SNC: A Selective Neighbor Caching Scheme for Fast Handoff in IEEE 802.11 Wireless Networks *

Sangheon Pack[a] *shpack@mmlab.snu.ac.kr*
Hakyung Jung[a] *hkjung@mmlab.snu.ac.kr*
Taekyoung Kwon[a] *tkkwon@snu.ac.kr*
Yanghee Choi[a] *yhchoi@snu.ac.kr*
[a]School of Computer Science & Engineering,
Seoul National University, Seoul, Korea

Mobility support is one of the most challenging issues in IEEE 802.11 networks. In the proactive neighbor caching (PNC) scheme, when a mobile host is connected to an access point (AP), its context (e.g. security association or QoS information) is propagated in advance to all of the AP's neighbors to reduce handoff processing time. In this paper, we propose a selective neighbor caching (SNC) scheme, which propagates a mobile host's context only to the selected neighbor APs considering handoff patterns. Therefore, the SNC scheme can reduce the message overhead on the links among APs. We evaluate the performance of the SNC and PNC schemes in terms of the cache hit probability and the signaling cost. Especially, we investigate the effect of mobility and cache size through extensive simulations. The results reveal that the SNC scheme provides a comparable cache hit probability while significantly reducing the signaling overhead in IEEE 802.11 networks. Moreover, although the SNC propagates relatively a small number of contexts to neighbor APs, the SNC scheme outperforms the PNC scheme when the cache size is small and the mobility is low.

I. Introduction

Public wireless local area network (LAN) systems based on the IEEE 802.11 standard [1] are becoming popular in hot spot areas such as convention centers, airports, campus, etc. Compared to other wireless Internet services, public wireless LAN systems can provide high-speed Internet connectivity of up to 11Mbps (IEEE 802.11b) or 54Mbps (IEEE 802.11a/g). Since the IEEE 802.11 standard was originally designed for an indoor network solution where hosts are stationary, host mobility was not a critical issue. However, recent works [2] [3] [4] [5] indicate that a mobile host (MH) moves from one access point (AP) to another. However, due to the lack of mobility support, there is a significant disruption while performing handoffs. Therefore, it becomes a vital issue to support host mobility in IEEE 802.11 networks for seamless mobile services (especially for real-time applications such as voice over IP (VoIP) services).

Current the IEEE 802.11 standard provides a limited link layer handoff functionality for MHs. The handoff procedure consists of scanning, authentication, and reassociation. As shown in [7], the handoff latency in the current IEEE 802.11 networks is not appropriate to support real-time multimedia applications, which require a handoff latency less than 50 ms [6].

To reduce the handoff latency in IEEE 802.11 wireless networks, a proactive neighbor caching (PNC) scheme was proposed in [8]. The PNC scheme uses a neighbor graph, which dynamically captures the mobility topology of a wireless network for prepositioning an MH's context. The PNC scheme ensures that the MH's context is always dispatched to all of the current AP's neighbors and thereby the handoff latency is reduced. Here, the context contains information regarding the MH's session, quality of service (QoS), and security association [15]. Experimental results show that the PNC scheme reduces the handoff latency from 15.37 ms to 1.69 ms. Currently, the PNC scheme is included in the Inter-Access Point Protocol (IAPP) specification [16], which is a standard protocol for the communications between IEEE 802.11 APs.

However, in the PNC scheme, an MH's context is propagated to all neighbor APs whenever a new (re)association is created. Therefore, the PNC scheme may result in high signaling overhead, especially when there are a lot of MHs (or fast moving MHs) in wireless networks. Furthermore, the previous measurement studies [3] [4] indicate that even in the case when there are a number of neighbor APs, mostly 3 or 4 APs are target points of the handoffs. Therefore,

*The preliminary version of this paper was published in the proceeding of IEEE International Conference on Communications (ICC) 2005, May 2005.

propagating the MH's context to a subset of neighbor APs is likely to be sufficient in most cases for seamless mobility.

In this paper, we propose a *selective neighbor caching (SNC)* scheme, which generalizes the PNC scheme by adding a concept of *neighbor weight*. The neighbor weight represents the handoff probability from a given AP to each neighbor AP. Based on the neighbor weight, the MH's context is propagated only to the selected neighbor APs (i.e. neighbor APs with equal neighbor weights to or higher neighbor weights than a threshold value). The neighbor graph and its neighbor weights can be easily calculated from statistics of the handoff patterns among APs (e.g. [12]).

The rest of this paper is organized as follows. Section II describes the overview of the proactive neighbor caching scheme. In Section III, the selective neighbor caching scheme based on the neighbor weight is proposed. In Section IV, the optimal threshold problem is formulated. Section V shows the simulation results in terms of the cache hit probability and signaling cost. In Section VI, we summarize the previous work for mobility support in IEEE 802.11 networks. Section VII concludes this paper.

II. Proactive Neighbor Caching (PNC) Scheme

Figure 1 illustrates a topology of AP connectivity in the PNC scheme. The AP (i.e. AP_C) located in the center is the current AP that the MH associated with. As shown in Figure 1, the neighbor APs have different handoff probabilities (or weights). These weights are not taken into account in the PNC scheme; however, they are depicted for comparison purposes with the SNC scheme. In the PNC scheme, the MH's context is propagated to all neighbor APs when the MH associates to the AP. In Figure 1, APs within the curve receive the MH's context in a proactive manner.

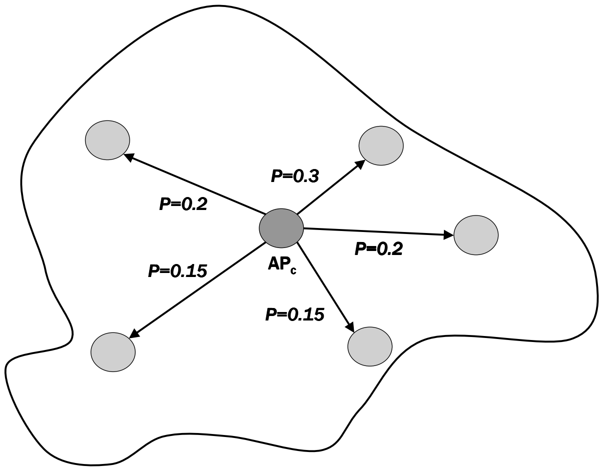


Figure 1: The PNC Operation

The mechanism of the PNC scheme is detailed as follows [8]. First, a neighbor graph is constructed at each AP in a distributed fashion and the MH's context is propagated to all neighbor APs based on the neighbor graph. How to construct the neighbor graph is as follows. First, define a undirected graph $G = (V, E)$ where $V = \{ap_1, ap_2, ..., ap_n\}$ is the set of all APs and there is an edge $e = (ap_i, ap_j)$ between ap_i and ap_j if they have a reassociation (or handoff) relationship. Then, $Neighbor(ap_i) = \{ap_{i_k} : ap_{i_k} \in V, (ap_i, ap_{i_k}) \in E\}$, where ap_{i_k} is the k-th neighbor AP of ap_i. The neighbor graph is automatically generated by the individual AP over time. Two methods can be used to make the edges (or to learn reassociation relationships) in the neighbor graph. Firstly, when an AP receives an IEEE 802.11 reassociation request frame from an MH, the message contains the basic service set identifier (BSSID) of the old-AP and hence establishes the reassociation relationship between the two APs. Secondly, the receipt of a $Move - Notify$ message from another AP via IAPP [16] also establishes the relationship between APs.

Several functions/notations used in the PNC scheme are listed as follows:

1. $Context(c)$: Denotes the context information of a client c.

2. $Cache(ap_k)$: Denotes the cache data structure maintained at ap_k.

3. $Propagate\_Context(ap_i, c, ap_j)$: Denotes the propagation of the client c's context information from ap_i to ap_j. This can be achieved by sending a $Context - Notify$ message from ap_i to ap_j.

4. $Obtain\_Context(ap_{from}, c, ap_{to})$: ap_{to} obtains $Context(c)$ from ap_{from} using an IAPP $Move - Notify$ message.

5. $Remove\_Context(ap_{old}, c, ap_{nghbr})$: ap_{old} sends a $Cache - Invalidate$ message to ap_{nghbr} in order to remove $Context(c)$ from $Cache(ap_{nghbr})$.

6. $Insert\_Cache(ap_j, Context(c))$: Insert $Context(c)$ to the cache data structure at ap_j and perform a least recently used (LRU) replacement if necessary.

The PNC algorithm is presented in Algorithm 1. If an MH associates to an AP, the AP propagates its context to all neighbor APs (lines 2-6). When an MH hands off to a new AP and its context is not in the cache of the new AP, the new AP requests the context to the old AP (lines 8-10). After receiving the context, the new AP propagates the context to its all neighbors (lines 11-13). After transferring context, the old AP and its neighbors remove the MH's context (lines 15-19). In each AP, an MH's context is inserted to the cache and the context may be replaced later on by the LRU policy. More detailed operations of the PNC scheme can be found at [8, 16].

Algorithm 1 PNC Algorithm (ap_j, c, ap_i)

1: ap_j: the current-AP, ap_i: the old-AP, c: the client;
2: **if** client c associates to ap_j **then**
3: **for all** $ap_i \in Neighbor(ap_j)$ **do**
4: $Propagate\_Context(ap_j, c, ap_i)$
5: **end for**
6: **end if**
7: **if** client c reassociates to ap_j from ap_k **then**
8: **if** Context(c) not in Cache(ap_j) **then**
9: $Obtain\_Context(ap_k, c, ap_j)$
10: **end if**
11: **for all** $ap_i \in Neighbor(ap_j)$ **do**
12: $Propagate\_Context(ap_j, c, ap_i)$
13: **end for**
14: **end if**
15: **if** client c reassociates to ap_k from ap_j **then**
16: **for all** $ap_i \in Neighbor(ap_j)$ **do**
17: $Remove\_Context(ap_j, c, ap_i)$
18: **end for**
19: **end if**
20: **if** ap_j received Context(c) from ap_i **then**
21: $Insert\_Cache(ap_j, Context(c))$
22: **end if**

III. Selective Neighbor Caching (SNC) Scheme

Figure 2 illustrates a neighbor graph with the handoff probabilities in the SNC scheme. Similarly to the PNC scheme, an AP in the SNC scheme proactively propagates the MH's context to neighbor APs. However, the AP sends the MH's context only to its neighbor APs whose handoff probabilities are equal to or higher than a pre-defined threshold value. Suppose the threshold value is 0.2. In Figure 2, among all five neighbor APs, three APs within the curve are selected to receive the context. Namely, the other APs whose handoff probabilities are less than 0.2 do not receive the context. Hence, if the MH moves to an AP not receiving the MH's context, the cache miss occurs and results in a longer handoff latency. Overall, by adjusting the threshold value, the SNC scheme can balance the tradeoff between the signaling traffic and the handoff latency. Note that if the threshold value is zero, the SNC scheme becomes equivalent to the PNC scheme.

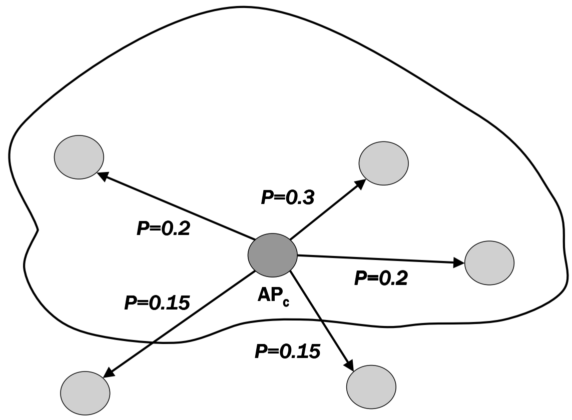


Figure 2: The SNC Operation

In the SNC scheme, an AP assigns the neighbor weight (or handoff probability) to each edge in the neighbor graph. After then, the AP propagates the MH's context only to the selected neighbors with equal to or higher neighbor weights than a predefined value (δ). The neighbor weight can be easily obtained through the construction procedure of the neighbor graph and it is recorded in the *weight table* denoted by $W = [w_i(j)]$, where $w_i(j)$ is the neighbor weight from ap_i to ap_j.

Let $C_i(j)$ be the number of handoff events from ap_i to ap_j during a monitoring interval. Then, $w_i(j)$ is calculated as follows.

$$w_i(j) = \frac{C_i(j)}{\sum_{all \;\; neighbor \;\; k} C_i(k)}$$

Based on the above calculation, when the context propagation is required, the current AP selects neighbor APs meeting the following condition:

$$w_i(j) \geq \delta$$

where δ is the predefined threshold value. After selecting the neighbor APs, the current AP propagates the MH's context to the selected APs. To implement the SNC scheme, a new function, $Update\_Weight(ap_i, ap_j)$ is added. $Update\_Weight(ap_i, ap_j)$ is performed in two cases: (1) when ap_i receives a $Move - Notify$ message from ap_j, and (2) when ap_j receives a reassociation request frame from the MH.

7. $Update\_Weight(ap_i, ap_j)$: Update the weight table by re-calculating the neighbor weight ($w_i(j)$) at ap_i and ap_j.

Algorithm 2 describes the procedure for the SNC scheme. In line 24, the context information of the client c is invalidated after $Remove\_Context(ap_j, c, ap_i)$ whereas the neighbor weight remains in the weight table.

Algorithm 2 SNC Algorithm (ap_j, c, ap_i)

1: ap_j: the current-AP, ap_i: the old-AP, c: the client;
2: **if** ap_i receives a $Move-Notify$ message from ap_j or ap_j receives a reassociation request frame **then**
3:   $Update\_Weight(ap_i, ap_j)$
4: **end if**
5: **if** client c associates to ap_j **then**
6:   **for all** $ap_i \in Neighbor(ap_j)$ **do**
7:     **if** $w_j(i) \geq \delta$ **then**
8:       $Propagate\_Context(ap_j, c, ap_i)$
9:     **end if**
10:   **end for**
11: **end if**
12: **if** client c reassociates to ap_j from ap_k **then**
13:   **if** Context(c) not in Cache(ap_j) **then**
14:     $Obtain\_Context(ap_k, c, ap_j)$
15:   **end if**
16:   **for all** $ap_i \in Neighbor(ap_j)$ **do**
17:     **if** $w_i(j) \geq \delta$ **then**
18:       $Propagate\_Context(ap_j, c, ap_i)$
19:     **end if**
20:   **end for**
21: **end if**
22: **if** client c reassociates to ap_k from ap_j **then**
23:   **for all** $ap_i \in Neighbor(ap_j)$ **do**
24:     $Remove\_Context(ap_j, c, ap_i)$
25:   **end for**
26: **end if**
27: **if** ap_j received Context(c) from ap_i **then**
28:   $Insert\_Cache(ap_j, Context(c))$
29: **end if**

IV. Optimal Threshold

To obtain the best performance in the SNC scheme, the threshold (δ) should be carefully determined. The lower the threshold is, the higher is the cache hit probability. However, as the threshold decreases, the total signaling cost increases. Therefore, it is important to find an optimal weight threshold minimizing the signaling cost while meeting the lower bound of the cache hit probability (P_{th}). To accomplish this, we formulate the optimization problem as follows.

$$\begin{array}{ll} \min & Cost \\ \text{s.t.} & H \geq P_{th} \end{array}$$

where $Cost$ is the signaling cost caused by the context transfer to neighbor APs and H is the cache hit probability. Let P_{ij} and C_{ij} be the handoff probability and context transfer cost from ap_i to ap_j, respectively. To calculate H and $Cost$, we first define a step function as Eq. (1).

$$u(P_{ij}) = \begin{cases} 1, & P_{ij} \geq \delta \\ 0, & P_{ij} < \delta \end{cases} \quad (1)$$

Using the step function, H and $Cost$ are given by Eqs. (2) and (3), respectively.

$$H = \sum_i \sum_j Q_j \cdot P_{ij} \cdot \pi_i \quad (2)$$

$$Cost = \sum_i \sum_j u(P_{ij}) \cdot C_{ij} \cdot \pi_i \quad (3)$$

where π_i is the steady probability at ap_i and Q_j is the probability that an MH's context information is found at ap_j when the MH moves to ap_j. Q_j is calculated depending on P_{ij}.

1. $P_{ij} \geq \delta$: Let t_i be the residence time of an MH at ap_i. Although the MH's context is propagated to ap_j in advance, the cache miss occurs if the number of arrivals of new contexts at ap_j is larger than the cache size (denoted by M). This event is referred to as *cache overflow*. Let P_{re} be the probability that the propagated context is replaced by cache overflow. We assume that the arrival process of new contexts follows a Poisson distribution with rate of λ. Then, P_{re} is given by Eq. (4).

$$P_{re} = \sum_{m=M}^{\infty} \frac{e^{-\lambda t_i}(\lambda t_i)^m}{m!} \quad (4)$$

Then, Q_j is given by Eq. (5) when P_{ij} is equal to or greater than δ. We assume that t_i follows an exponential distribution with rate of μ.

$$\begin{aligned} Q_j &= 1 - \int_0^{\infty} P_{re}(t) \cdot f(t)dt \\ &= 1 - \int_0^{\infty} P_{re}(t) \cdot \mu e^{-\mu t}dt \quad (5) \end{aligned}$$

2. $P_{ij} < \delta$: If P_{ij} is less than δ, no context is propagated to ap_j. Even though ap_j does not receive any context, ap_j may contain the MH's context in two cases: i) the MH visited ap_j previously and the context is not replaced yet or ii) the MH visited ap_k, which is a neighbor AP of ap_j and $P_{kj} \geq \delta$, and the context is not replaced. Figure 3 illustrates these two cases. Let τ_0, τ_1, and τ_2 denote the entering times to ap_k, ap_i, and ap_j, respectively. In addition, T_j is the sum of the residence time at APs before entering to ap_j (i.e. $\tau_2 - \tau_0$). T_j is expressed as follows:

$$T_j = t_k + \sum_{l=0}^{\infty} t_l + t_i$$

where t_k and t_i are the residence times at ap_k and ap_i, respectively.

In this paper, we assume the AP residence time is ideally and identically distributed (i.i.d.). In addition, since the AP residence time is assumed to follow an exponential distribution of rate of μ. T_j also follows an exponential distribution and its service rate (μ_T) is as follows.

$$\mu_T = \frac{\mu}{N} \qquad N \geq 2$$

where N is the number of APs visited during T_j.

Similarly to case 1 (i.e. $P_{ij} \geq \delta$), P_{re} is given by Eq. (6).

$$P_{re}(T_j) = \sum_{m=M}^{\infty} \frac{e^{-\lambda T_j} (\lambda T_j)^m}{m!} \tag{6}$$

As mentioned before, to occur the cache hit in ap_j at τ_2, one of the following two conditions should be met.

- $ap_k = ap_j$: the MH has previously visited ap_j.
- $ap_k \neq ap_j$ and $P_{kj} \geq \delta$: the MH has not visited ap_j, but ap_k is a neighbor AP of ap_j and $P_{kj} \geq \delta$.

Let P_{visit} be the probability for the above two cases. Then, P_{visit} is equal to $\pi_j + \sum_k \pi_k \cdot u(P_{kj})$. Accordingly, Q_j is given by Eq. (7) when P_{ij} (as in Eq. (1)) is less than δ.

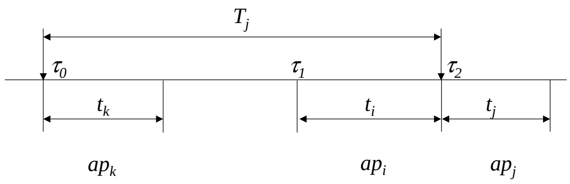


Figure 3: The timing diagram: $P_{ij} < \delta$

$$\begin{aligned} Q_j &= P_{visit} \cdot \left(1 - \int_0^{\infty} P_{re}(t) \cdot f_T(t) dt\right) \\ &= P_{visit} \cdot \left(1 - \int_0^{\infty} P_{re}(T_j) \cdot \mu_T e^{-\mu_T t} dt\right) \end{aligned} \tag{7}$$

where μ_T is the average rate of T_j.

The above optimization problem is to find the maximum δ that meets the condition, $H > P_{th}$. To resolve this problem, we propose a simple binary search algorithm as described in Algorithm 3. L and R denote the left and right boundary values, which are set to 0 and 1, respectively. Flag, F, is used to check whether there are weight thresholds meeting the condition, $H > P_{th}$. In this algorithm, $H(\delta)$ denotes the cache hit probability when the weight threshold value is δ. If $H(\delta)$ is larger than P_{th}, a larger δ value is tested in the next iteration. Otherwise, a smaller δ value is evaluated. This iterations are repeated for N_{iter} times. However, if there are no δ values meeting the constraint, $H > P_{th}$, more iterations ($2 \cdot N_{iter}$) are performed.

V. Performance Evaluation

V.A. Simulation Model

In the case of the SNC scheme, only the selected neighbor APs receive the MH's context. Therefore, unnecessary signaling cost can be reduced. However, if the MH moves to an AP not receiving the context information in advance, the cache miss occurs and the MH should perform a $Obtain\_Context()$ procedure. Accordingly, the average handoff latency increases. To evaluate the performance of the SNC scheme, we have conducted simulations in a reference network consisting of six APs. In our simulations, there are two types of MHs: *fast* and *slow MHs*. The fast MH's AP residence time follows a Gamma distribution which mean is $100 sec$ and variance is $1000 sec^2$. On the other hand, the slow MH's residence time follows a Gamma distribution with mean of $600 sec$ and

Algorithm 3 Determination of optimal weight threshold (δ^*)

```
L ← 0;
R ← 1;
δ ← (L + R)/2;
F ← FALSE;
N ← N_iter;
while N > 0 do
    if H(δ) > P_th then
        L ← δ;
        δ ← (L + R)/2;
        F ← TRUE;
    else
        R ← δ;
        δ ← (L + R)/2;
    end if
    N ← N − 1;
    if F = FALSE and N = 0 then
        N ← N_iter * 2;
    end if
end while
```

variance $6000sec^2$. Total simulation time is $10000sec$ and the number of MHs is 100. In the PNC and SNC schemes, $Remove\_Context()$ procedure is implemented to remove the propagated context maintained at the non-neighbor AP. However, each AP manages its neighbor cache by the LRU replacement policy, so that $Remove\_Context()$ procedure may be redundant. Therefore, we investigate the effect of $Remove\_Context()$ procedure in the next section. We illustrate the handoff probability (or weight) as a matrix form ($\mathbf{P}$) shown in Eq. (8). In Eq. (8), an element represents P_{ij} that is the handoff probability from ap_i to ap_j.

$$\mathbf{P} = \begin{bmatrix} 0 & 0.2 & 0.3 & 0.1 & 0.4 & 0 \\ 0.25 & 0 & 0.15 & 0.2 & 0.1 & 0.3 \\ 0.3 & 0.2 & 0 & 0.1 & 0.3 & 0.1 \\ 0.2 & 0.4 & 0.05 & 0 & 0.15 & 0.2 \\ 0.1 & 0.3 & 0.2 & 0.2 & 0 & 0.2 \\ 0.3 & 0.2 & 0.25 & 0.15 & 0.1 & 0 \end{bmatrix} \quad (8)$$

The steady state probability at each AP can be obtained from Eq. (9) [17].

$$\Pi \cdot \mathbf{P} = \Pi \quad (9)$$

where $\Pi = \{\pi_1, \ \pi_2, \ \pi_3, \ \pi_4, \ \pi_5, \ \pi_6\}$. In the given example, Π is calculated as follows:

$$\Pi = \{0.1856, 0.2034, 0.1630, 0.1318, 0.1771, 0.1391\}$$

In the simulation, we use the above handoff probability to investigate the cache hit probability and signaling cost. The signaling cost is defined as the number of context propagations exchanged between APs. In terms of threshold value (δ), three values (0.1, 0.15, and 0.2) are evaluated. If δ is equal to 0.0, the SNC scheme refers to the PNC scheme.

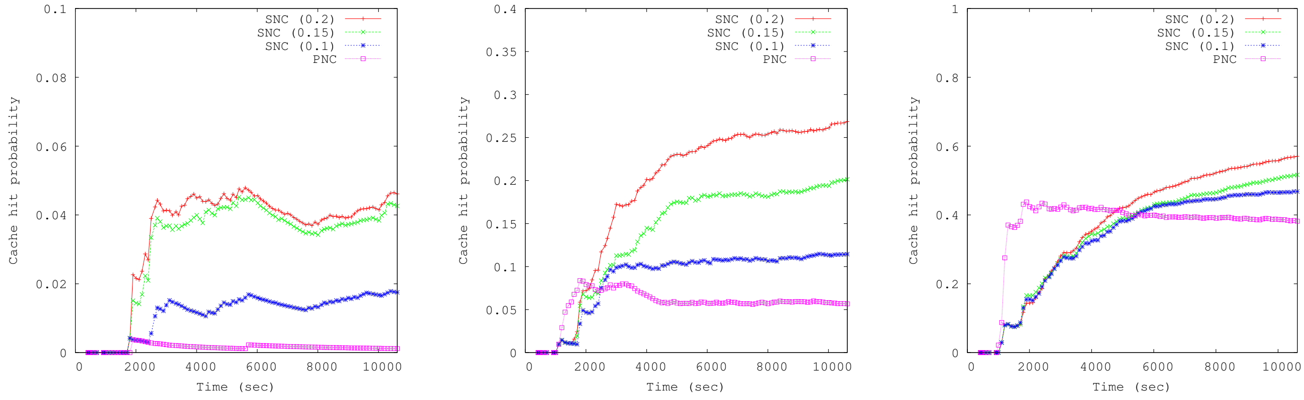
V.B. Cache Hit Probability

Intuitively, the PNC scheme can provide a higher cache hit probability than the SNC scheme. However, as shown in Figure 4, the SNC scheme has a higher cache hit probability than the PNC scheme. This is because the MH's mobility is low and there is no $Remove\_Context()$ procedure. In this case, the PNC scheme, which actively propagates the MH's context to all neighbors, occupies the small-sized cache more aggressively. Furthermore, since $Remove\_Context()$ procedure is not implemented, the effect of context propagations is more notable. In addition, when the MH has a lower mobility, the cache hit probability that the propagated context will be used in near future is too small. In other words, the propagated context will be replaced by other new contexts before it is reused. Consequently, propagating the MH's context to more neighbor APs results in negative effects, i.e. more frequent cache replacements and a lower cache hit probability.

However, as the cache size increases from 30 to 90, the performance of the PNC scheme approaches to that of the SNC scheme. This is because a large number of propagations in the PNC scheme does not degrade the cache hit probability, if the cache size is sufficiently large.

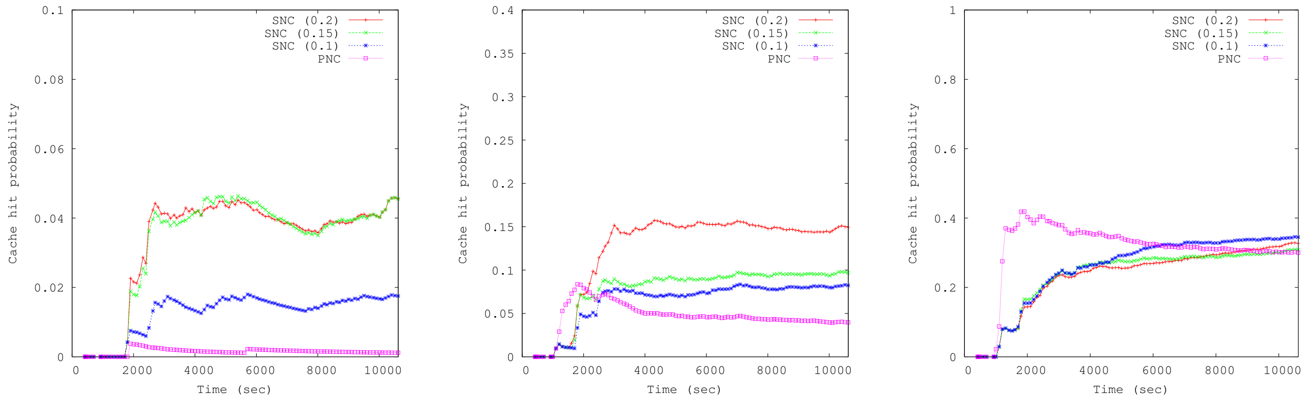
Similar trends are observed in Figure 5. Namely, even though $Remove\_Context()$ procedure is implemented, the performances of the SNC and PNC schemes is highly dependent the MH's mobility. In short, if the MH's mobility is low, the SNC scheme outperforms the PNC scheme. When we compare Figures 5 with 4, the relative gain of the SNC scheme to the PNC scheme is less when $Remove\_Context()$ procedure is implemented. $Remove\_Context()$ procedure prohibits unnecessary occupations of MH's context by eliminating the MH's context at non-neighbor APs. Therefore, the negative effect of the PNC scheme is reduced. In addition, not implementing $Remove\_Context()$ procedure is better if the MH's mobility is low. Namely, in the case of low mobility, maintaining the MH's context as long as possible allows a higher cache hit probability by not removing the previous contexts.

Figure 6 shows the cache hit probability when the MH's mobility is high and no $Remove\_Context()$



(a) Cache size = 30 (low) (b) Cache size = 60 (low) (c) Cache size = 90 (low)

Figure 4: Cache hit probability: no $Remove\_Context()$ procedure and low mobility



(a) Cache size = 30 (low) (b) Cache size = 60 (low) (c) Cache size = 90 (low)

Figure 5: Cache hit probability: $Remove\_Context()$ procedure and low mobility

procedure is implemented. Unlike Figures 4, the PNC scheme shows a higher hit probability even if the cache size is small. In this case, the MH has a higher mobility, the propagated MH's context will be more frequently referenced. Therefore, it is better to propagate the MH's context to as many neighbor APs as possible. However, the difference between the PNC and SNC schemes is not significant. On the other hand, the cache hit probability increases proportional to the increase of the cache size. In addition, overall trends become similar to the result [18] with a infinite cache size. From Figures 4 and 6, we conclude that the effect of mobility is more dominant than the cache size, in terms of the cache hit probability.

Figure 7 plots the cache hit probability as simulation time increases, when the mobility is high and $Remove\_Context()$ procedure is implemented. Overall trends are similar to the results without $Remove\_Context()$ procedure. However, in the case of a higher mobility, the the effect of δ is more apparent in Figure 7 compared with Figure 6. If there is no $Remove\_Context()$ procedure, the MH's context can be replaced only by cache replacement policy. Hence, all contexts are accumulated in the cache, so that the slight difference of δ has little impact on the cache hit probability. On the other hand, if $Remove\_Context()$ procedure is implemented, the propagated MH's context at the non-neighbor AP will be invalidated. Therefore, how many APs receive the MH's context has a direct effect on the cache hit probability.

As shown in Figures 4, 5, 6, and 7, when $Remove\_Context()$ procedure is not implemented, the SNC scheme shows a slightly higher cache hit probability in the case of low mobility. In other words, $Remove\_Context()$ procedure does not guarantee a higher cache hit probability in all cases. Therefore, if the MH's mobility is low, the implicit cache removal (i.e. no $Remove\_Context()$ procedure) is a better choice because the implicit scheme requires a less message overhead, as will be elaborated in the next section. In addition, if the cache size is not sufficiently large, the SNC scheme outperforms the PNC scheme.

V.C. Signaling Cost

As mentioned before, the SNC scheme has an advantage to the PNC scheme in terms of reducing the signaling cost, which is defined the number of contexts transmitted during the simulation time. Tables 1, 2, and 3 summarize the relative signaling cost to the PNC scheme (i.e. the signaling cost of the PNC scheme is normalized to 1.0) when the cache size is 30, 60, and 90, respectively.

As the cache size increases, the relative signaling cost decreases. Since the cache hit probability is proportional to the cache size, the frequency of context transfers is reduced when the cache size is large. However, the effect of cache size is not remarkable.

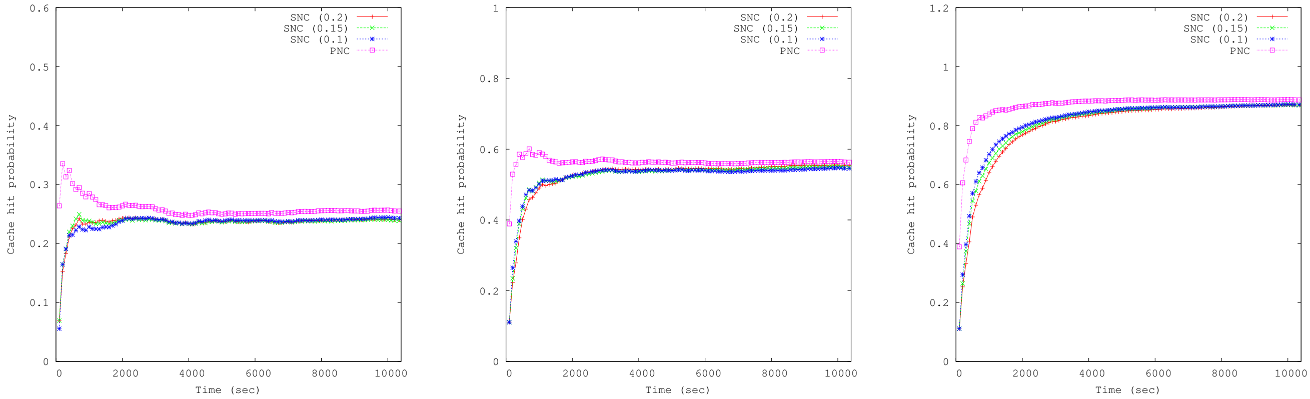
Tables 1, 2, and 3 also show the variation of the relative signaling cost as δ is changed. By increasing δ from 0.1 to 0.2, the relative signaling cost can be lowered by 10%-25%. Especially, when $Remove\_Context()$ procedure in not implemented, the reduction of the relative signaling cost is more significant. In terms of mobility, an MH with high mobility requires more signaling cost because it performs more handoffs than an MH with low mobility.

As shown in Tables 1, 2, and 3, $Remove\_Context()$ procedure results in a higher relative signaling cost by 19-27%. However, as indicated in the previous section, $Remove\_Context()$ procedure does not provides a higher cache hit probability, especially when the cache size is small and the mobility is low. The results reveal that the implicit cache removal policy is better than the explicit cache removal policy, when we consider the signaling cost as well as the cache hit probability. Consequently, the proposed SNC scheme out performs the PNC scheme.

VI. Previous Works

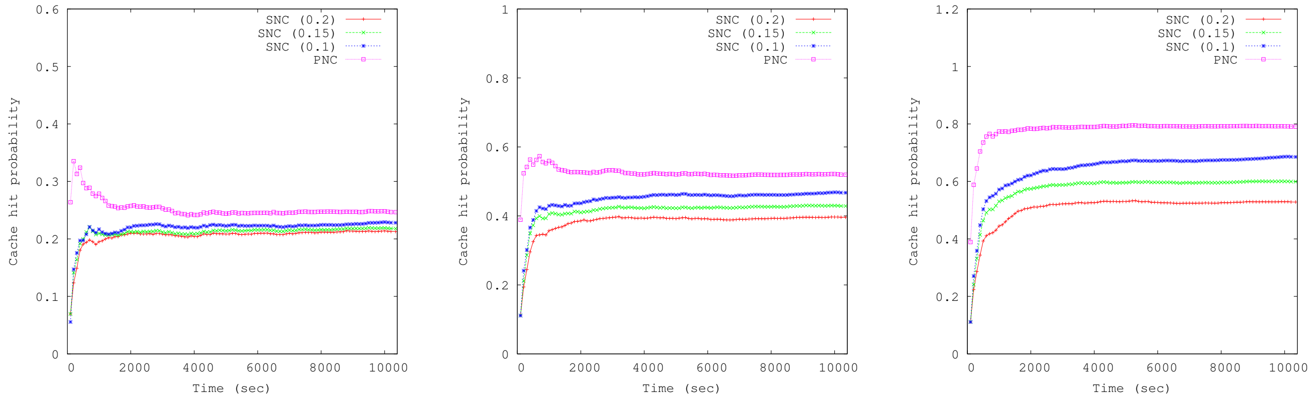
Handoff latency in IEEE 802.11 networks can be divided into three types of latency: scanning latency, authentication latency, and reassociation latency [7]. The PNC scheme have been added to the final specification of IAPP [16]. In addition to the PNC scheme, a few schemes have been proposed to reduce the handoff latency in IEEE 802.11 networks.

As described in [7], the scanning latency is the dominant latency among three types of latency. To reduce the scanning latency, a new scheme was proposed in [9]. [9] described the use of a novel discovery method using a neighbor graph (NG) and non-overlap graph (NOG). This scheme (referred to as *NG-pruning scheme*) focused on reducing the total number of probed channels as well as the total time spent waiting on each channel. They suggested two algorithms: the NG and NG-pruning algorithms. The rationale behind the NG-pruning scheme is to ascertain whether a channel needs to be probed or not (by the NG algorithm) and whether the MH has to wait more time on a specific channel before the expiration of *MaxChannelTime* (by the NG-pruning algo-



(a) Cache size = 30 (high) (b) Cache size = 60 (high) (c) Cache size = 90 (high)

Figure 6: Cache hit probability: no $Remove\_Context()$ procedure and high mobility



(a) Cache size = 30 (high) (b) Cache size = 60 (high) (c) Cache size = 90 (high)

Figure 7: Cache hit probability: $Remove\_Context()$ procedure and high mobility

Table 1: Relative signaling cost: Cache size=30

Case	$\delta = 0.2$	$\delta = 0.15$	$\delta = 0.1$
No $Remove\_Context()$ and high mobility	0.51174	0.61845	0.71962
No $Remove\_Context()$ and low mobility	0.45163	0.52605	0.61458
$Remove\_Context()$ and high mobility	0.70721	0.77089	0.83132
$Remove\_Context()$ and low mobility	0.66061	0.70627	0.76047

Table 2: Relative signaling cost: Cache size=60

Case	$\delta = 0.2$	$\delta = 0.15$	$\delta = 0.1$
No $Remove\_Context()$ and high mobility	0.47136	0.58719	0.69696
No $Remove\_Context()$ and low mobility	0.43090	0.51433	0.60830
$Remove\_Context()$ and high mobility	0.69885	0.76335	0.82515
$Remove\_Context()$ and low mobility	0.65614	0.70444	0.75883

Table 3: Relative signaling cost: Cache size=90

Case	δ = 0.2	δ = 0.15	δ = 0.1
No $Remove\_Context()$ and high mobility	0.42663	0.55132	0.67051
No $Remove\_Context()$ and low mobility	0.40310	0.48930	0.58654
$Remove\_Context()$ and high mobility	0.69271	0.75711	0.81962
$Remove\_Context()$ and low mobility	0.65050	0.79765	0.75259

rithm). The NG abstracts the hand-off relationships between the different APs. Using the NG, the set of channels on which neighboring APs are currently operating and the set of neighbor APs on each channel can be learned. Based on this information, an MH can determine whether a channel needs to be probed or not. On the other hand, the NOG abstracts the non-overlapping relations among the APs. Two APs are considered to be non-overlapping if and only if the MH cannot communicate with both of them with acceptable link quality. So that, if the MH received a probe response frame from AP_i, and AP_i is non-overlapping with AP_j, the MH cannot receive a response message from AP_j (i.e. principle of non-overlapping). By using the NOG, the MH can prune some of the APs that are non-overlapping with the AP groups that have already responded from the waiting group.

In [10], a new handoff scheme using a selective scanning algorithm and caching mechanism was proposed (referred to as the *channel map scheme*). The rationale of selective scanning is to scan only a well-selected subset of all available channels. Channel selection is performed by means of a channel mask that is built when the driver is first loaded at the AP. (i.e. do full-scanning at first, and construct channel masks according to the information so obtained.) In IEEE 802.11b, only three channels do not overlap among all 11 channels (The IEEE 802.11b standard provides for 14 possible channels, but only the first 11 channels of them are used in the US.). Hence, in a well configured wireless network, all or most of the APs operate on channels 1, 6, and 11. Consequently, the channel map is formed using the formula "channels scanned at first full-scanning + 1 + 6 + 11 - the current channel". By using this channel mask, an MH can reduce the amount of unnecessary time that it spends probing non-existent channels among neighboring APs. To further reduce the handoff delay, the cache mechanism was also introduced. The basic idea of the caching mechanism is to store the handoff history of each MH. When an MH associates with an AP, the AP is inserted into the cache. When a handoff is needed, the MH first checks the entries in the cache corresponding to the MAC address of the current AP. If there is a corresponding cache entry, the MH can associated with the AP without the need for any further probing procedure.

In [11], *SyncScan* which is a fast handoff scheme for continuously tracking nearby access points by synchronizing short listening periods at the client with periodic transmissions from each access point. Namely, SyncScan replaces the large transient overhead of active access point discovery with a continuous process that passively monitors other channels for the presence of nearby access points. The potential disruption of channel switching is minimized by synchronizing short listening periods at the client with regular periodic transmissions from each access point. This approach does not require any modification to the 802.11 protocol itself and it can be incrementally deployable. However, the proposed continuous scanning scheme is not free from the synchronization problem. In addition, its performance was evaluated only in the limited network environment.

In [13], the authors split the handoff process into three phases: detection, search, and execution. To reduce the detection time, an MH starts the search phase as soon as collision can be excluded as a reason for failure. Namely, based on the probability distribution, if a frame and its two consecutive retransmissions fail, the MH can conclude that the frame failure is caused by the MH's movement (i.e. further handoff process (search phase) is required) not collision. In terms of search time, a selective active scanning scheme, which scans a smaller list of configured channels not all available channels, was mentioned. At last, the pre-authentication scheme was described in order to reduce the authentication time in the execution phase.

In [13], Velayos at al. proposed a fast handoff detection scheme. In this work, they used the frame loss distribution caused by collisions, in order to determine the optimal handoff timing to a new AP. To reduce the handoff detection time, the MH starts the channel probe procedure as soon as it judges that collision can be excluded as a reason for frame failure. Namely, based on the probability distribution, if a frame and its two consecutive retransmissions fail, the MH can conclude that the frame failure is caused by the MH's movement (i.e. a further handoff process is required) and not by collision. In addition, they derived new values for *MaxChannelTime* and *MinChannelTime* from their measurement results and analytical models. Specifically, they used the smaller values of 1 msec and 10.24 msec for *MaxChannelTime* and *MinChannelTime*, respectively. As a result of these reduced timer values, the active scanning operation provides a less channel probe delay.

A predictive handoff scheme reducing the authentication and reassociation latencies was proposed in [12]. In this scheme, multiple APs are proacitvely authenticated depending on the MH's mobility pattern and service class. To predict the mobility pattern, the concept of the frequent handoff region (FHR) was introduced. The FHR is a set of APs which have a high possibility of being visited by an MH in the near future. The FHR is constructed based on the handoff frequency and the MH's priority at the centralized system. The FHR can readily be implemented based on the IEEE 802.1x model [14]. From several measurement studies [4], the number of APs associated with an MH during its service time is bounded to 2 or 3 APs. Therefore, the FHR-based scheme preauthenticates a part of the adjacent APs which are located at a maximum two hop distance from the current AP. Accordingly, the FHR-based scheme may not provide a lower handoff delay in a highly mobile environment.

VII. Conclusion

It is an important issue to reduce handoff latency and support seamless mobility in IEEE 802.11 networks. In this paper, we proposed the selective neighbor caching (SNC) scheme, which balances trade-off between the handoff delay and the signaling cost. The SNC scheme propagates a mobile host's context only to the selected neighbor access points (APs) considering handoff patterns. When the context transfer is performed, the neighbor APs with equal handoff probabilities to or higher handoff probabilities than a predefined threshold value are selected. We investigate the effect of mobility, cache size on the cache hit probability and the signaling cost through comprehensive simulations. The results reveal that the SNC scheme provides a comparable cache hit probability while significantly reducing the signaling overhead in IEEE 802.11 networks. In our future works, we will evaluate the performance of the SNC scheme using real WLAN traffic traces.

Acknowledgment

This work was supported in part by the Brain Korea 21 project of the Ministry of Education and in part by the National Research Laboratory project of the Ministry of Science and Technology, 2005, Korea.

References

[1] IEEE 802.11b WG, Part 11, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specification: High-speed Physical Layer Extension in the 2.4 GHz Band," *IEEE*, September 1999.

[2] D. Tang and M. Baker, "Analysis of a Metropolitan-area Wireless Network," in *Proc. ACM MOBICOM*, Sept. 1999.

[3] A. Balachandran, G. Woelker, P. Bahl, and P. Rangan, "Characterizing User Behaviour and Network Performance in a Public Wireless LAN," in *Proc. ACM SIGMETRIC*, June 2002.

[4] D. Schwab and R. Bunt , "Characterising the Use of a Campus Wireless Network," in *Proc. IEEE INFOCOM*, March 2004.

[5] F. Chinchilla, M. Lindsey, M. Papadopouli, "Analysis of wireless information locality and association patterns in a campus," in *Proc. IEEE INFOCOM*, March 2004.

[6] International Telecommunication Union, "General Characteristics of International Telephone Connections and International Telephone Circuits," ITU-TG.114, 1998.

[7] A. Mishra, M. Shin, W. Arbaugh, "An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process," *ACM SIGCOMM Computer Communications Review*, Vol. 33, No. 2, April 2003.

[8] A. Mishra, M. Shin, W. Arbaugh, "Context Caching using Neighbor Graphs for Fast Handoffs in a Wireless Network," in *Proc. IEEE INFOCOM*, March 2004.

[9] M. Shin, A. Mishra, and W. Arbaugh, "Improving the Latency of 802.11 Hand-offs using Neighbor Graphs," in *Proc. ACM Mobisys 2004*, Sep. 2004.

[10] S. Shin, A. Forte, A. Rawat, and H. Schulzrinne, "Reducing MAC Layer Handoff Latency in IEEE 802.11 Wireless LANs," in *Proc. ACM MobiWac 2004*, October 2004.

[11] I. Ramani and S. Savage, "SyncScan: Practical Fast Handoff for 802.11 Infrastructure Networks," in *Proc. IEEE INFOCOM 2005*, March 2005.

[12] S. Pack and Y. Choi, "Fast Handoff Scheme based on Mobility Prediction in Public Wireless LAN Systems," *IEE Proceedings Communications*, vol. 151, no. 05, pp. 489-495, October 2004.

[13] H. Velayos and G. Karlsson, "Techniques to Reduce IEEE 802.11b MAC Layer Handover Time," in *Proc. IEEE ICC*, June 2004.

[14] "IEEE Standards for Local and Metropolitan Area Networks: Port based Network Access Control," IEEE Std 802.1x-2001, June 2001.

[15] M. Nakhjiri, C. Perkins, and R. Koodli, "Context Transfer Protocol," IETF Internet Draft, draft-ietf-seamoby-ctp-01.txt, March 2003.

[16] IEEE, "Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation," IEEE Draft 802.1f/Final Version, January 2003.

[17] L. Kleinrock, "Queuing Systems Volume 1: Theory," John Wiley & Sons, 1975.

[18] S. Pack, H. Jung, T. Kwon, and Y. Choi, "A Selective Neighbor Caching Scheme for Fast Handoff in IEEE 802.11 Networks," in *Proc. IEEE ICC*, May 2005.

Biography

Sangheon Pack. He received B.S. (2000, magna cum laude) and Ph.D. (2005) degrees from Seoul National University, both in computer engineering. From July 2005, he is a post-doctor fellow of the Broadband Communication Laboratory (BBCR) in University of Waterloo, Canada. He is a member of the IEEE and ACM. During 2002-2005, he was a recipient of the Korea Foundation for Advanced Studies (KFAS) Computer Science and Information Technology Scholarship. He has been also a member of Samsung Frontier Membership (SFM) from 1999. He received a student travel grant award for the IFIP Personal Wireless Conference (PWC) 2003. He was a visiting researcher at Fraunhofer FOKUS, German in 2003. His research interests include mobility management, wireless multimedia transmission, and QoS provision issues in the next-generation wireless/mobile networks.

Hakyoung Jung. He received a B.S. degree in the School of Computer Science and Engineering at the Seoul National University, Korea, in 2005. He is now a master course student in the Multimedia and Communications Lab. at the Seoul National University. His research interests are mobility management, game theory models, and wireless sensor networks

Taekyoung Kwon. He is an assistant professor in Multimedia & Mobile Communications Lab., School of Computer Science and Engineering, Seoul National University He received his Ph.D., M.S., and B.S. degrees in computer engineering from Seoul National University in 2000, 1995, and 1993, respectively. He was a visiting student at IBM T. J. Watson Research Center in 1998 and a visiting scholar at the University of North Texas in 1999. His recent research areas include radio resource management, wireless technology convergence, mobility management, and sensor network.

Yanghee Choi He received B.S. in electronics engineering from Seoul National University, M.S. in electrical engineering from Korea advanced Institute of Science, and Doctor of Engineering in Computer Science from Ecole Nationale Superieure des Telecommunications (ENST) in Paris, in 1975, 1977 and 1984 respectively. Before joining the School of Computer Engineering, Seoul National University in 1991, he has been with Electronics and Telecommunications Research Institute (ETRI) during 1977-1991, where he served as director of Data Communication Section, and Protocol Engineering Center. He was research student at Centre National d'Etude des Telecommunications (CNET), Issy-les-Moulineaux, during 1981-1984. He was also Visiting Scientist to IBM T. J. Watson Research Center for the year 1988-1989. He is now leading the Multimedia Communications Laboratory in Seoul National University. He was editor-in-chief of Korea Information Science Society journals. He was chairman of the Special Interest Group on Information Networking. He has been associate dean of research affairs at Seoul National University. He was president of Open Systems and Internet Association of Korea. His research interest lies in the field of multimedia systems and high-speed networking.