Kim SK, Choi JW, Nyang DH *et al.* Smart proactive caching scheme for fast authenticated handoff in wireless LAN. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 22(3): 476~480 May 2007

Smart Proactive Caching Scheme for Fast Authenticated Handoff in Wireless LAN

Sin-Kyu Kim[1], Jae-Woo Choi[2], Dae-Hun Nyang[2], Gene-Beck Hahn[3], and Joo-Seok Song[4]

[1] *Electronics Telecommunications Research Institute, Daejeon, Korea*

[2] *Graduate School of Information Technology and Telecommunications, Inha University, Incheon, Korea*

[3] *Mobile Communication Technology Research Lab, LG Electronics, Anyang, Korea*

[4] *Department of Computer Science, Yonsei University, Seoul, Korea*

E-mail: skkim@etri.re.kr; elvin0@empal.com; nyang@inha.ac.kr; gbhahn@lge.com; jssong@emerald.yonsei.ac.kr

Received December 2, 2005; revised March 27, 2007.

Abstract Handoff in IEEE 802.11 requires the repeated authentication and key exchange procedures, which will make the provision of seamless services in wireless LAN more difficult. To reduce the overhead, the proactive caching schemes have been proposed. However, they require too many control packets delivering the security context information to neighbor access points. Our contribution is made in two-fold: one is a significant decrease in the number of control packets for proactive caching and the other is a superior cache replacement algorithm.

Keywords wireless, local area network (LAN), security, authentication, simulation

1 Introduction

Wide prevalence of WLAN (Wireless LAN) is accompanied with many security problems. Due to the properties, WLAN is more vulnerable to the passive and active attacks than wire-line LAN. To resolve the problems regarding the security of WLAN, many efforts have been taken from IEEE standard task groups. Especially, Task Group i of IEEE 802.11 working group has defined the enhanced security functionalities in medium access control layer[1]. Also, IEEE 802.1x has been published for user authentication and for port based access control[2].

Today, handover between APs (Access Point) is not common in WLAN. However, it is expected to become more and more popular. Task group f of IEEE 802.11 has completed the IAPP (Inter-Access Point Protocol) to reduce the burden of handover by information exchange regarding security and QoS contexts between APs[3]. Using the IAPP, several proactive caching schemes have been proposed to reduce the handoff latency caused from the repeated authentication[4,7]. For proactive caching, it is necessary to send in advance context to adjacent APs. Since the proactive caching schemes multicast the context information to all adjacent APs, lots of IAPP packets are redundantly sent to the neighbor APs even though the neighbors already have the same context. When the number of users increases, the number of redundant IAPP packets will also be greatly increased. Ultimately, this degrades the network throughput, which may result in handoff latency.

In this paper, we propose a smart proactive caching scheme to reduce the redundant IAPP packets for fast authenticated handoff. Also, we propose more efficient cache replacement algorithm than LRU (Least Recently Used). Our proactive caching scheme shows good cache hit ratio while significantly reducing the traffics required to proactively cache the context information.

2 Related Work

Recently, the most widely used authentication protocol is EAP-TLS (Extensible Authentication Protocol-Transport Layer Security) in Wireless LAN[5,6]. As a result of authentication in EAP-TLS, authentication server and a supplicant share the secret called PMK (Pairwise Master Key). The secret is later transferred from the authentication server to the authenticator via secure channel. After an authenticator obtains the secret resulted from EAP-TLS operations, it performs 4-way handshake protocol of IEEE 802.11i with supplicant to make a session key called PTK (Pairwise Transient Key)[1]. However, the pre-authentication requires additional overhead for IEEE 802.1x authentication with RADIUS server.

IEEE 802.11f developed IAPP. The purpose of this work is to define a framework to exchange information between APs, where the information would be useful for QoS and security. Especially, they define layer 2 update frame so that any layer 2 devices can update their forwarding tables with the new location of the supplicants.

When a supplicant that is authenticated at an old AP moves to a new AP, it must do full authentication and key exchange procedures again. If a proof for the authentication at old AP is securely sent to new AP, the supplicant does not need to do the repeated authentication. Also, if a PMK (Primary Master Key) of IEEE 802.11i in old AP is sent to new AP in a secure fashion, the supplicant and new AP can establish a secure channel without the aid of RADIUS server. For this, proactive caching strategy is proposed using NG (Neighbor

Graph) by Mishra *et al.*[4]

NG is a data structure that lists the APs directly connected with a specific AP. In NG Scheme, each AP makes NG and authenticates its neighbor APs. And, when a supplicant is authenticated at an AP, the authentication information is sent to all the neighbor APs.

However, their proposal is suffered from the redundant transmission of authentication context, because APs that are adjacent with old and new AP receive the authentication information in twice.

3 Smart Proactive Caching Scheme

In this section, we suggest a method to efficiently propagate the context information to neighbors, where "efficiently" means that the number of packets required to cache propagation is significantly reduced. Also, a new cache replacement algorithm to enhance the cache hit ratio is proposed.

3.1 Smart Cache Propagation Method

Our method of propagating the context exactly simulates the broadcasting method of NG while eliminating the redundant transmission of the same packets such as Cache-Notify packets. Thus, the cache hit ratio is not degraded in spite of the reduced signaling traffic. Followings are the data structures and functions required in our method.

- *c.context*: security context information for client c.
- $AP_i.cache$: cache storage maintained by AP_i.
- $AP_i.AAT$ (*Authenticated AP Table*): authentication table that maintains the list of neighbor APs that have already authenticated clients associated to AP_i. Table 1 describes the internal structure of AAT.
- *c.AAL* (*Authenticated AP List*): the list of neighbor APs of the AP, which the client *c* has associated to.

Table 1. AAT Format

Field	Contents
AP_i	ID of AP that has AP authentication table (AAT).
Client IDs	IDs of clients
Neighbor IDs	Neighbors of AP_i
Authentication	Authentication state about the neighbors of AP_i
Authenticated AP's ID	ID of APs that has already been authenticated by Cache-Notify.
Old AP	ID of AP that sent Cache-Notify or L2_Update message to AP_i.

- *L2_Update*(AP_{new}, *c*, *p*, $AP_{\text{new-n}}$): the L2_Update message is sent from AP_{new} to $AP_{\text{new-n}}$. That is, $AP_{\text{new-n}}$ updates the client *c* related information in its cache and AAT. The priority value for our enhanced cache replacement algorithm is contained in the reserved field of L2_Update message.
- *Cache_AAT_Update*($AP_{\text{new-n}}$, *c*, *p*): this function performs the following in $AP_{\text{new-n}}$ when it receives L2_Update message from AP_{new}:
  - Update Cache: ignores when there is no information on *c* in the cache. Otherwise, adjust the priority according to *p*.
  - Update AAT: ignores when there is no information on *c* in the table. "Old AP" field of the AAT corresponding to the client *c* is modified to AP_{new}'s ID that has been sent in Cache-Notify message.
- *Context_Notification*(AP_{new}, *c.context*, *p*, *c.AAL*, $AP_{\text{new-n}}$): send Cache-Notify message including the context on *c*, priority *p*, and AAL from AP_{new} to $AP_{\text{new-n}}$. Fig.1 shows the concept of our smart propagation method by illustrating the Cache-Notify messages that are not sent redundantly.
- *Insert_AP_Cache*($AP_{\text{new-n}}$, *c.context*, *p*, *c.AAL*): when $AP_{\text{new-n}}$ receives a Cache-Notify message from AP_{new}, this function performs the following:
  - Update Cache: $AP_{\text{new-n}}$ writes *c.context* into its cache with priority *p*.
  - Update AAT: *c.AAL* is stored into the authentication status field of $AP_{\text{new-n}}.AAT$. "Old AP" field of the AAT corresponding to the client *c* is modified to AP_{new}'s ID that has sent Cache-Notify messages.
- *Delete_Context_Table*(AP_{old}, *c*, $AP_{\text{old-n}}$): to increase the cache hit ratio by invalidating the obsolete cache data of $AP_{\text{old-n}}$, AP_{old} might send Cache-Invalidation messages to $AP_{\text{old-n}}$. We suggest using a Move-Response message by setting client's context to NULL instead of defining a new message.

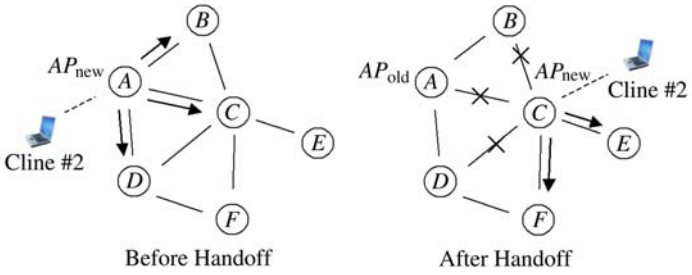

Fig.1. Concept of our smart propagation method.

- *Delete_AP_ID*(AP_{old}, $AP_{\text{old-n}}$): when AP_{old} sends a Move-Response message to $AP_{\text{old-n}}$, AP_{old} removes $AP_{\text{old-n}}$ from its AAT.

Cache-Invalidation message can be used to improve the cache hit ratio by deleting authenticated client's information from $AP_{\text{old}}.cache$ in NG. However, since the authenticated client's information is deleted from all $AP_{\text{old-n}}$ during handoff, AP_{new} sends the client's information to the APs included in $AP_{\text{old-n}}$ and $AP_{\text{new-n}}$ again. To reduce this duplication messages, we suggest an AAT format. In AAT, neighbor IDs field contains

the neighbors of a specific AP. Hence, AP_{new} can find the APs included in $AP_{\text{old-n}}$ and $AP_{\text{new-n}}$. By doing this, we can erase the redundant transmissions of Cache-Notify messages.

When a client in AP_{old} roams to AP_{new}, a number of Cache-Notify messages are made as follows.

$N(AP_i)$: Number of neighbor APs of AP_i
$AP_{\text{new-old-n}}$: Neighbor APs with AP_{new} and AP_{old}
NG Scheme: $N(AP_{\text{new}})$
Our Scheme: $N(AP_{\text{new}}) - N(AP_{\text{new-old-n}})$

Our Scheme can reduce $N(AP_{\text{new-old-n}})$ Cache-Notify messages than NG Scheme.

Table 2 shows the Smart Cache Propagation Algorithm.

Table 2. Smart Cache Propagation Algorithm

AP_{old} = old AP, AP_{new} = new AP, c = client, N = neighbors
1: **if** assoc.req arrives from client c
2: **for** all neighbor AP_i **do**
3: Send L2_Update(c, p) to AP_i
4: Send Cache-Notify($c.context$, p, $c.N$) to AP_i
5:
6: **else if** reassoc.req(AP_{old}) arrives from client c
7: **for** all neighbor AP_i **do**
8: Send L2_Update(c, p) to AP_i
9: **if** AP_i is not included in $AP.AAT$
10: Send Cache-Notify($c.context$, $p, c.N$) to AP_i
11:
12: **else if** Move-Request message arrives from AP_{new}
13: **for** all neighbor AP_i **do**
14: **if** Neighbor AP list(N) of AP_i of AAT does not include AP_{new}
15: $//AP_i$ is not neighbor of AP_{new}
16: Send Move-Response(c) with Cache-Invalidation identifier to AP_i
17: Delete-APID(AP_i)
18:
19: **else if** Cache-notify message arrives
20: Update-Cache(c, p)
21: Update-AAT$(c, c.N)$
22:
23: **else if** L2_Update message arrives from AP_j
24: Update-AAT(c, AP_j)
25: Update-Cache(c, p)
26:
27: **else if** Move-response with Cache-invalidation identifier arrives
28: Delete-Cache(c)

Figs. 2 and 3 show the message delivery process when cache miss and cache hit occurs during handoff.

3.2 Cache Replacement Algorithm Using Weighted LRU

In this section, we suggest a new cache replacement algorithm to increase a cache hit ratio. In this algorithm, the priority is calculated by considering handoff probability.

3.2.1 Weight

Weight w is the inverse of handoff probability[5].

$H(i,j) = \frac{N(i,j)}{R(i,j)}$, $w(i,j) = \frac{1}{H(i,j)}$
$H(i,j)$: Handoff probability (from AP_i to AP_j.)
$N(i,j)$: A number of handoffs (from AP_i to AP_j).
$R(i,j)$: Staying time in AP_i when handoff occurs (from AP_i to AP_j.)

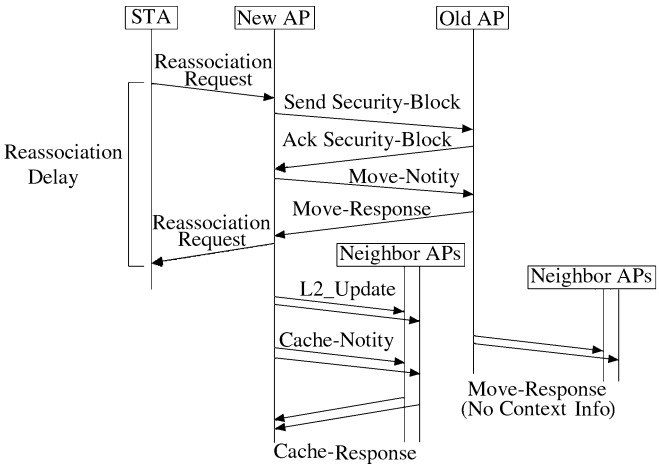

Fig.2. Message delivery process when cache miss occurs during handoff.

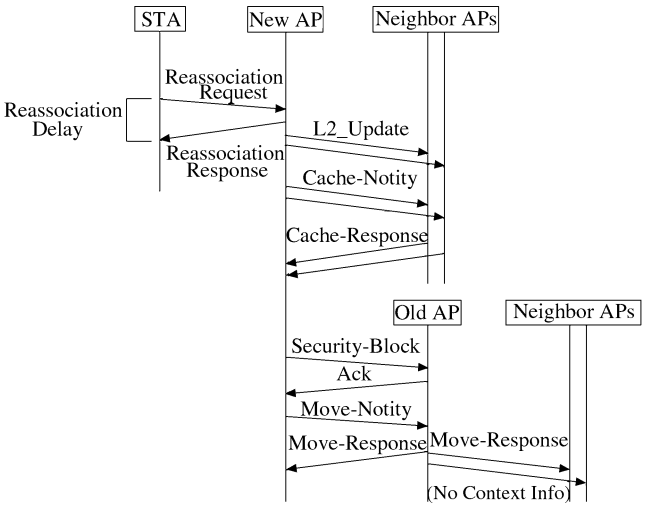

Fig.3. Message delivery process when cache hit occurs during handoff.

3.2.2 Priority

The priority is included in the Cache-Notify or L2_Update. The priority is decided in the weight. Table 3 shows the decision priority $p(i,j)$ function.

Table 3. Decision Priority Function

1: **if** $1 \leqslant w(i,j) \leqslant 3$
2: **return** 0;
3: **else if** $4 \leqslant w(i,j) \leqslant 6$
4: **return** 1;
5: **else if** $7 \leqslant w(i,j) \leqslant 9$
6: **return** 2;
7: **else if** $10 \leqslant w(i,j) \leqslant 12$
8: **return** 3;

4 Performance Evaluation

We evaluate the performance of our scheme through a number of simulations. The simulations are mainly focused on the cache hit ratio and reduction of signaling traffic.

Simulation environment can be presented as follows.

- Discrete simulation using C++.
- AP has a random location and no mobility.
- Node location has a probability of uniform distribution.
- All nodes connect to an AP.
- $1 \leqslant$ initial weight $\leqslant 12$ (probability of uniform distribution).
- Simulation time is 30 000 ticks (one tick has a node mobility as mobility probability).
- The result is presented as a mean of 5 time simulations.

4.1 Simulation Results

4.1.1 Cache Hit Ratio as with the Increasing Number of Clients

Fig.4 compares the cache hit ratio of LRU and WLRU while increasing the number of clients in fifty APs. If the size of cache is sufficient, both ensure the similar cache hit ratio. When the cache size is not sufficient, however, the cache hit ratio of WLRU is higher than that of LRU. The difference of cache hit ratio is nearly the same when the number of clients is either 400 or 500. We can see that the proposed algorithm provides higher hit ratio to the users with high mobility without sacrificing the cache hit ratio of users with low mobility.

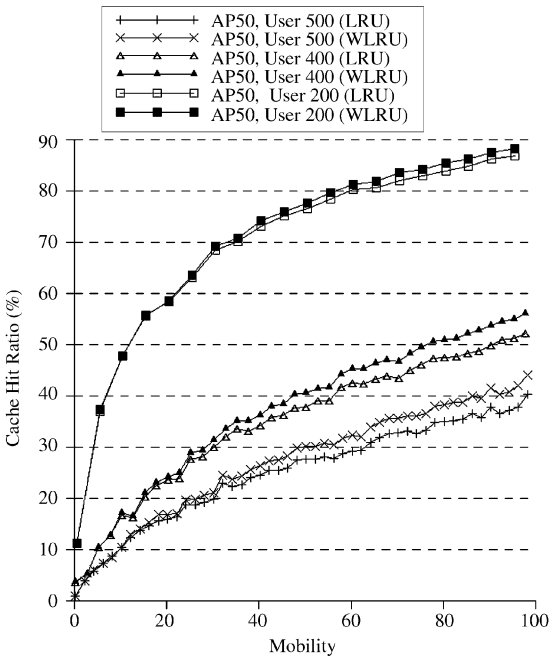

Fig.4. Cache hit ratio with varying mobility.

4.1.2 Total Traffics of NG and Our Scheme under the Same Environments

Fig.5 provides the quantitative comparison for the traffics of both NG and our scheme. Total traffic value means the number of transmissions for cache-notify messages. When the number of clients is 500, our scheme critically works better than NG. Specifically, the traffic gain of our scheme over NG is over 2 000 000 during the simulation period. The reason of this is that our scheme reduces the duplicated message by checking AAT at its AP.

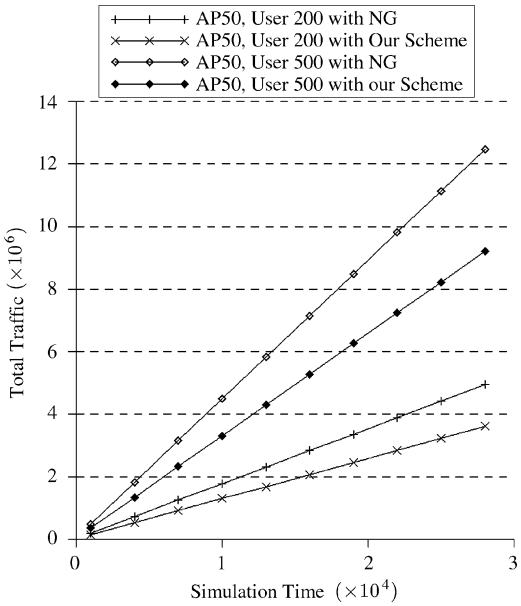

Fig.5. Total traffic with increasing client number.

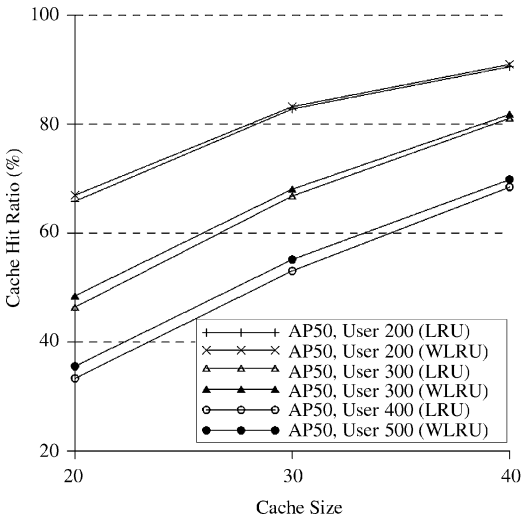

Fig.6. Cache hit ration with varying cache size.

4.1.3 Cache Hit Ratio Regarding the Number of Clients and Cache Size

Fig.6 compares the average cache hit ratio of LRU/WLRU at each cache size while increasing the

number of clients. We can see that the hit ratio of WLRU is higher than that of LRU as with the increasing number of clients when the cache size is not sufficient.

4.1.4 *Traffics as with the Increasing Number of APs and Clients*

Fig.7 compares the traffics of NG and our scheme. As the number of APs increases, the number of overlapped APs will increase. This will affect to the total message traffics. Ultimately, the difference of traffics becomes critical due to the increased traffic volumes as the size of clients increases.

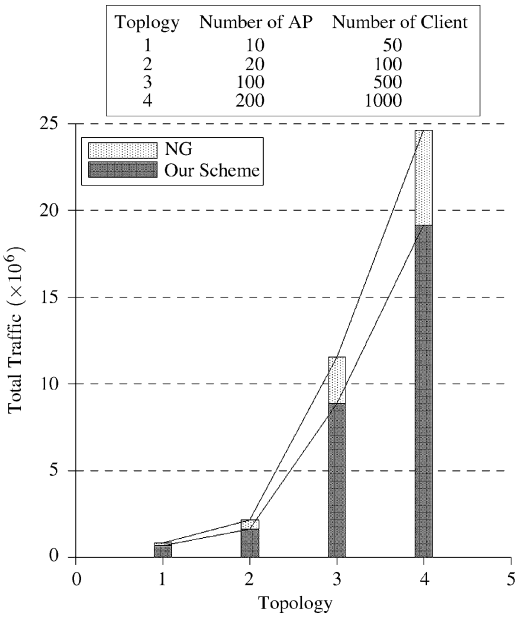

Fig.7. Total traffic in each topology.

5 Conclusion

In this paper, we propose an enhanced traffic management mechanism and cache replace algorithm. In the NG scheme, each AP just authenticates its adjacent APs. However, it incurs the waste of data traffics since the authentication range of old and new APs overlapped. This results from the fact that the context information is sent to all neighbor APs for every handoff.

In our scheme, each AP maintains the AAT. The traffic overloads in entire network can be reduced since the duplicated messages are not sent even when the authentication range of old and new APs is overlapped. Besides the method of reducing the traffics, the WLRU algorithm ensures to increase the cache hit ratio. Thus, with WLRU, we can reduce the handoff latency by growing the hit ratio. And, we expect that the proposed scheme could be used in the WiMAX (IEEE 802.16).

References

[1] Amendment 6: Medium access control security enhancements. IEEE Standard 802.11, IEEE, July 2004.
[2] Port-based network access control. IEEE Standard 802.1x-2001, IEEE, June 2001.
[3] Recommended practice for multi-vendor access point interoperability via an IAPP across distribution systems supporting IEEE 802.11 operation. IEEE Standard 802.11f, IEEE, July 2003.
[4] Mishra A, Shin M, Arbaugh W A. Context caching using neighbor graphs for fast handoffs in a wireless network. In *Proc. IEEE Infocom 2004*, Hong Kong, Mar. 2004, Vol.1, pp.351~361.
[5] Sangheon Pack, Yanghee Choi. Fast inter-AP handoff using predictive-authentication scheme in a public wireless LAN. In *Proc. IEEE Networks 2002*, Atlanta, August 2002, pp.15~26.
[6] PPP EAP TLS authentication protocol. *RFC 2716*, IETF, October 1999.
[7] Mishra A, Shin M, Petroni N *et al.* Proactive key distribution using neighbor graphs. *IEEE Wireless Communications*, Feb. 2004, 11(1): 26~36.
[8] Sangheon Pack, Yanghee Choi. Pre-authenticated fast handoff in a public wireless LAN based on IEEE 802.1x model. In *Proc. IFIP TC6 Personal Wireless Communications*, Singapore, October 2002, Vol. 234, pp.175~182.
[9] Arunesh Mishra, Minho Shin, William Arbaugh. An empirical analysis of the IEEE 802.11 MAC layer handoff process. *ACM Computer Communication Review*, Apr. 2003, 33(2): 93~102.