Wireless Data Access in Mobile Hotspots: A Simulation Study

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

***Abstract*— Mobile hotspots enable ubiquitous Internet access while users are on-board a vehicle. In this paper, we investigate wireless data access applications with strong consistency in mobile hotspots. Through extensive simulations, we quantify the performances of two well-known strongly consistent data access algorithms: poll-each-read (PER) and callback (CB). We develop a comprehensive simulation model that considers wireless link characteristics in mobile hotspots. Simulation results demonstrate the access-to-update ratio is an important factor to determine the performances of PER and CB. In addition, the effects of cache size, data access pattern, and wireless link bandwidth on the wireless data access algorithms are analyzed. Finally, valuable observations for performance optimizations are given.**

I. Introduction

Recently, wireless local area network (WLAN)-based hotspot services are being widely deployed in public areas such as convention centers, cafes, airports, and shopping malls. On the other hand, the extension of hotpot services in moving vehicles such as subways, trains, buses, and ships is gaining much attention [1]. The hotspot service in a mobile platform is referred to as *mobile hotspot* [2], which is a novel concept to realize ubiquitous and always best connected (ABC) services in future wireless/mobile networks.

For a successful deployment, various research for mobile hotspots has been conducted: mobility management [3], quality of service (QoS) support [4], link layer transmission technique [2], gateway architecture [5], and testbed implementation [6]. In this paper, we focus on wireless data access applications in mobile hotspots because it is perceived as a killer application for ubiquitous computing. Specifically, we consider wireless data access applications requiring *strong consistency*, which is widely accepted for various applications such as mobile banking, stock trade, and so on.

For strongly consistent wireless data access, two algorithms, *poll each read (PER)* [7] and *callback (CB)* [8], have been proposed. These two algorithms have been analyzed in [9], [10] and extended in [11], [12]. All of these works follow a typical cellular system architecture with a wireless one-hop link. However, data access applications in mobile hotspots are built in a different network topology with a heterogeneous wireless two-hop link. Figure 1 illustrates the network topology for wireless data access in mobile hotspots, more specifically an Internet access application in public transportation systems. A mobile terminal (MT) accesses an application server (AS) in the Internet through an access point (AP), which is installed at the public transportation system and provides connection to the AS in the wired Internet. The first wireless hop from the MT is a wireless local area network (WLAN) link, whereas the second wireless hop is a wireless wide area network (WWAN) link. This WLAN-WWAN integrated link provides several advantages in mobility management and resource management for data access in mobile hotspots [13].

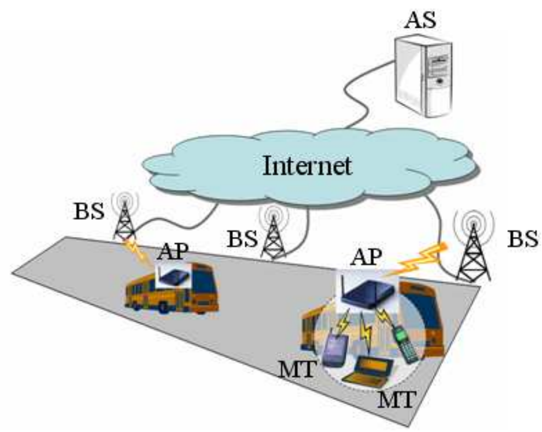

Fig. 1. Internet access in public transportation systems.

Since wireless data access applications in mobile hotspots are based on the different network model, their performances should be thoroughly studied in realistic environments. In this paper, we carry out comprehensive simulation studies to evaluate the performances of PER and CB in mobile hotspots. We quantify the transmission cost in PER and CB, and the effects of cache size, access pattern, data object size, and wireless link bandwidths are extensively analyzed. The main contribution of this paper is to conduct comprehensive and extensive simulations, which exploit heterogeneous wireless link characteristics in mobile hotspots. The observations from the simulations can be utilized to optimize the performance of wireless data access algorithms in mobile hotspots.

The remainder of this paper is organized as follows. The PER and CB algorithms are described in Section II. Simulation environments are presented in Section III and simulation results are given Section IV. Section V concludes this paper with the concluding remarks.

II. Strongly Consistent Data Access Algorithms

In this section, we describe PER and CB algorithms in mobile hotspots, where the access point (AP) and the base station (BS) relay the received messages transparently toward the application server (AS).

A. System Model

In a wireless data access algorithms in mobile hotspots, an MT connects to an AS through the WLAN-WWAN integrated link. Although WLAN provides high bandwidth compared to WWAN, its coverage is relatively short. On the other hand, the service area of WWAN is wide but the utilizable bandwidth is smaller than WLAN. Consequently, WLAN is used to connect MTs within a mobile hotspot, whereas WWAN is used to serve many mobile hotspots in a wide area. In this way, the WLAN-WWAN integrated link can accommodate more users without excessive usage of the WWAN link.

Since we assume the wireless data access application over reliable transport and/or data link protocols, no packet loss is observed at the application layer. Moreover, the time interval between two events that access the same object is relatively larger than the channel delay variations, so that it is reasonable to further consider the WWAN and WLAN links with fixed but arbitrary bandwidths. Consequently, the wireline link and WLAN link is assumed to be an symmetric link, and their bandwidths are considered as B Mbps and L Mbps, respectively. For the WWAN link, which is usually considered as an asymmetric link, uplink and downlink bandwidths are W_{Up} Kbps and W_{Down} Kbps, respectively.

We assume a sever-only write system, where all modifications to objects are accomplished only by the AS. MTs only access the AS for obtaining some data from the AS through the WLAN-WWAN integrated link. To effectively use the wireless link, each MT has a cache and the size of the cache is represented as K_{MT}. In the literature, several cache replacement schemes have been reported, e.g., least frequently used (LFU), least recently used (LRU), and so on. To improve the cache efficiency, a new cache replacement scheme based on the update to access ratio has been introduced in [11]. However, the accurate update rate can be measured only by the AS because object modifications occurs only at the AS. Therefore, we consider an LRU scheme as the MT cache replacement scheme.

Each MT is identified by an identifier j. Let O_i be the ith data object. O_i is associated with a time sequence number t ($t > 0$), which is assigned in an increasing manner. That is, O_i with $t + \Delta$ ($\Delta > 0$) is an up-to-date object than O_i with t. For instance, an data object O_i that is cached in a MT j at time t can be uniquely identified by (i, j) with t. For wireless data access algorithms, the following messages are defined.

- **Access(i, t)**: It requests an access of object i. For PER, $t > 0$ specifies the current sequence number for a cached object; on the other hand, $t = 0$ represents that there is no object in the cache. For CB, since invalidation by the AS is performed, t is not used and it is always set to 0.

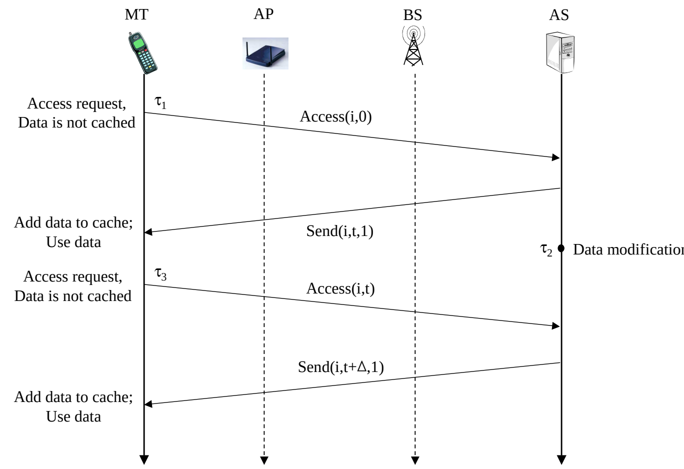

Fig. 2. PER operation.

- **Send(i, t, F)**: This message is used for sending a data object or confirming **Access(i, t)** in PER. i and t denote the object index and the sequence number, respectively. F is a flag indicating whether a data object is included in the message or not. If F is set (i.e., $F = 1$), the object i is transmitted with this message. Otherwise, only a confirmation message is sent.
- **Update(i, j)**: It invalidates the object i in the cache of an MT j.
- **Ack(i, j)**: This message acknowledges the receipt of **Update(i, j)**.

B. Poll-Each-Read (PER) Algorithm

In PER, an MT polls the AS to check whether its cached object is up-to-date one whenever it accesses an object. If the cached object is the latest one, the AS sends a confirmation message without any data object. Otherwise, the AS delivers the requested data object to the MT.

Figure 2 illustrates the operation of PER. Let τ_i be the time instance when an event occurs and $\tau_i < \tau_{i+1}$. At time τ_1, an MT tries to access a data object O_i. Since the O_i is not cached, the MT sends **Access(i, 0)** to the AS. Receiving the message, the AS sends back **Send(i, t, 1)** where O_i is included. Let τ_2 be the modification time of O_i. Then, the cached data object for O_i in the MT becomes invalid. When the MT retries to access O_i at time τ_3, it sends **Access(i, t)** to the AS to validate the cached object. Since O_i is updated at time t_2, the AS returns **Send(i, $t+\Delta$, 1)** with the up-to-date O_i to the MT. Note that $t + \Delta$ is the same as τ_2, where $\Delta > 0$. No caches at the AP and BS are considered, so that every messages are transparently delivered between the MT to the AS.

C. Callback (CB) Algorithm

CB satisfies the requirement of strongly consistency by the invalidation process. When a data object is modified, the AS explicitly notifies to MTs that the data object is modified and they are not valid any more. Receiving the

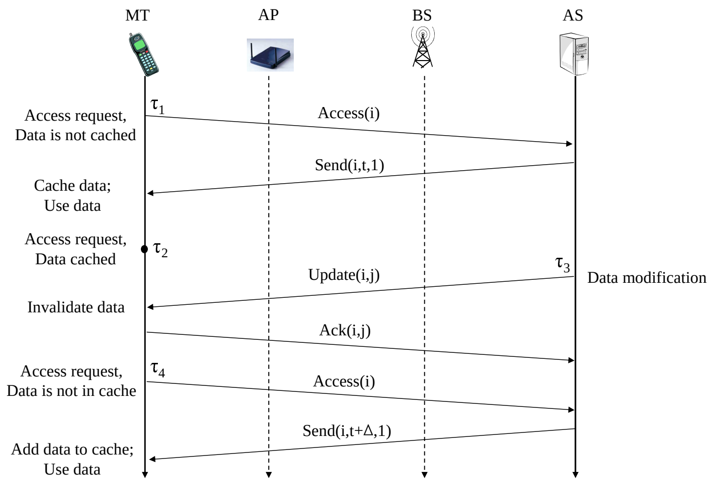

Fig. 3. CB operation.

notification message, the MT discards the object from its cache. Accordingly, if an MT has an object in its cache, it is guaranteed that the object is up-to-date and there is no need to check whether the object is valid or not. On the other hand, if an MT does not have an object in its cache, it should contact the AS in order to receive the latest object.

The CB operation is illustrated in Figure 3. Through the initial access at time τ_1, the MT gets the data object O_i. At time τ_2, since O_i is still stored in the cache, the MT can use the object without accessing to the AS. When the object is modified at time τ_3, the AS sends **Update(**i**,** j**)** to the MT where j is the identifier of the MT. After the MT receives the **Update** message, it discards O_i from its cache and sends an acknowledgement message **Ack(**i**,** j**)** to the AS. Since the object is removed from the cache, the MT should contact the AS and get the updated O_i if it is needed.

In PER, even though an object is found in the cache, the MT should always confirm that the object is up-to-date one. Therefore, for frequent data accesses, PER can lead to a significant transmission cost. On the other hand, no confirmation procedure is required in CB. Therefore, if there is an object in the cache, the MT can use the cached one without contacting the AS. However, to this end, the AS should invalidate MTs with a cached object whenever a modification to the object occurs, which incurs a high transmission cost when the object update events are frequent. Consequently, an obvious trade-off relationship between PER and CB exists and it is affected by the access and update rates. We will compare the performances of PER and CB under different access and update rates in the later.

III. Simulation Environment

To evaluate the performances of PER and CB, we have developed an event-driven simulator and performed comprehensive simulations. N data objects are in the AS and the relative frequency for data objects follows a Zipf-like distribution [14]. Specifically, let O_i be the ith most popular object ($1 \leq i \leq N$). Then, the probability that an data access is for O_i is given by

$$p_i = \frac{\Omega}{i^{\alpha}}, \quad (1)$$

where $\Omega = \left(\sum_{i=1}^{N} \frac{1}{i^{\alpha}}\right)^{-1}$ and α $(0 \leq \alpha \leq 1)$ is a constant determining the skewness in the Zipf-like distribution and α of 0.8 is used in simulations.

Two types of events are defined: Access and Update. The inter-Access event arrival time for O_i follows an exponential distribution with rate μ_i, which is given by $p_i\mu$ where μ is the net access rate to data objects. On the other hand, the inter-Update event arrival time is drawn from a Gamma distribution with mean $1/\lambda$ and the variance ν. As an important performance parameter, the access-to-update ratio ρ is defined as

$$\rho = \frac{\sum_{i=1}^{N} \mu_i}{N\lambda} = \frac{\mu}{N\lambda}. \quad (2)$$

In the simulations, we measure the traffic volume for a data access event, which can be calculated as the the product of the message size and the hop distance [15]. However, different types of links (i.e., wired, WWAN, WLAN links) are involved in mobile hotspots, so that their bandwidth characteristics should be considered for the transmission cost. Accordingly, we define the weighted transmission cost as the transmission cost divided by the corresponding link bandwidth, and thus its unit is $bytes * hops/Mbps$. Let S_{access}, S_{update}, and S_{ack} denote the sizes of **Access**, **Update**, and **Ack** messages, respectively. In addition, S_{send} and S_{data} represent the size of **Send** message without a data object and the size **Send** message with a data object, respectively. The WWAN and WLAN links are one-hop link whereas the wired link is H hops. The bandwidth in each link is defined in Section II and then the weighted transmission cost in each part can be calculated. The weighted total transmission cost C_T can be computed by the sum of the weighted transmission costs in wired, WWAN, WLAN links. For instance, in PER, an **Access** message is sent by the MT and no cached object exists in the MT cache, the transmission cost for this case is $S_{access}/(L+W_{Up}+B)+(S_{send}+S_{data})/(L+W_{Down}+B)$. Default parameter values for simulations are summarized in Table I, which are based on [9].

TABLE I

Default parameter values for simulation.

S_{access}	S_{update}	S_{ack}	S_{send}	S_{data}	N
45 bytes	45 bytes	45 bytes	45 bytes	727 bytes	100
W_{Down}	W_{Up}	B	L	K_{MT}	H
384 Kbps	96 Kbps	100 Mbps	11 Mbps	20	10

IV. Simulation Results

A. Effect of ρ

Figure 4 illustrates the effect of the access to update ratio ρ. When ρ is low, the update rate dominates the access rate. Therefore, the possibility that the MT cache has a stable

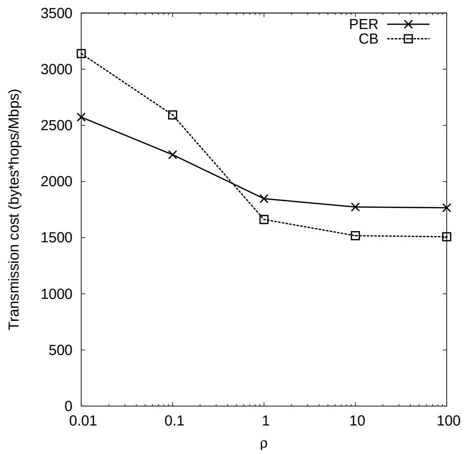

Fig. 4. Effects of ρ.

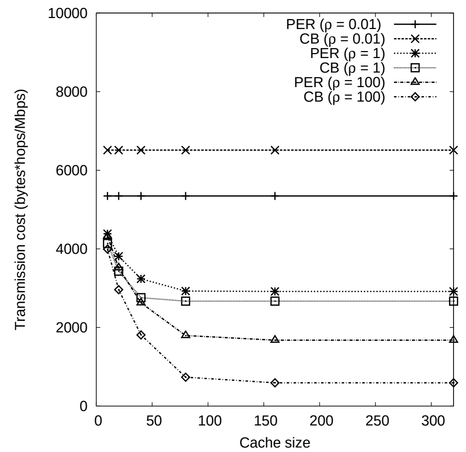

Fig. 5. Effect of cache size.

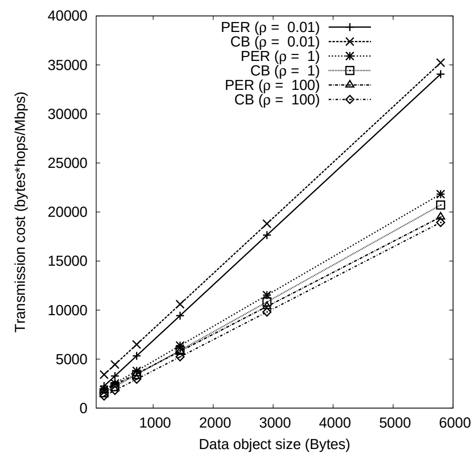

Fig. 6. Effects of S_{data}.

data object is high, and it reduces the effectiveness of cache significantly. On the other hand, since object updates occur infrequently for high ρ, the MT cache can be referenced more actively. Consequently, it can be seen that the transmission cost decreases as ρ increases. As shown in Figure 4, PER shows better performance than CB when ρ is low. On the other hand, CB outperforms PER for high ρ. In other words, no single algorithm always shows the best performance in different ρ. There exists a threshold the transmission costs of PER and CB are balanced. Therefore, it is a good approach to adaptively choose either PER or CB depending on ρ.

B. Effect of Cache Size

Figure 5 shows the effect of the cache size in different ρ. It can be seen that the cache size has no significant impact on the transmission cost for low ρ. This is because the object access events are not frequent in low ρ. Therefore, the transmission cost cannot be reduced even though a larger MT cache is employed. On the other hand, for $\rho = 1$ and $\rho = 100$, the transmission cost decreases as the MT cache size increases. In other words, object access events can be resolved by the cached object at the MT cache in these cases. However, when the cache size exceeds a certain point, i.e., 40 for $\rho = 1$ and 80 for $\rho = 100$, the gain of a large cache becomes insignificant. This is due to the total number of objects used in simulations is fixed at 100. Hence, it can be shown that the optimal cache size is dependent on the total number of objects. Also, the optimal cache size is affected by ρ. Specifically, a larger size cache is required for high ρ.

C. Effect of Data Object Size

The transmission cost is largely dependent on the data object size S_{data}. Especially, the size of a multimedia data object (e.g., image, video, etc.) is typically larger than the text-based data object. As shown in Figure 6, the transmission cost increases drastically as the data object size increases. However, the difference between PER and CB remains regardless of the data object size. Obviously, the main bottleneck in mobile hotspots is the WWAN link. Therefore, it is required to reduce the transmission cost over the WWAN link. To this end, a proxy cache can be installed at the vehicle and it can mitigate the impact of large size data objects [16].

D. Effect of WWAN Bandwidth

As mentioned before, the WWAN link is the main bottleneck for improving the performance of wireless data access algorithms. Therefore, we investigate the effect of WWAN bandwidth. From Figure 7, it can be seen that the transmission cost can be significantly reduced by increasing the WWAN downlink transmission for all ρ. On the other hand, Figure 8 indicates that the transmission cost can be also decreased when a larger WWAN uplink bandwidth is allocated. However, the decreasing rate becomes minor when the allocated WWAN uplink bandwidth is larger than a specific value, i.e., around 192 Kbps. In wireless data access applications, the uplink is mainly used for transmissions of object request messages and acknowledgement messages, and the message sizes are smaller than data objects. Therefore, a relatively small uplink bandwidth is sufficient to reduce the transmission cost.

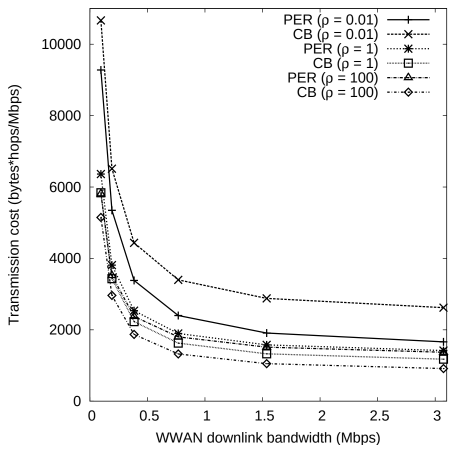

Fig. 7. Effects of WWAN downlink bandwidth.

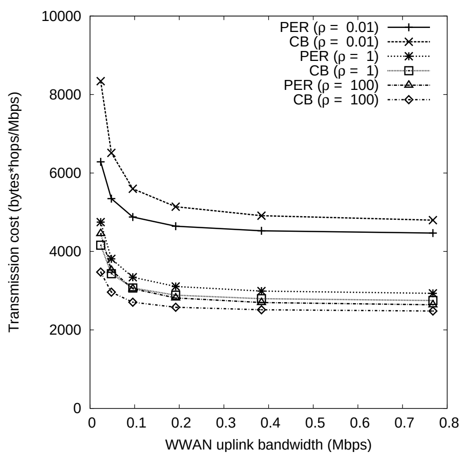

Fig. 8. Effect of WWAN uplink bandwidth.

E. Effect of WLAN Data Rate

Current IEEE 802.11b specification defines multiple data rates, which are determined by channel conditions. Also, IEEE 802.11a provides a high data rate of 54 Mbps, and multiple data rates are supported. Figure 9 shows the transmission cost under different WLAN data rates. It can be seen that there is no apparent decrease in the transmission cost even though a higher WLAN data rate is provided. This is because the WLAN data rate of 11 Mbps is substantially larger than the WWAN bandwidth, and thus the WWAN link is a performance bottleneck rather than the WLAN link. Consequently, it can be concluded that WWAN bandwidth allocation is a more critical issue than increasing WLAN data rates.

V. Conclusion

In this paper, we have analyzed the performance of PER and CB in mobile hotspots through extensive simulations. Simulation results demonstrate that the performance of wireless data access algorithms is highly sensitive to the access-to-update ratio. In addition, the access-to-update ratio affects the optimal cache size. It can be also shown that the WWAN link has a significant impact on the transmission cost of wireless data access algorithms in mobile hotspots, and this result indicates that a suitable WWAN bandwidth allocation method should be devised to improve the performance. In our future work, we will investigate an adaptive wireless data access algorithm in mobile hotspots.

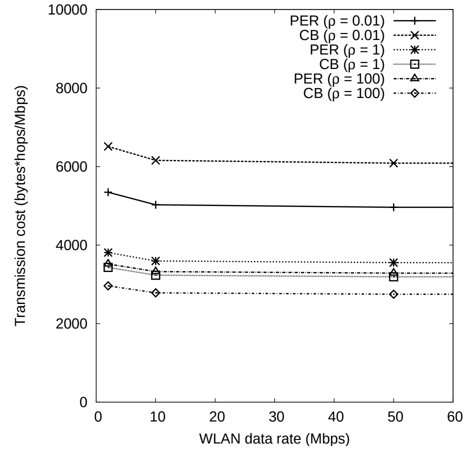

Fig. 9. Effect of WLAN data rate.

Acknowledgment

This work was supported in part by the Brain Korea (BK) 21 project of the Ministry of Education, 2006, Korea.

References

[1] J. Ott and D. Kutscher, “Drive-thru Internet: IEEE 802.11b for “Automobile” Users,” in *Proc. IEEE INFOCOM 2004*, March 2004.

[2] D. Ho and S. Valaee, “Information Raining and Optimal Link-Layer Design for Mobile Hotspots,” *IEEE Transactions on Mobile Computing*, 4(3), pp. 271-284, May/June 2005.

[3] V. Devarapalli et al., “Network Mobility (NEMO) Basic Support Protocol,” IETF RFC 3963, January 2005.

[4] A. Iera et al., “End-to-End QoS Provisioning in 4G with Mobile Hotspots,” *IEEE Network*, 19(5), pp. 26-34, September/October 2005.

[5] P. Rodriguez et al., “MAR: A Commuter Router Infrastructure for the Mobile Internet,” in *Proc. ACM MOBISYS 2004*, June 2004.

[6] K. Lan et al., “MOBNET: The Design and Implementation of a Network Mobility Testbed for NEMO protocol,” in *Proc. IEEE LANMAN 2005*, September 2005.

[7] J. Howard et al., “Scale and performance in a distributed file system,” *ACM Trans. Comput. Syst.*, 6(1), pp. 51-58, Feburuary 1988.

[8] M. Nelson et al., “Caching in the sprite network file system,” *ACM Trans. Comput. Syst.*, 6(1), pp. 134-154, Feburuary. 1988.

[9] Y. Lin et al., “Effects of Cache Mechanism on Wireless Data Access,” *IEEE Trans. Wireless Communications*, 2(6), pp. 1247-1258, Nov. 2003.

[10] Y. Fang and Y. Lin, “Strongly Consistent Access Algorithms for Wireless Data Networks,” *ACM WINET*, 11(3), pp. 243-254, May 2005.

[11] H. Chen et al., “Update-based Cache Replacement Policies in Wireless Data Access,” *IEEE Trans. on Mobile Computing*, to appear.

[12] Y. Xiao and H. Chen, “An Adaptive Callback Cache Access for Wireless Internet,” in *Proc. IEEE Globecom 2005*, December 2005.

[13] S. Pack et al., “Performance Analysis of Mobile Hotspots with Heterogeneous Wireless Links,” submitted for publication, 2006.

[14] L. Berslau et al., “Web Caching and Zipf-like Distributions: Evidence and Implications,” in *Proc. IEEE INFOCOM*, March 1999.

[15] S. Lo et al., “Architecture for Mobility and QoS Support in All-IP Wireless Networks,” *IEEE IEEE Journal on Selected Areas in Communications*, 22(4), pp. 691-705, May 2004.

[16] S. Pack et al., “Proxy-based Wireless Data Access Algorithms in Intelligent Transportation Systems,” submitted for publication, 2006.