

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

Sangheon Pack, Hakyung Jung, Taekyoung Kwon, and Yanghee Choi
School of Computer Science and Engineering
Seoul National University, Seoul, Korea
Email: {shpack, hkjung}@mmlab.snu.ac.kr and {tkkwon, yhchoi}@snu.ac.kr

Abstract—Mobility support in IEEE 802.11 networks is a challenging issue. Recently, a new scheme called proactive neighbor caching (PNC) was proposed and adopted as an IEEE standard. The PNC scheme introduces a neighbor graph, which dynamically captures the mobility topology of a wireless network for pre-positioning a mobile host (MH)’s context. However, the PNC scheme may result in significant signaling overhead because the MH’s context is propagated to all neighbor access points (APs). In this paper, we propose a selective neighbor caching (SNC) scheme, which propagates a MH’s context to only the selected neighbor APs considering handoff frequencies between APs. When the context transfer is needed, neighbor APs with equal handoff probabilities to or higher handoff probabilities than a predefined threshold value are selected. We also derive an optimal threshold value when the target cache hit probability is given. Simulation results reveal that the SNC scheme significantly reduces the signaling overhead while guaranteeing a comparable cache hit probability compared to the PNC scheme.

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. Unlike existing wireless Internet services, public wireless LAN system 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 is 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 lack of mobility support, there is significant disruption while performing handoff. Therefore, it becomes a vital issue to support host mobility in IEEE 802.11 networks for seamless mobile services.

Current the IEEE 802.11 standard provides a limited handoff functionality with MHs. The handoff procedure consists of scanning, authentication, and reassociation [7]. 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 lower 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 pre-positioning a MH’s context. The PNC scheme ensures that the MH’s context is always dispatched one

hop ahead and thereby the handoff latency is reduced. Here, the context [13] contains information regarding the MH’s session, quality of service (QoS), and security. Experimental results show that the PNC scheme reduces the reassociation latency by an order of magnitude from 15.37 ms to 1.69 ms. Currently, the PNC scheme is included in the Inter-Access Point Protocol (IAPP) specification [14], which is a standard protocol for the communications between APs.

However, in the PNC scheme, a 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 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, propagating the MH’s context to a subset of neighbor APs may be sufficient for seamless mobility.

In this paper, we propose an enhanced neighbor caching scheme called *selective neighbor caching (SNC)* scheme. The SNC scheme generalizes the PNC scheme by adding a concept of *neighbor weight*. The neighbor weight represents the handoff probability for 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 by measuring handoff patterns among APs.

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

II. PREVIOUS WORK

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 [14]. In addition to the PNC scheme [8], 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]. The proposed scheme can reduce the total number of scanned channels as well as the total time spent waiting on each channel. Specifically, two algorithms were introduced: NG (neighbor graph) algorithm and NG-pruning algorithm. The NG algorithm uses the neighbor graph whereas the NG-pruning algorithm further improves the channel discovery process by using the non-overlap graph.

In [11], the authors split the handoff process into three phases: detection, search, and execution. To reduce the detection time, a 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.

On the other hand, a predictive handoff scheme reducing the authentication and reassociation latencies was proposed in [10]. To predict the mobility pattern, the frequent handoff region (FHR) was introduced. The FHR is a set of APs having higher possibilities that an MH visits to them in near future. The FHR is constructed based on the handoff frequency and users' priority. The FHR can be simply implemented based on the IEEE 802.1x model [12].

III. PROACTIVE NEIGHBOR CACHING (PNC) SCHEME

Figure 1 illustrates the operation of the PNC scheme. The AP (i.e., AP_C) located in the center is the current AP that the MN associated with. As shown in Figure 1, 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. The handoff probabilities can be calculated using a similar method to [10]. In the PNC scheme, a MH's context is propagated to all neighbor APs when the MH associates to an AP. In Figure 1, APs within the curve represent APs receiving the MH's context in a proactive manner.

The detailed mechanism of the PNC is as follows [8]. First, a neighbor graph is constructed at each AP 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 an undirected graph $G = (V, E)$ where $V = \{ap_1, ap_2, \dots, ap_n\}$ 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 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 can be automatically generated by the individual AP over time. Two ways can be used to make the edges (or learn reassociation relationships)

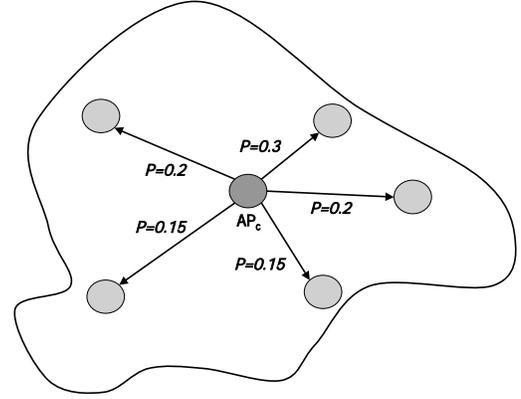


Fig. 1. The PNC Operation

in the neighbor graph. Firstly, when an AP receives an IEEE 802.11 reassociation request frame from a 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 [14] also establishes the relationship between APs.

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

- 1 $Context(c)$: Denotes the context information related to 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 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 a MH associates to an AP, the MH propagates its context to all neighbor APs (lines 2-6). If the cached context does not exist in the new AP, the new AP requests the context to the old AP. After receiving the context, the new AP propagates the context to its all neighbors (lines 7-14). After transferring context, the old-AP and its neighbors removes the MH's context (lines 15-19). In each AP, a MH's context is inserted to the cache and the context may be replaced by the LRU policy.

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
```

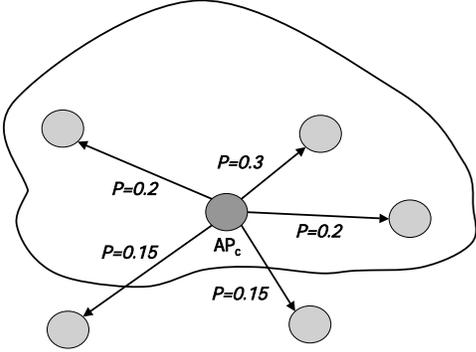


Fig. 2. The SNC Operation

IV. SELECTIVE NEIGHBOR CACHING (SNC) SCHEME

Figure 2 shows the operation of the SNC scheme. Similar 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 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, three APs within the curve are selected as the candidate neighbors and 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.

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.

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

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 candidate 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 at ap_i and ap_j when ap_i receives a *Move - Notify* message from ap_j and ap_j receives a reassociation request frame from the MH, respectively.

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 shows the modified procedure for the SNC scheme. In line 24, the context information of client c is invalidated after $Remove_Context(ap_j, c, ap_i)$ whereas the neighbor weight remains in the weight table.

V. OPTIMAL WEIGHT THRESHOLD

To obtain the best performance in the SNC scheme, the threshold neighbor weight (δ) should be carefully determined. The lower the threshold weight is, the higher the cache hit probability is. However, as the weight 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.

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

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 . To calculate H and $Cost$, we first define a step function as Eq. (1).

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

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  $w_j(j) \geq \delta$  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
```

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 context information was delivered to ap_j in advance. Q_j is calculated as follows:

$$Q_j = \sum_k u(P_{kj}) \cdot P_{kj}$$

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 shown 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.

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

```
1:  $L \leftarrow 0$ ;
2:  $R \leftarrow 1$ ;
3:  $\delta \leftarrow (L + R)/2$ ;
4:  $F \leftarrow \text{FALSE}$ ;
5:  $N \leftarrow N_{iter}$ ;
6: while  $N > 0$  do
7:   if  $H(\delta) > P_{th}$  then
8:      $L \leftarrow \delta$ ;
9:      $\delta \leftarrow (L + R)/2$ ;
10:     $F \leftarrow \text{TRUE}$ ;
11:  else
12:     $R \leftarrow \delta$ ;
13:     $\delta \leftarrow (L + R)/2$ ;
14:  end if
15:   $N \leftarrow N - 1$ ;
16:  if  $F = \text{FALSE}$  and  $N = 0$  then
17:     $N \leftarrow N_{iter} * 2$ ;
18:  end if
19: end while
```

VI. PERFORMANCE EVALUATION

In the case of the SNC scheme, only the selected neighbor APs receives 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. As a result, 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. The AP residence time follows a Gamma distribution whose mean is $600sec$ and variance is $6000sec^2$. Total simulation time is $10000sec$ and the number of MHs is 100. We illustrate the handoff probability (or weight) as a matrix form (\mathbf{P}) shown in Eq. (4). In Eq. (4), an element P_{ij} is the handoff probability from ap_i to ap_j .

$$\mathbf{P} = \begin{bmatrix} 0 & 0.2 & 0.3 & 0.1 & 0.4 & 0 \\ 0.25 & 0 & 0.15 & 0.2 & 0.1 & 0.3 \\ 0.3 & 0.2 & 0 & 0.1 & 0.3 & 0.1 \\ 0.2 & 0.4 & 0.05 & 0 & 0.15 & 0.2 \\ 0.1 & 0.3 & 0.2 & 0.2 & 0 & 0.2 \\ 0.3 & 0.2 & 0.25 & 0.15 & 0.1 & 0 \end{bmatrix} \quad (4)$$

The steady state probability at each AP can be obtained from Eq. (5) [15].

$$\mathbf{\Pi} \cdot \mathbf{P} = \mathbf{\Pi} \quad (5)$$

where $\mathbf{\Pi} = \{\pi_1, \pi_2, \pi_3, \pi_4, \pi_5, \pi_6\}$.

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), the optimal value obtained from Algorithm 3 and several values are evaluated. If δ is equal to 0.0, the SNC scheme is reduced to the PNC scheme.

Figure 3 shows the cache hit probability in the PNC and SNC schemes as time goes by. In the case of the SNC scheme,

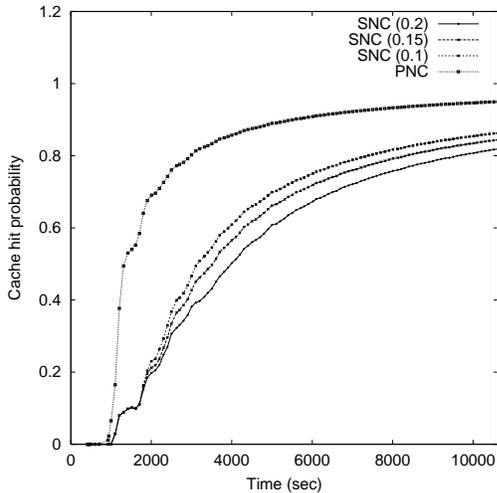


Fig. 3. Cache hit probability: SNC vs. PNC

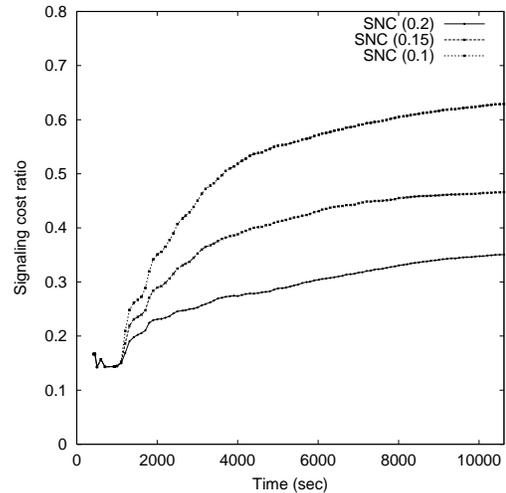


Fig. 4. Signaling cost ratio: SNC to PNC

three δ values are evaluated. At the beginning of simulation, the cache hit probability is too low because there are no MHs' contexts at APs' cache at the initial stage. However, over time, MHs' contexts are propagated to neighbor APs and the cache hit probability remarkably increases. However, the cache hit probability becomes stable after the neighbor graph is sufficiently constructed. The SNC scheme shows a lower hit probability than the PNC scheme, but the difference decreases as simulation time proceeds. The differences among the cache hit probabilities using different δ values are not significant.

Signaling costs in the PNC and SNC schemes are compared in Figure 4. We assume that the context transfer cost for each link has the same value (i.e., C_{ij} is a constant value.). Figure 4 shows the relative signaling cost of the SNC scheme to that of the PNC scheme. Namely, the relative signaling cost of the PNC scheme is always 1. The reduction of signaling cost is important especially when the size of an IEEE 802.11 wireless network becomes bigger and there are a large number of MHs in the network. As shown in Figure 4, the SNC scheme results in a lower signaling cost than the PNC scheme. Specifically, the SNC scheme can reduce the signaling cost by 37%-64%. In contrast to the cache hit probability, the difference of the signaling costs with different δ values are more notable.

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 that balances trade-off between the handoff delay and the signaling cost. Using the handoff probability, the SNC scheme showed a similar cache hit probability to the PNC scheme while reducing the signaling overhead significantly. The performances of the SNC and PNC schemes are expected to be highly dependent on the cache size and cache replacement policy, which will be our future works.

ACKNOWLEDGMENT

This work was supported in part by the BK 21 project of the Ministry of Education and in part by the NRL project of the Ministry of Science and Technology, 2004, 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 et al., "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 et al., "Analysis of wireless information locality and association patterns in a campus," in *Proc. IEEE INFOCOM*, March 2004.
- [6] ITU, "General Characteristics of International Telephone Connections and International Telephone Circuits," ITU-TG.114, 1998.
- [7] A. Mishra et al., "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 et al., "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* September 2004.
- [10] S. Pack and Y. Choi, "Fast Handoff Scheme based on Mobility Prediction in Public Wireless LAN Systems," *IEE Proceedings Communications*, Vol. 151, No. 05, October 2004.
- [11] H. Velayos and G. Karlsson, "Techniques to Reduce IEEE 802.11b MAC Layer Handover Time," in *Proc. IEEE ICC*, June 2004.
- [12] IEEE, "IEEE Standards for Local and Metropolitan Area Networks: Port based Network Access Control," IEEE Std 802.1x-2001, June 2001.
- [13] M. Nakhjiri, C. Perkins, and R. Koodli, "Context Transfer Protocol," IETF Internet Draft, draft-ietf-seamoby-ctp-01.txt, March 2003.
- [14] 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 802.11f/Final Version, January 2003.
- [15] L. Kleinrock, "Queueing Systems Volume 1: Theory," John Wiley & Sons, 1975.