}$$

Let t_l be the lifetime of a session key set. Consider Fig. 3(a). After the MN enters an ADA, the Mobile IP AAA authentication procedure is executed to retrieve a session key set. If the validation timer of this session key set expires before the MN moves to another ADA (i.e., $t_v < t_a$), then the lifetime t_l of this session key set is equal to the validation period of this session key set (i.e., $t_l = t_v$). Afterward, the MN gets a new session key set. Let τ_a be the time period between the time when the MN gets a session key set and the time when the MN moves to the next ADA. Then, from [19], the density function $r_a(\tau_a)$ for the distribution of τ_a can be obtained as follows:

$$r_a(\tau_a) = \eta_a \int_{t_a=\tau_a}^{\infty} f_a(t_a) = \eta_a \left[1 - F_a(t_a)\right]\Big|_{t_a=\tau_a}$$

where F_a is the distribution function of t_a. Since t_a is exponentially distributed, τ_a and t_a have the same distribution, i.e., $r_a(\tau_a) = \eta_a e^{-\eta_a \tau_a}$.

If the MN moves to another ADA before the validation timer of the session key set expires, then the lifetime of the session key set is equal to τ_a (i.e., $t_l = \tau_a$). If the MN moves to another ADA before the validation timer of the session key set expires, as shown in Fig. 3(b), then the lifetime of the session key set is equal to t_a (i.e., $t_l = t_a$). Then, we have $t_l = \min(t_v, t_a)$ or $t_l = \min(t_v, \tau_a)$. Since τ_a and t_a have the same distribution, we obtain $t_l = \min(t_v, t_a)$. From [20], the density function $f_l(t_l)$ of t_l can be obtained by

$$f_l(t_l) = f_v(t_l)\left[1 - F_a(t_l)\right] + f_a(t_l)\left[1 - F_v(t_l)\right] \tag{4}$$

where F_v is the distribution function of t_v. From (1) and (2), (4) can be rewritten as

$$f_l(t_l) = (\mu_v + \eta_a) e^{-(\mu_v + \eta_a) t_l}. \tag{5}$$

Let t_A denote the time period when an MN resides in an SDA. Suppose that, during $t_A^{(m)}$, the MN visits m ADAs (i.e., the MN moves m steps) and that the MN resides in ADA i for a period $t_{a,i}$. Then, $t_A^{(m)} = t_{a,1} + t_{a,2} + \cdots + t_{a,m}$ has density function

$$\begin{aligned} & f_A^{(m)}\left(t_A^{(m)}\right) \\ &= \int_{t_{a,1}=0}^{t_A^{(m)}} \int_{t_{a,2}=0}^{t_A^{(m)} - t_{a,1}} \cdots \int_{t_{a,m-1}=0}^{t_A^{(m)} - t_{a,1} - \cdots - t_{a,m-2}} f_a(t_{a,1}) \\ &\quad \times f_a(t_{a,2}) \cdots f_a(t_{a,m-1}) \\ &\quad \times f_a\left(t_A^{(m)} - t_{a,1} - \cdots - t_{a,m-1}\right) dt_{a,m-1} \cdots dt_{a,1} \\ &= \int_{t_{a,1}=0}^{t_A^{(m)}} \int_{t_{a,2}=0}^{t_A^{(m)} - t_{a,1}} \cdots \int_{t_{a,m-1}=0}^{t_A^{(m)} - t_{a,1} - \cdots - t_{a,m-2}} \left(\prod_{i=1}^{m-1} \eta_a e^{-\eta_a t_{a,i}}\right) \\ &\quad \times \eta_a e^{-\eta_a \left(t_A^{(m)} - t_{a,1} - \cdots - t_{a,m-1}\right)} dt_{a,m-1} \cdots dt_{a,1}. \end{aligned} \tag{6}$$

Our analytical model considers a uniform random walk model for the MN's movement, which follows a regular SDA/ADA overlay structure, as shown in Fig. 4. This structure has widely been adopted to simulate the wireless mobile networks in several studies [21]–[25]. In this configuration, the ADAs are grouped into several r-layer SDAs. Each SDA covers $3r^2 - 3r + 1$ ADAs. Fig. 4 shows four-layer SDAs. There are seven SDAs (A, B, C, D, E, F, and G) and ADAs within the SDAs. The ADA at the center of the SDA is called layer-0 ADA. The ADAs that surround layer $x - 1$ ADAs are called layer-x ADAs. There

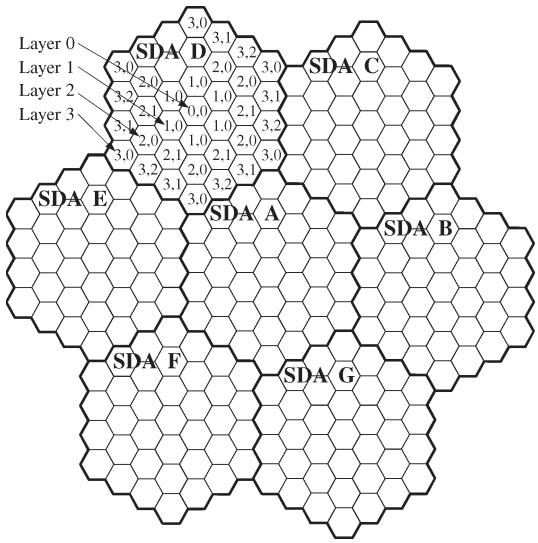

Fig. 4. SDA layout.

are $6x$ ADAs in layer x, except that exactly one ADA is in layer 0. An r-layer SDA overlays ADA from layer 0 to layer $r-1$.

Based on the equal routing probability assumption (i.e., the MN moves to each of the neighboring ADAs with probability 1/6), following the study [21], the ADAs in an SDA can be classified into different types. The type format of an ADA is $\langle x, y\rangle$, where "x" indicates that the ADA is in layer x, and "y" represents the $(y+1)$th type in layer x. The type of ADAs in Fig. 4 is for a four-layer SDA. Based on the random walk model, we derive the time when an MN crosses the boundary of an r-layer SDA. In the model, state (x, y) indicates that the MN is in one of the ADAs of type $\langle x, y\rangle$, where $0 \le x < r$ and $0 \le y \le x-1$. State (r, j) indicates that the MN leaves the SDA from state $(r-1, j)$, where $0 \le j \le r-1$. Let $S(r)$ be the total number of states for an r-layer SDA random walk. Then, $S(r) = (r(r+1)/2)$.

Let $p_{(x,y),(x',y')}$ be the one-step transition probability from state (x, y) to state (x', y'), i.e., the probability that the MN moves from a $\langle x, y\rangle$ ADA to a $\langle x', y'\rangle$ ADA in one step. Let $\mathbf{P} = (p_{(x,y),(x',y')})$ be the transition matrix of this random walk model. From [21], $\mathbf{P}$ is an $S(r) \times S(r)$ matrix, which is expressed as

$$\mathbf{P} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1/6 & 1/3 & 1/6 & 1/3 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1/6 & 0 & 1/3 & 1/6 & \cdots & 0 & 0 & 0 \\ 0 & 1/3 & 1/3 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{pmatrix}_{S(r)\times S(r)}. \tag{7}$$

We use the Chapman–Kolmogorov equation [26] to compute the probability for the number of steps that an MN takes from an SDA to another. For $m \ge 1$, let

$$\mathbf{P}^{(m)} = \begin{cases} \mathbf{P}, & \text{if } m = 1 \\ \mathbf{P} \times \mathbf{P}^{(m-1)}, & \text{if } m > 1. \end{cases} \tag{8}$$

An element $p^{(m)}_{(x,y),(x',y')}$ in $\mathbf{P}^{(m)}$ is the probability that the random walk moves from state (x, y) to state (x', y') with exactly m steps. Define $p_{m,(x,y),(r,j)}$ as the probability that an MN initially resides in a $\langle x, y\rangle$ ADA, moves into a $\langle r-1, j\rangle$ ADA at the $(m-1)$th step, and then leaves the SDA at the mth step. Then, $p_{m,(x,y),(r,j)}$ can be expressed as, for $0 \le j < r-1$

$$p_{m,(x,y),(r,j)} = \begin{cases} p_{(x,y),(r,j)}, & \text{for } m = 1 \\ p^{(m)}_{(x,y),(r,j)} - p^{(m-1)}_{(x,y),(r,j)}, & \text{for } m > 1. \end{cases} \tag{9}$$

Equation (9) can be solved using transition probability matrix (7) and (8). Let $q_{(r-1,j)}$ be the probability that an MN enters the SDA through a $\langle r-1, j\rangle$ ADA at the first step. $q_{(r-1,j)}$ can be computed from $p_{m,(x,y),(r,j)}$ [22]. More details are given in [22]. For example, for a six-layer SDA, we have $q_{(5,0)} = 27.27\%$ and $q_{(5,j)} = 18.18\%$ for $1 \le j \le 4$. $q_{(r-1,y)}p_{m,(r-1,y),(r,j)}$ is the probability that an MN enters an SDA through a $\langle r-1, y\rangle$ ADA at the first step, moves into a $\langle r-1, j\rangle$ ADA at the $(m-1)$st step, and then leaves the SDA at the mth step. Thus, using (7), the density function $f_A(t_A)$ for the MN residence time in an r-layer SDA, where $r > 1$, is

$$f_A(t_A) = \sum_{m=1}^{\infty}\sum_{y=0}^{r-2}\sum_{j=0}^{r-2} q_{(r-1,y)}p_{m,(r-1,y),(r,j)} f_A^{(m)}\left(t_A^{(m)}\right). \tag{10}$$

Let $f_a^*(s)$ be the Laplace transform of $f_a(t_a)$. Then, from (7) and the Laplace transform convolution rule, the Laplace transform $f_A^{(m)*}(s)$ for $f_A^{(m)}(\cdot)$ can be computed as follows:

$$f_A^{(m)*}(s) = [f_a^*(s)]^m. \tag{11}$$

From (10) and (11), the Laplace transform of $f_A(t_A)$ is

$$\begin{aligned} f_A^*(s) &= \sum_{m=1}^{\infty}\sum_{y=0}^{r-2}\sum_{j=0}^{r-2} q_{(r-1,y)}p_{m,(r-1,y),(r,j)} f_A^{(m)*}(s) \\ &= \sum_{m=1}^{\infty}\sum_{y=0}^{r-2}\sum_{j=0}^{r-2} q_{(r-1,y)}p_{m,(r-1,y),(r,j)} [f_a^*(s)]^m \end{aligned} \tag{12}$$

where $r > 1$. From (3), since t_a is exponentially distributed, Laplace transform $f_a^*(s)$ is

$$f_a^*(s) = \frac{\eta_a}{\eta_a + s}. \tag{13}$$

Applying (13) into (12), (12) is rewritten as

$$f_A^*(s) = \sum_{m=1}^{\infty}\sum_{y=0}^{r-2}\sum_{j=0}^{r-2} q_{(r-1,y)}p_{m,(r-1,y),(r,j)} \left(\frac{\eta_a}{\eta_a + s}\right)^m \tag{14}$$

where $r > 1$.

As discussed in Section III, if there are valid cached session key sets, the cached key set in the AAAF supports the local authentication for an MN without involving the AAAH when the MN moves from one ADA to another belonging to the same SDA or the validation timer of the currently used key set

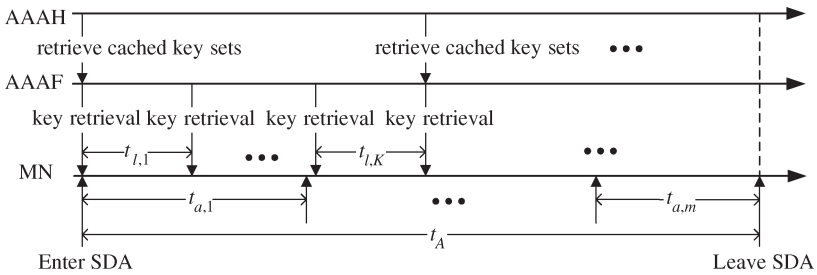

Fig. 5. Timing diagram for the AKC algorithm.

expires. On the other hand, if all cached session key sets are run out or the MN moves into a new SDA, the full Mobile IP AAA authentication procedure is executed to get new session key sets (which will be cached in the AAAF and the HA) from the AAAH. Fig. 5 shows the timing diagram for an MN entering an SDA and then leaving the SDA.

For a specific period τ, let $\Theta(n, K, \tau)$ be the probability that, in AKC with cache size K, the MN gets new session key sets (to be cached in the AAAF) from the AAAH n times. If $(n-1)K+k$ (where $1 \le k \le K$) local authentications are performed, then the key-set retrievals from the AAAH (where each key-set retrieval contains K session key sets) should be executed n times. Then, from [20] and using (4), $\Theta(n, K, \tau)$ can be computed using

$$\Theta(n,K,\tau) = \sum_{k=1}^{K} \left\{ \frac{[(\mu_v+\eta_a)\tau]^{(n-1)K+k}}{[(n-1)K+k]!} \right\} e^{-(\mu_v+\eta_a)\tau}. \tag{15}$$

Let N be the number of session key-set retrievals from the AAAH, whereas the MN resides in an SDA and the cache size is K. Let $\Pr[N=n]$ be the probability that, with cache size K, there are n key-set retrievals from the AAAH to the AAAF. $\Pr[N=n]$ can be derived as follows:

$$\Pr[N=n] = \int_{t_A=0}^{\infty} \Theta(n,K,t_A) f_A(t_A) dt_A. \tag{16}$$

Applying (15) into (16), we have

$$\begin{aligned}
\Pr[N=n] &= \int_{t_A=0}^{\infty} \sum_{k=1}^{K} \left\{ \frac{[(\mu_v+\eta_a)t_A]^{(n-1)K+k}}{[(n-1)K+k]!} \right\} \\
&\quad \times e^{-(\mu_v+\eta_a)t_A} f_A(t_A) dt_A \\
&= \sum_{k=1}^{K} \left\{ \frac{(\mu_v+\eta_a)^{(n-1)K+k}}{[(n-1)K+k]!} \right\} \\
&\quad \times \int_{t_A=0}^{\infty} t_A^{(n-1)K+k} f_A(t_A) e^{-(\mu_v+\eta_a)t_A} dt_A \\
&= \sum_{k=1}^{K} \left\{ \frac{(\mu_v+\eta_a)^{(n-1)K+k}}{[(n-1)K+k]!} \right\} (-1)^{(n-1)K+k} \\
&\quad \times \left[\frac{d^{(n-1)K+k} f_A^*(s)}{ds^{(n-1)K+k}} \right] \Bigg|_{s=\mu_v+\eta_a}.
\end{aligned} \tag{17}$$

From Appendix A, we have

$$\begin{aligned}
\frac{d^{(n-1)K+k} f_A^*(s)}{ds^{(n-1)K+k}} &= \sum_{m=1}^{\infty} \sum_{y=0}^{r-2} \sum_{j=0}^{r-2} q_{(r-1,y)} p_{m,(r-1,y),(r,j)} \\
&\quad \times \left[\frac{(-1)^{(n-1)K+k} \eta_a^m [m+(n-1)K+k-1]!}{(\eta_a+s)^{m+(n-1)K+k}(m-1)!} \right].
\end{aligned} \tag{18}$$

Applying (19) to (17), we have

$$\begin{aligned}
\Pr[N=n] &= \sum_{k=1}^{K} \sum_{m=1}^{\infty} \sum_{y=0}^{r-2} \sum_{j=0}^{r-2} q_{(r-1,y)} p_{m,(r-1,y),(r,j)} \\
&\quad \times \left\{ \frac{(\mu_v+\eta_a)^{(n-1)K+k}}{[(n-1)K+k]!} \right\} \\
&\quad \times \left\{ \frac{\eta_a^m [m+(n-1)K+k-1]!}{(\mu_v+2\eta_a)^{m+(n-1)K+k}(m-1)!} \right\} \\
&= \sum_{k=1}^{K} \sum_{m=1}^{\infty} \sum_{y=0}^{r-2} \sum_{j=0}^{r-2} q_{(r-1,y)} p_{m,(r-1,y),(r,j)} \\
&\quad \times \frac{[m+(n-1)K+k-1]!}{[(n-1)K+k]!(m-1)!} \\
&\quad \times \left[\frac{\eta_a^m (\mu_v+\eta_a)^{(n-1)K+k}}{(\mu_v+2\eta_a)^{m+(n-1)K+k}} \right].
\end{aligned} \tag{19}$$

Then, the expected number of N for AKC with key cache size K can be obtained by

$$E[N] = \sum_{n=1}^{\infty} n \Pr[N=n]. \tag{20}$$

The limitation of the proposed analytical model is that the precise total transmission cost and average transmission time for authentication message exchanges cannot be calculated. The main purpose of our analytical model is to model the user mobility behavior in the Mobile IP AAA framework so that it can be used to study the transmission cost performance under different network settings. The analytical model is validated against the simulation experiments. Fig. 6 shows $E[N]$ against K with various μ_v/η_a ratios for the analytical model and the simulation results. The details of the parameter settings will be described in the next section. This figure validates the accuracy of our analytical model via simulations.

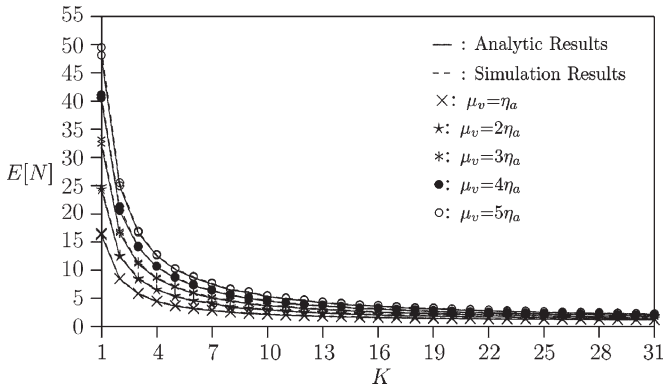

Fig. 6. Comparison between the analytical and simulation results ($\alpha = 1$, $\beta = 3\delta$, $\gamma = 12\delta$, and $r = 5$).

V. Performance of AKC With Cache Size K

The signaling cost associated with location updates may become very significant, particularly when the number of MNs is large. This problem is regarded as an important issue in many previous works, e.g., [27] and [28]. In this section, we conduct simulations to investigate the performance of AKC with cache size K in terms of bandwidth consumption (for total message exchanges of the Mobile IP authentication procedure executed when an MS moves around an SDA) and the handoff latency (i.e., the total transmission time for the whole message exchanges of the Mobile IP AAA authentication procedure executed when an MN moves from an ADA to another ADA).

The simulation technique used in this paper is the event-driven approach based on the random walk model, which has widely been used in the wireless network studies [23], [29]. In the simulation, the Mobile IP network is modeled as a regular SDA/ADA overlay structure, as shown in Fig. 4. In each simulation, the MN starts moving from an arbitrary ADA, resides in this ADA for a time period t_a, and then moves to one of its neighbors with probability 1/6. We simulate 10 000 000 user movements in each run to ensure the convergence of the simulation results.

We maintain two counters C_s and C_r to count the number of SDAs (which the MN has crossed) and key sets (that are retrieved from the AAAH by the MN), respectively. Initially, C_s and C_r are set to 0. Consider an arbitrary MN movement, where the MN moves from ADA_o to ADA_n. If ADA_o and ADA_n belong to different SDAs, counter C_s is increased by one. Each time the MN moves into a new ADA or the expiration timer of the session key set expires, the MN requests a new session key set, and we increase counter C_r by one. Then, the expected number $E[N_r]$ of the session key sets retrieved by the MN when it resides in an SDA can be computed by $E[N_r] = (C_r/C_s)$. With cache size K, the expected number $E[N]$ of session key-set retrievals from the AAAH when the MN stays in an SDA is obtained by $E[N_r]/K$.

Let β, γ, and δ be the sizes of a Mobile IP message, a Diameter message, and a shared key, respectively. Let message transmission time t_{x-y} be the time length between the time when node x transmits a message to node y and the time when the message is received by node y. The bandwidth consumption and handoff latency for Mobile IP AAA are discussed with four events.

Event 1) Intra-AAAH movement: The MN moves between two ADAs served by the AAAH. In this event, the AAAH directly authenticates the MN, and six messages are delivered, which includes one Mobile IP message (i.e., step 1) in Fig. 2), one Mobile IP message carrying two nonces (i.e., step 8) in Fig. 2), one Diameter message (i.e., step 3) in Fig. 2), one Diameter message carrying two nonces (i.e., step 5) in Fig. 2), and two Diameter messages carrying two shared session keys and two nonces (i.e., steps 4) and 6) in Fig. 2). The bandwidth consumption for this event is $2\beta + 4\gamma + 12\delta$, and the handoff latency for this event is $2(t_{\text{MN-HA}} + t_{\text{HA-AAAH}})$.

Event 2) Intra-AAAF movement: The MN moves between two ADAs served by the same AAAF. In this event, if the cache in the AAAF is not empty, the MN can locally be authenticated (i.e., by the AAAF), and steps 1), 2), 7), and 8) in the authentication procedure are executed. In addition, the FA and the HA exchange Registration Request and Registration Reply messages to update the MN's location information. Totally, six steps are performed, which includes three Mobile IP messages (i.e., step 1) in Fig. 2 and one pair of Registration Request and Registration Reply messages), one Mobile IP message carrying two nonces (i.e., step 8) in Fig. 2), one Diameter message (i.e., step 2) in Fig. 2), and one Diameter message carrying two shared session keys and two nonces (i.e., step 7) in Fig. 2). The bandwidth consumption is $4\beta + 2\gamma + 6\delta$, and the handoff latency is $2(t_{\text{MN-FA}} + t_{\text{FA-AAAF}} + t_{\text{FA-HA}})$.

If the cache is empty, the MN has to execute the full authentication procedure, which includes one Mobile IP message (i.e., step 1) in Fig. 2), one Mobile IP message carrying two nonces (i.e., step 8) in Fig. 2), two Diameter message (i.e., steps 2) and 3) in Fig. 2), two Diameter messages carrying K sets of two session keys and two nonces (i.e., steps 4) and 6) in Fig. 2), one Diameter message carrying K sets of two nonces (i.e., step 5) in Fig. 2), and one Diameter message carrying two shared session keys and two nonces (i.e., step 7) in Fig. 2). The bandwidth consumption is $2\beta + 6\gamma + \delta(10K + 6)$, and the handoff latency is $2(t_{\text{MN-FA}} + t_{\text{FA-AAAF}} + t_{\text{AAAF-AAAH}} + t_{\text{AAAH-HA}})$.

Event 3) Inter-AAA server movement: The MN moves between ADAs served by different AAA servers. Each time the MN moves into a new SDA, the full authentication procedure is executed, the bandwidth consumption is $2\beta + 6\gamma + \delta(10K + 6)$, and the handoff latency is $2(t_{\text{MN-FA}} + t_{\text{FA-AAAF}} + t_{\text{AAAF-AAAH}} + t_{\text{AAAH-HA}})$.

TABLE I
BANDWIDTH CONSUMPTION AND TRANSMISSION LATENCY OF EVENT i FOR THE AUTHENTICATION PROCEDURE WITH AKC (CACHE SIZE K)

i	θ	Bandwidth consumption	Transmission latency
1	1	$c_{s,i} = 2\beta + 4\gamma + 12\delta$	$t_{s,i} = 2(t_{MN-HA} + t_{HA-AAAH})$
	0	$c_{f,i} = 2\beta + 4\gamma + 12\delta$	$t_{f,i} = 2(t_{MN-HA} + t_{HA-AAAH})$
2	1	$c_{s,i} = 4\beta + 2\gamma + 6\delta$	$t_{s,i} = 2(t_{MN-FA} + t_{FA-AAAF} + t_{FA-HA})$
	0	$c_{f,i} = 2\beta + 6\gamma + (10K + 6)\delta$	$t_{f,i} = 2(t_{MN-FA} + t_{FA-AAAF} + t_{AAAF-AAAH} + t_{AAAH-HA})$
3	1	$c_{s,i} = 2\beta + 6\gamma + (10K + 6)\delta$	$t_{s,i} = 2(t_{MN-FA} + t_{FA-AAAF} + t_{AAAF-AAAH} + t_{AAAH-HA})$
	0	$c_{f,i} = 2\beta + 6\gamma + (10K + 6)\delta$	$t_{f,i} = 2(t_{MN-FA} + t_{FA-AAAF} + t_{AAAF-AAAH} + t_{AAAH-HA})$
4	1	$c_{s,i} = 2\beta + 2\gamma + 6\delta$	N/A
	0	$t_{s,i} = 2\beta + 6\gamma + (10K + 6)\delta$	N/A

Event 4) The validation timer of the session key set expires. If the cache is empty, the MN exercises the full authentication procedure. The bandwidth consumption is $2\beta + 6\gamma + \delta(10K + 6)$. If the key cache is not empty, the MN can be authenticated and locally get a new session key set (i.e., from AAAF), which includes one Mobile IP message (i.e., step 1) in Fig. 2), one Mobile IP message carrying two nonces (i.e., step 8) in Fig. 2), one Diameter message (i.e., step 2) in Fig. 2), and one Diameter message carrying two shared session keys and two nonces (i.e., step 7) in Fig. 2). The bandwidth consumption is $2\beta + 2\gamma + 6\delta$. In this scenario, there are no MN movements between different ADAs, and there is no handoff latency.

We use functions $C(K)$ and $L(K)$ to measure the bandwidth consumption and the average handoff latency, respectively, for the case where the MN moves around an SDA, where AKC with a fixed cache size of K is applied in the Mobile IP AAA authentication procedure. Let M_i denote the number of occurrences of event i that an MN encounters when the MN resides in an SDA.

Let $c_{s,i}$ and $t_{s,i}$ ($c_{f,i}$ and $t_{f,i}$) be the bandwidth consumption and handoff latency, respectively, for the Mobile IP AAA procedure when the MN encounters event i and the key cache is nonempty (empty), respectively. From the preceding discussion, Table I lists the values of $c_{s,i}$, $c_{f,i}$, $t_{s,i}$, and $t_{f,i}$ for event i. $C(K)$ and $L(K)$ functions for AKC with cache size K are expressed as follows:

$$C(K) = \sum_{i=1}^{4} \sum_{j=1}^{M_i} [\theta c_{s,i} + (1-\theta) c_{f,i}]$$

$$L(K) = \sum_{i=1}^{3} \sum_{j=1}^{M_i} [\theta t_{s,i} + (1-\theta) t_{f,i}]$$

where

$$\theta = \begin{cases} 1, & \text{if cache is nonempty} \\ 0, & \text{if cache is empty.} \end{cases}$$

In our simulation, the ADA residence time is assumed to be a Gamma distribution with mean $1/\eta_a$ and variance $v_a = (1/\eta_a^2\alpha)$ (where α is the shape parameter). The Gamma distribution has been used to well approximate many other distributions [22]–[24], [29]. To simplify our discussion, we normalize β and γ by δ. Typically [1], $\beta = 3\delta$, and $\gamma = 12\delta$. The impacts of several input parameters of the performance are discussed here.

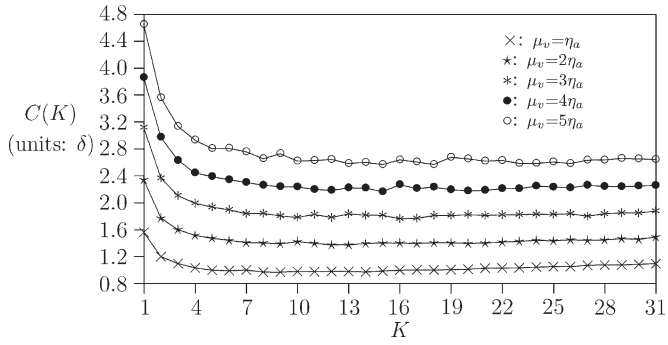

Fig. 7. Effects of μ_v/η_a and K on $C(K)$ ($\alpha = 1$, $\beta = 3\delta$, $\gamma = 12\delta$, and $r = 5$).

Effects of K and ratio μ_v/η_a on $E[N]$. As shown in Fig. 6, $E[N]$ increases as μ_v/η_a increases. A larger μ_v/η_a implies that the expiration timer of shared session keys is shorter, and the MN has more chances to change the key. Thus, there are more key sets retrieved from the AAAH. Fig. 6 also shows that $E[N]$ is a decreasing function of K, which implies that, as the key cache size increases, the MN has better chance of being locally authenticated. When $K \geq 16$, this phenomenon becomes insignificant.

Effects of K and ratio μ_v/η_a on $C(K)$. Fig. 7 shows $C(K)$ as functions of K and μ_v/η_a. This figure shows that, as K increases, $C(K)$ decreases and then slightly increases. As K increases, we observe two facts.

Fact 1) The number of session key-set retrievals from the AAAH are decreased (i.e., $E[N]$), which causes smaller $C(K)$ values.

Fact 2) More bandwidths are consumed to deliver the multiple key sets from the AAAH, which increases the cost of $C(K)$.

When K is small, Fact 1) dominates. When K is large, Fact 2) balances the effects of Fact 1). Thus, we observe that, as K increases, $C(K)$ decreases and then slightly increases. The figure also shows that, as μ_v/η_a increases, $C(K)$ increases. As μ_v/η_a increases, $E[N]$ increases, and larger $C(K)$ values are observed.

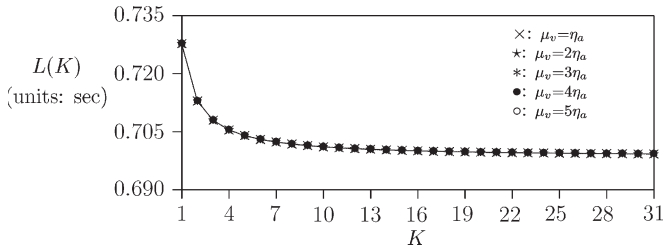

Fig. 8. Effects of μ_v/η_a and K on $L(K)$.

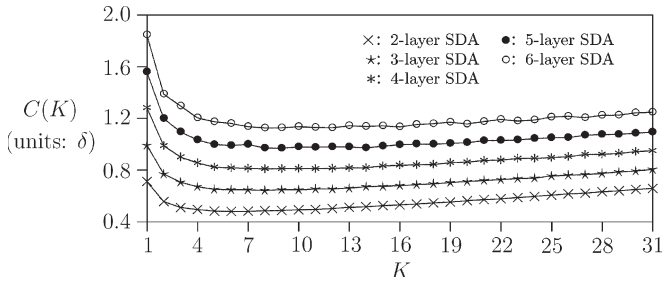

Fig. 9. Effects of SDA size on $C(K)$ ($\mu_v = \eta_a$, $\alpha = 1$, $\beta = 3\delta$, and $\gamma = 12\delta$).

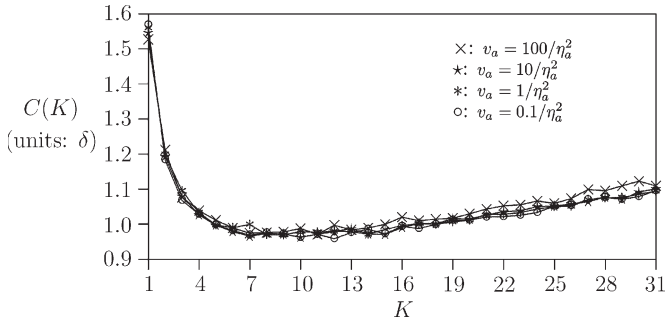

Fig. 10. Effects of v_a on $C(K)$ ($\mu_v = \eta_a$, $\beta = 3\delta$, $\gamma = 12\delta$, and $r = 5$).

Effects of K and ratio μ_v/η_a on $L(K)$. In Fig. 8, we follow the performance metrics in [30] to set $t_{\text{MN–FA}} = 10$ ms, $t_{\text{FA–AAAF}} = 2$ ms, $t_{\text{AAAH–HA}} = 2$ ms, $t_{\text{AAAF–AAAH}} = 30$ ms, and $t_{\text{FA–HA}} = 30$ ms. The figure indicates that μ_v/η_a does not affect the performance of $L(K)$, and $L(K)$ is a decreasing function of K, which is due to the fact that, as the key cache size increases, the MN has better chance of being locally authenticated, and thus, the handoff latency decreases. When $K \geq 16$, this phenomenon becomes insignificant. The performance trend for $L(K)$ is similar to that for $C(K)$.

Effects of SDA size on $C(K)$. Fig. 9 shows that larger $C(K)$ values are observed as r increases. A larger r implies a larger service area of an SDA, and the MN has fewer chances to cross an SDA. More session key sets are required when it stays in a larger SDA. Thus, we have larger $C(K)$ values.

Effects of variance v_a on $C(K)$. Fig. 10 shows $C(K)$ for different settings of v_a. The figure shows that the impact of v_a is insignificant. The performance of $C(K)$ is independent of the distribution of the ADA residence time, which justifies the exponential assumption for ADA residence times in our analytical model.

VI. Automatic K-Selection Algorithm

From the preceding discussion, it is obvious that one may increase the key cache size K to reduce the number of signaling message exchanges. However, the total bandwidth consumption for message exchanges may increase. We observe that the cost of $C(K)$ decreases and then increases as K increases. It is desirable to select an appropriate K value to minimize the transmission cost. In this section, we propose an algorithm called the automatic K-selection algorithm to dynamically adjust cache size K to minimize $C(K)$. The algorithm can easily be implemented in the MN. Each time an MN moves into a new SDA, it calculates a new value of K by referencing the performance of $C(K)$ in the previous SDA.

Let $K(j)$ be the value of K selected for the jth iteration (i.e., when the MN resides in the jth SDA). The automatic K-selection algorithm is described here.

1) *Initialization:* When the MN subscribes to the AAAH, the AAAH assigns an initial value $K(1)$ to the MN (e.g., we set $K(1) = 5$). Then, each time that the MN enters an SDA, two steps are executed.
   a) *Measurement step:* Counters Cr_1, Cr_2, Cr_3, and Cr_4 are implemented in the MN to count the number of occurrences of events 1), 2), 3), and 4) (discussed in Section V), respectively, whereas the MN resides in the jth SDA.
   b) *Decision step:* When the MN leaves the jth SDA, we determine if the value of $K(j+1)$ is to be adjusted to $K_1 = K(j) - 2$ (i.e., $K(j)$ is decremented by two), $K_2 = K(j) - 1$ (i.e., $K(j)$ is decremented by one), $K_3 = K(j)$ (no change), $K_4 = K(j) + 1$ (i.e., $K(j)$ is incremented by one), or $K_5 = K(j) + 2$ (i.e., $K(j)$ is incremented by two). We compute C_i for K_i for $i = 1, 2, 3, 4$, and 5, with the following expression:

$$\begin{aligned} C_i = &\left(\left\lfloor \frac{Cr_1 + Cr_2 + Cr_4}{K_i} \right\rfloor + Cr_3\right) \\ &\times [2\beta + 6\gamma + \delta(10K_i + 6)] \\ &+ \left(Cr_1 + Cr_2 + Cr_4 - \left\lfloor \frac{Cr_1 + Cr_2 + Cr_4}{K_i} \right\rfloor\right) \\ &\times (4\beta + 2\gamma + 6\delta). \end{aligned} \tag{21}$$

The first term in (21) represents the bandwidth consumption for authentication from the AAAH. The second term in (21) represents the bandwidth consumption for local authentication. $\lfloor (Cr_1 + Cr_2 + Cr_4)/K_i \rfloor + Cr_3$ in the first term is the number of session key-set retrievals from the AAAH, and $2\beta + 6\gamma + \delta(10K_i + 6)$ is the bandwidth consumption. $Cr_1 + Cr_2 + Cr_4 - \lfloor (Cr_1 + Cr_2 + Cr_4)/K_i \rfloor$ in the second term estimates the number of local authentications. Based on (21), $K(j+1)$ is selected as follows:

$$K(j+1) = K_l \tag{22}$$

where $1 \leq l \leq 5$, and $C_l = \min_{1 \leq i \leq 5} C_i$.

From the preceding discussion, it seems reasonable that, when the MN enters the $(j+1)$th SDA, the key cache size is

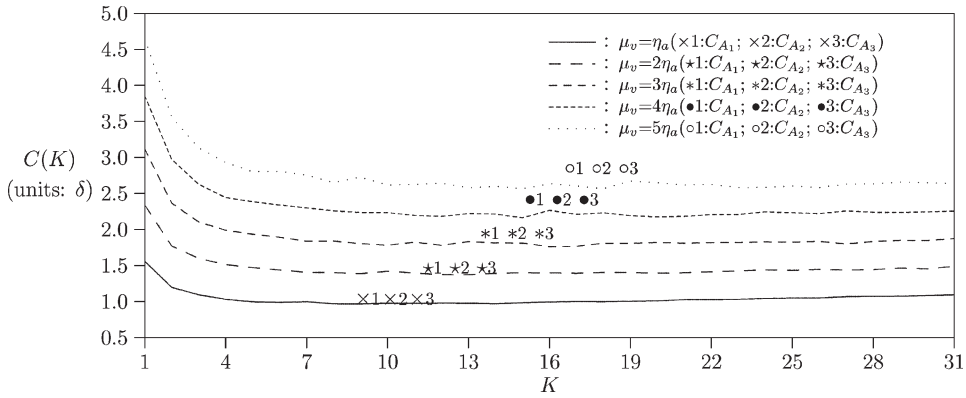

Fig. 11. Performance of the K-selection algorithm ($\alpha = 1$, $\beta = 3\delta$, $\gamma = 12\delta$, and $r = 5$).

TABLE II
$C(K)$ PERFORMANCE OF A_1, A_2, AND A_3

Ratio	C_{A_1}	C_{A_2}	C_{A_3}
$\mu_v = \eta_a$	1.03638	1.03243	1.03373
$\mu_v = 2\eta_a$	1.47343	1.47233	1.47242
$\mu_v = 3\eta_a$	1.95294	1.95091	1.95123
$\mu_v = 4\eta_a$	2.42008	2.41897	2.41976
$\mu_v = 5\eta_a$	2.85493	2.8525	2.85305

assigned the value $K(j+1)$ computed in (22). However, as shown in [29], there may be a small error in (22). Therefore, we attempt to "adjust" the setup of $K(j+1)$ by running simulation tests. As observed in the simulations, this selection is not the best choice. We claim that the best performance is achieved with the following expression:

$$K(j+1) = K_l + 1 \tag{23}$$

where $1 \le l \le 5$, and $C_l = \min_{1 \le i \le 5} C_i$. We will validate this adjustment later via simulation experiments.

We define three key cache size assignments for the $(j+1)$th iteration as follows:

A_1 key cache size is $K(j+1)+1$;
A_2 key cache size is $K(j+1)$;
A_3 key cache size is $K(j+1)-1$.

Fig. 11 shows the performance of A_1, A_2, and A_3 and compares them with AKC (with cache size K), where C_{A_1}, C_{A_2}, and C_{A_3} are the measured costs for A_1, A_2, and A_3, respectively. The figure indicates that all A_1, A_2, and A_3 yield very good performance. Table II lists the values of C_{A_1}, C_{A_2}, and C_{A_3}. The table indicates that selection A_2 yields the best performance of $C(K)$.

VII. CONCLUSION

In this paper, we have proposed an analytical model and conducted simulations to study the performance of the key caching mechanism for the Mobile IP authentication procedure in terms of the expected number $E[N]$ of session key-set retrievals from the AAAH (when the MN resides in the service area of an AAA server), the total bandwidth consumption $C(K)$ for message exchange, and the latency for the MN movement. Our study yields three observations.

1) Increasing cache size K can significantly decrease the expected number $E[N]$ of session key-set retrievals from the AAAH and the averaged handoff latency $L(K)$ when the MN resides in an SDA. When the cache size is large enough (in our study, $K \ge 16$), the improvement becomes insignificant.
2) The bandwidth consumption cost $C(K)$ for AKC with cache size K are concave curves. That is, as K increases, $C(K)$ quickly drops and then slightly increases. There exists an optimal value of K that minimizes $C(K)$.
3) The $C(K)$ performance is independent of the distribution of the ADA residence time.

The study on the AKC mechanism with cache size K suggests that K be adjusted based on the authentication traffic so that the cost of $C(K)$ can be minimized. Then, we propose an automatic K-selection algorithm that dynamically selects the value of K according to the performance of $C(K)$ in the previous SDA. Our study shows that the automatic K-selection algorithm effectively identifies an appropriate value of K to reduce the bandwidth consumption.

It is noted that our analytical model for key caching can also be extended to some other networks, such as the IEEE 802.11r [31], where a context transfer scheme is required to deliver security information between access points for fast handoff. Due to space limitations, we will not cover this part, i.e., the effects of key caching on the IEEE 802.11r, in this paper, which will be treated as future work.

The complete performance analysis for Mobile IP AAA key caching has never been treated in previous works. This study can be considered as the first work that provides an analytical aspect for this problem. Furthermore, based on this analysis, our work can intelligently select the key cache size, which is also a contribution of this work.

APPENDIX

Let $f_A^*(s) = \sum_{m=1}^{\infty} \sum_{y=0}^{r-1} \sum_{j=0}^{r-2} q_{(r-1,y)} p_{m,(r-1,y),(r,j)} [\eta_a/(\eta_a+s)]^m$. We prove that the following hypothesis is true:

$$\frac{d^n f_A^*(s)}{ds^n} = \sum_{m=1}^{\infty} \sum_{y=0}^{r-1} \sum_{j=0}^{r-2} q_{(r-1,y)} p_{m,(r-1,y),(r,j)} \times \left[\frac{(-1)^n \eta_a^m (m+n-1)!}{(\eta_a+s)^{m+n}(m-1)!} \right].$$

Proof: We prove it by induction on n.

Basic: Consider the case when $n = 1$, i.e.,

$$\frac{df_A^*(s)}{ds} = \sum_{m=1}^{\infty}\sum_{y=0}^{r-1}\sum_{j=0}^{r-2} q_{(r-1,y)}p_{m,(r-1,y),(r,j)}\left[\frac{-m\eta_a^m}{(\eta_a+s)^{m+1}}\right].$$

The hypothesis holds.

Inductive Step: Suppose that the hypothesis holds when $n = k$. For $n = k + 1$, we have

$$\begin{aligned}\frac{d^{k+1}f_A^*(s)}{ds^{k+1}} &= d\left[\frac{d^k f_A^*(s)}{ds^k}\right]/ds \\ &= d\Bigg\{\sum_{m=1}^{\infty}\sum_{y=0}^{r-1}\sum_{j=0}^{r-2} q_{(r-1,y)}p_{m,(r-1,y),(r,j)} \\ &\qquad \times\left[\frac{(-1)^k\eta_a^m(m+k-1)!}{(\eta_a+s)^{m+k}(m-1)!}\right]\Bigg\}/ds \\ &= \sum_{m=1}^{\infty}\sum_{y=0}^{r-1}\sum_{j=0}^{r-2} q_{(r-1,y)}p_{m,(r-1,y),(r,j)} \\ &\quad \times\left\{\frac{(-1)^k\left[-(m+k)\right]\eta_a^m(m+k-1)!}{(\eta_a+s)^{m+k+1}(m-1)!}\right\} \\ &= \sum_{m=1}^{\infty}\sum_{y=0}^{r-1}\sum_{j=0}^{r-2} q_{(r-1,y)}p_{m,(r-1,y),(r,j)} \\ &\quad \times\left[\frac{(-1)^{k+1}\eta_a^m(m+k)!}{(\eta_a+s)^{m+k+1}(m-1)!}\right].\end{aligned}$$

Thus, the hypothesis holds in all cases. ■

REFERENCES

[1] C. E. Perkins, "IP Mobility Support for IPv4," *RFC 3344*, Aug. 2002.

[2] S. Glass, T. Hiller, S. Jacobs, and C. E. Perkins, "Mobile IP Authentication, Authorization, and Accounting Requirements," *RFC 2977*, Oct. 2000.

[3] J. D. Solomon, *Mobile IP the Internet Unplugged*. Englewood Cliffs, NJ: Prentice–Hall, 1998.

[4] C. Rigney, S. Willens, A. Rubens, and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)," *RFC 2977*, Jun. 2000.

[5] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, and J. Arkko, "Diameter Base Protocol," *RFC 3588*, Sep. 2003.

[6] P. Calhoun, T. Johansson, C. E. Perkins, E. T. Hiller, and P. McCann, "Diameter Mobile IPv4 Application," *RFC 4004*, Aug. 2005.

[7] M. Cappiello, A. Floris, and L. Veltri, "Mobility amongst heterogeneous networks with AAA support," in *Proc. IEEE Int. Control Conf.*, May 2002, vol. 4, pp. 2064–2069.

[8] L. Salgarelli, M. Buddhikot, J. Garay, S. Patel, and S. Miller, "Efficient authentication and key distribution in wireless IP networks," *IEEE Wireless Commun. Mag.*, vol. 10, no. 6, pp. 52–61, Dec. 2003.

[9] H. Kim and H. Afifi, "Improving mobile authentication with new AAA protocols," in *Proc. IEEE Int. Control Conf.*, May 2003, pp. 497–501.

[10] Z. Zhen and S. Sampalli, "AAA architecture for mobile IP in overlay networks," in *Proc. IEEE LCN*, Nov. 2005, pp. 771–774.

[11] M. Long, C.-H. Wu, and J. D. Irwin, "Localized authentication for wireless LAN Internet working roaming," in *Proc. IEEE WCNC*, Mar. 2004, pp. 264–267.

[12] C. Politis, K. A. Chew, N. Akhtar, M. Georgiades, R. Tafazolli, and T. Dagiuklas, "Hybrid multilayer mobility management with AAA context transfer capabilities for all-IP networks," *IEEE Wireless Commun. Mag.*, vol. 11, no. 4, pp. 76–88, Aug. 2004.

[13] W. Liang and W. Wang, "A local authentication control scheme based on AAA architecture in wireless networks," in *Proc. IEEE Veh. Technol. Conf.—Fall*, Sep. 2004, pp. 5276–5280.

[14] I. Kim and H. Kim, "Local authentication mechanism for micro mobility in wireless active network environment," in *Proc. ICACT*, Feb. 2006, pp. 1135–1139.

[15] P. Jayaraman, R. Lopez, Y. Ohba, M. Parthasarathy, and A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA) framework," *RFC 5193*, May 2008.

[16] E. Fogelstroem, A. Jonsson, and C. E. Perkins, "Mobile IPv4 Regional Registration," *RFC 4857*, Jun. 2007.

[17] H. Soliman, C. Castelluccia, K. E. Malki, and L. Bellier, "Hierarchical Mobile IPv6 Mobility Management (HMIPv6)," *RFC 4140*, Aug. 2005.

[18] C. E. Perkins and P. Calhoun, "Authentication, Authorization, and Accounting (AAA) Registration Keys for Mobile IPv4," *RFC 3957*, Mar. 2005.

[19] L. Kleinrock, *Queueing systems*, vol. 1. New York: Wiley-Interscience, 1975.

[20] R. V. Hogg and E. A. Tanis, *Probability and Statistical Inference*, 6th ed. Englewood Cliffs, NJ: Prentice–Hall.

[21] I. F. Akyildiz, Y.-B. Lin, W.-R. Lai, and R.-J. Chen, "A new random walk model for PCS networks," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 7, pp. 1254–1260, Jul. 2000.

[22] Y.-B. Lin, W.-R. Lai, and R.-J. Chen, "Performance analysis for dual band PCS networks," *IEEE Trans. Comput.*, vol. 49, no. 2, pp. 148–159, Feb. 2000.

[23] S.-R. Yang and Y.-B. Lin, "Performance evaluation of location management in UMTS," *IEEE Trans. Veh. Technol.*, vol. 52, no. 6, pp. 1603–1615, Nov. 2003.

[24] Y.-B. Lin and S.-R. Yang, "A mobility management strategy for GPRS," *IEEE Trans. Wireless Commun.*, vol. 2, no. 6, pp. 1178–1188, Nov. 2003.

[25] I. F. Akyildiz, S. M. Ho, and Y.-B. Lin, "Movement-based location update and selective paging for PCS networks," *IEEE/ACM Trans. Netw.*, vol. 4, no. 4, pp. 629–638, Aug. 1996.

[26] S. M. Ross, *Stochastic Processes*. Hoboken, NJ: Wiley, 1983.

[27] J. Xie and I. F. Akyildiz, "A novel distributed dynamic location management scheme for minimizing signaling costs in mobile IP," *IEEE Trans. Mobile Comput.*, vol. 1, no. 3, pp. 163–175, Jul.–Sep. 2002.

[28] Y. J. Lee and I. F. Akyildiz, "A new scheme for reducing link and signaling costs in mobile IP," *IEEE Trans. Comput.*, vol. 52, no. 6, pp. 706–712, Jun. 2003.

[29] Y.-B. Lin and Y.-K. Chen, "Reducing authentication signaling traffic in third-generation mobile network," *IEEE Trans. Wireless Commun.*, vol. 2, no. 3, pp. 493–501, May 2003.

[30] T. T. Kwon, M. Gerla, and S. Das, "Mobility management for VoIP service: Mobile IP vs. SIP," *IEEE Wireless Commun. Mag.*, vol. 9, no. 5, pp. 66–75, Oct. 2002.

[31] *IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-specific Requirements—Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Amendment 2: Fast Basic Service Set (BSS)*, IEEE Std. 802.11r-2008, Jul. 2008.

Phone Lin (M'02–SM'06) received the BSCSIE and Ph.D. degrees from National Chiao Tung University, Hsinchu, Taiwan, in 1996 and 2001, respectively.

From August 2001 to July 2004 and from August 2004 to July 2008, he was an Assistant Professor and Associate Professor with the Department of Computer Science and Information Engineering (CSIE), National Taiwan University (NTU), Taipei, Taiwan, respectively. Since August 2008, he has been a Professor with the Department of CSIE and the Graduate Institute of Networking and Multimedia, NTU. He has authored more than 20 international Science Citation Index journal papers (most of which are IEEE TRANSACTIONS and Association for Computing Machinery (ACM) papers). He is a Guest Editor for the *ACM/Springer MONET Special Issue on Wireless Broad Access* and an Associate Editorial Member for the *Wireless Communications and Mobile Computing Journal*. His current research interests include personal communications services, wireless Internet, and performance modeling.

Dr. Lin is an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and a Guest Editor for the IEEE WIRELESS COMMUNICATIONS Special Issue on Mobility and Resource Management. He has been the recipient of many research awards, such as the Best Young Researcher, the Third IEEE ComSoc Asia-Pacific Young Researcher Award in 2007, the Research Award for Young Researchers from the Pan Wen-Yuan Foundation in Taiwan in 2004, the K. T. Li Young Researcher Award from the ACM Taipei Chapter in 2004, the Wu Ta You Memorial Award from the National Science Council in Taiwan in 2005, the Fu Suu-Nien Award from NTU in 2005 for his research achievements, and the 2006 Young Electrical Engineering Award from the Chinese Institute of Electrical Engineering. He was listed in *Who's Who in Science and Engineering (R)* in 2006.

Shin-Ming Cheng (S'05–M'07) received the B.S. and Ph.D. degrees in computer science and information engineering from National Taiwan University (NTU), Taipei, Taiwan, in 2000 and 2007, respectively.

In 2007, he joined the Department of Electrical Engineering, NTU, as a Postdoctoral Researcher. His research interests include network security, performance modeling, and cognitive radio networks.

Wanjiun Liao (M'97–SM'05) received the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, in 1997.

In 1997, she joined the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, where she is currently a Full Professor. Her research interests include wireless networks, multimedia networks, and broadband access networks.

Dr. Liao is currently an Associate Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and was on the Editorial Board of the IEEE TRANSACTIONS ON MULTIMEDIA from 2004 to 2007. She has served as the Technical Program Committee (TPC) chair/cochair of many international conferences, including the Tutorial Cochair of the IEEE INFOCOM 2004, the Technical Program Vice Chair of the IEEE Globecom 2005 Symposium on Autonomous Networks, and the Technical Program Cochair of the IEEE Globecom 2007 General Symposium and will be a TPC Cochair of the IEEE Vehicular Technology Conference in the Spring of 2010. She has received many research awards. Papers she coauthored with her students have received the Best Student Paper Award at the First IEEE International Conferences on Multimedia and Expo (ICME) in 2000, the Best Paper Award at the First International Conferences on Communications, Circuits and Systems in 2002, the Republic of China Distinguished Women Medal in 2000, the ACM Taiwan/Taipei K. T. Li Young Researcher Award in 2003, and the National Science Council Outstanding Research Award in 2006.