This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2007 proceedings.

Efficient Data Access Algorithms for ITS-based Networks with Multi-Hop Wireless Links

Sangheon Pack
School of EE
Korea University, Korea
Email: shpack@korea.ac.kr

Humphrey Rutagemwa, Xuemin Shen, Jon Mark
Department of Electrical & Computer Engineering
University of Waterloo, Canada
Email: {humphrey,xshen,jwmark}@bbcr.uwaterloo.ca

Kunwoo Park
School of CSE
Seoul National University, Korea
Email: kwpark@mmlab.snu.ac.kr

***Abstract*— In this paper, we investigate efficient data access algorithms in intelligent transportation system (ITS)-based networks with multi-hop wireless links. We introduce a proxy cache (PC) and propose PC-based poll-each-read (P-PER) and PC-based callback (P-CB) data access algorithms to reduce the transmission cost over the bottleneck wireless links. Extensive simulation results are given to demonstrate the performance of P-PER and P-CB. It is shown that P-PER and P-CB can improve the cache hit performance and reduce the transmission cost significantly. A tradeoff between P-PER and P-CB suggests the need to use a hybrid proxy-based approach to attain optimal performance of data access in ITS-based networks with multi-hop wireless links.**

I. Introduction

Traffic congestion in transportation systems causes accidents and delay which can result in significant loss of lives, waste of energy, and loss in productivity. The global efforts to reduce traffic congestion, enhance productivity, and save lives, time, money, energy and the environment have led to the development of intelligent transportation systems (ITS). ITS incorporates a number of information technologies which are integrated in the vehicles and transport infrastructures to facilitate vehicle-to-infrastructure communications or vehicle-to-vehicle communications. Recent advances in wireless communication technologies (e.g., Universal Mobile Telecommunication System (UMTS), IEEE 802.11 (Wi-Fi), and IEEE 802.16 (WiMAX)) coupled with the increasing amount of time people spend on the road have open up new applications for highly dynamic ITS-based mobile networks that enhance the existing intelligent transport systems. Figure 1 shows a network scenario for emerging applications in ITS-based networks with multi-hop wireless links. The mobile terminal (MT) can access the application server (AS) through the access point (AP) and the base station (BS). The first wireless hop from the MT is the wireless local area network (WLAN) link and the second wireless hop is the wireless wide area network (WWAN) link. The integrated WLAN-WWAN link provides several advantages in mobility management and resource management [1].

In order for users who are on the road to enjoy seamless mobile services, the ITS-based networks need to be deployed extensively. However, the deployment of ITS-based networks with multi-hop wireless links poses several research challenges

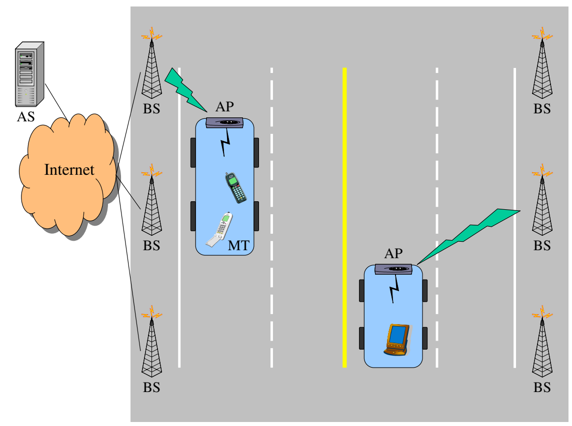


Fig. 1. Wireless data access in ITS-based networks with multi-hop wireless links.

such as latency and throughput performance improvement, QoS provisioning, mobility managements, etc.

This study focuses on issues related to latency and throughput enhancement. In the literature, several schemes have been proposed for wireless data access applications [2] and distribution applications [3], [4] in ITS-based mobile networks. Persone *et al.* [2] consider an information service for personal digital assistant (PDA)-based navigation tools and propose a zone-based mobility model for analyzing the performance. Wischhof *et al.* [3] develop a segment-oriented data abstraction and dissemination (SODAD) scheme for self-organizing inter-vehicle networks. Munaka *et al.* [4] propose a reliable multicast system consisting of a multicast group management scheme, a data-retransmission scheme, and a data recovery processing method. Unfortunately, these schemes do not address the potential latency and throughput penalty because of the bottleneck posed by the wireless link in ITS-based mobile networks.

Data caching is one of the promising techniques that can be used to enhance the latency and throughput performance of data access applications running over ITS-based networks with multi-hop wireless links. The use of data caching requires the identification of the nodes where data can be cached, the appropriate cache replacement policy, and the degree of cache consistency. Even though the data object can be cached at any node along the data path, only nodes that can significantly

1-4244-0353-7/07/$25.00 ©2007 IEEE

improve the performance should be considered. Usually, all nodes in a data path have limited memory to store cached data object. Therefore, an efficient policy is required to determine cached data object that can be overwritten when the reserved memory space allocated for data caching is full. Cached data object may be required to maintain strong consistency or weak consistency depending on the nature of data applications. For weak consistency, a stale copy of the data may be returned to the user whereas for strong consistency, the consistency between cached copies and the original ones is always enforced and no stale copy of the modified data is allowed to be used.

In this paper, we consider wireless data access applications that require a strongly consistent data cache in ITS-based networks with multi-hop wireless link. We introduce a proxy cache (PC) and propose two enhanced strongly consistent data access algorithms: *PC-based poll-each-read (P-PER)* and *PC-based callback (P-CB)*. Through extensive simulations, we evaluate the performance of P-PER and P-CB in terms of the cache hit probability and the transmission cost. In addition, the effects of cache size, access pattern, and data size are investigated. To the best of our knowledge, this is the first research work in the open literature that focuses on the strongly consistent data access algorithms in ITS-based networks with multi-hop wireless links.

The remainder of this paper is organized as follows. Section II describes the system model under consideration. Proxy-based data access algorithms are proposed in Section III. Extensive simulations results are given in Section IV, followed by concluding remarks in Section V.

II. System Description

We consider data access applications where MTs in a public transportation system access data objects in the AS as depicted in Figure 1. An MT connects to an AS through the WLAN-WWAN integrated link. WLAN supports higher data rate than WWAN, but it has a smaller service coverage area than WWAN. Consequently, WLAN is used to connect a number of MTs to the AP whereas WWAN is used to connect the AP and the BS. The wireline link is used to connect the BS to the Internet. We assume that data access applications are running over reliable transport and/or data link protocols. Therefore, at the application layer, the WWAN and WLAN links can apparently be characterized as having negligible packet losses. In addition, 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 the WLAN link are considered to have symmetrical bandwidths of B Mbps and L Mbps, respectively. On the other hand, the WWAN link is considered to have asymmetrical links where uplink and downlink bandwidths are W_{Up} Kbps and W_{Down} Kbps, respectively. We assume that all modifications to objects are only made by the AS. The MT has a cache with a limited size K_{MT}. The AP is installed in a public transportation system, and the proxy cache (PC) of a size K_{PC} is co-located with the AP. The PC is shared by multiple MTs and it has little or no concern for size and processing capability. Therefore K_{PC} is much larger than K_{MT}.

III. Proposed Data Access Algorithms

Poll-each-read (PER) [9] and *callback (CB)* [10] are two strongly consistent wireless data access algorithms reported in the literature. In PER, the MT always attempts to read data objects from the AS. On the other hand, in CB, the AS always informs MTs that a data object is modified and the cache should be invalidated. These algorithms have been analyzed in [5], [6] and extended in [7], [8]. In this study, we consider data access applications that run over ITS-based networks. Since the network model under consideration is different from previously considered network models, new data caching algorithms need to be established and studied. From Section II, the WWAN link is the bottleneck due to its limited bandwidth. Consequently, in our proposed algorithms, we consider a two-tier caching architecture where the PC is installed at the AP in order to reduce transmission cost over the WWAN link. At the first tier, the MT and the PC respectively act as a client and a server, and the cache in the MT is utilized. At the second tier, the PC plays the role of a client whereas the AS acts as a server. In the rest of this section, we first introduce the basic terminologies and then describe the operations of the proposed *PC-based PER (P-PER)* and *PC-based CB (P-CB)* algorithms.

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 (i.e., O_i with $t + \Delta$ ($\Delta > 0$) is a more recent data object than O_i with t). Each MT is identified by an identifier j and a transportation system has a group identifier k. For instance, an MT j that resides in a transportation system k can be uniquely identified by (j, k). The following terminologies are defined for wireless data access algorithms.

- **Access**(i, t): This message requests an access of the object i. For PER, $t > 0$ specifies the current sequence number for a cached object whereas $t = 0$ represents that there is no object in the cache. For CB, invalidation is always performed by the AS and t is always set to zero.
- **Send**(i, t, F): This message is used to send a data object or to confirm **Access**(i, t) in PER. i and t denote the object index and the sequence number, respectively. F is a flag indicating whether the data object is included in the message or not. If the object i is transmitted with this message, F is set to one. On the other hand, if only a confirmation message is sent, F is set to zero.
- **Update**(i, j, k): This message invalidates the object i in the cache of MT j located in the transportation system k.
- **Ack**(i, j, k): This message acknowledges the receipt of **Update**(i, j, k).

A. PC-based Poll-Each-Read (P-PER)

To describe the operations of P-PER, we consider four possible cases: 1) there are no cached objects in both the MT

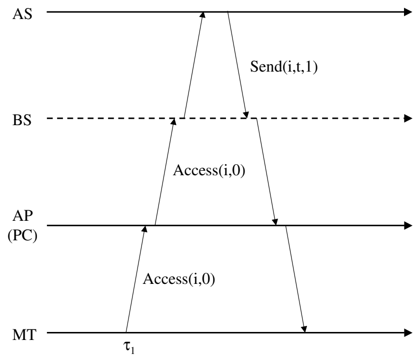


Fig. 2. P-PER operation: There are no cached objects both in the MT and the PC.

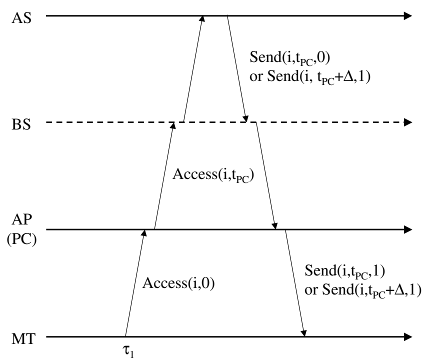


Fig. 3. P-PER operation: There is no cached object in the MT, but there is one in the PC.

and the PC; 2) there is no cached object in the MT, but there is one in the PC; 3) there is a cached object in the MT, but not in the PC; and 4) there are cached objects in both the MT and the PC.

1) There are no cached objects in both the MT and the PC: Figure 2 shows the P-PER operation. The MT first sends **Access(**i**,** 0**)** to the PC. Since the PC also does not have any cached object for O_i, it relays **Access(**i**,** 0**)** to the AS. After that, the AS delivers the up-to-date O_i via **Send(**i**,** t**,** 1**)**, and the PC and the MT maintain the object in their caches.

2) There is no cached object in the MT, but there is one in the PC: From Figure 3, the MT has no cached object for O_i, so it sends **Access(**i**,** 0**)** to the PC. Assume that the PC has a cached object and its time sequence is t_{PC}. After receiving **Access(**i**,** 0**)** from the MT, the PC converts the message into **Access(**i**,** t_{PC}**)** and transmits **Access(**i**,** t_{PC}**)** to the AS. If there is no modification to O_i after t_{PC}, **Send(**i**,** t_{PC}**,** 0**)** is returned; otherwise, **Send(**i**,** $t_{PC}+\Delta$**,** 1**)** is sent to the PC. When the PC receives **Send(**i**,** t_{PC}**,** 0**)**, it is confirmed that the PC's cached object is the latest object. Therefore, no transmission cost for the data object incurs between the AS and the PC, and the PC delivers its cached data object to the MT. On the other hand, if the PC receives **Send(**i**,** $t_{PC} + \Delta$**,** 1**)**, the PC updates its cache as the new object and relays it to the MT.

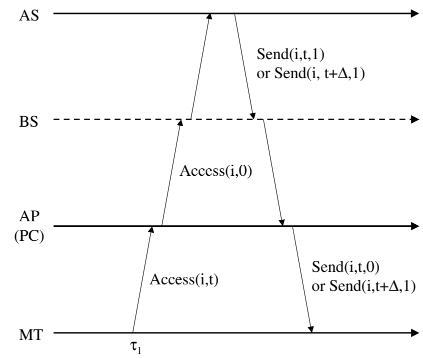


Fig. 4. P-PER operation: There is a cached object in the MT, but not in the PC.

3) There is a cached object in the MT, but not in the PC: As mentioned before, the PC is shared by multiple MTs. Therefore, a cached object in the PC can be replaced by accesses of other objects. In this case, only the MT has a cached object. Figure 4 shows the P-PER operation for this situation. The MT sends **Access(**i**,** t**)** to the PC, and then the PC performs a conversion of the message. This conversion is required because the PC should cache the up-to-date O_i by accessing the AS. If **Access(**i**,** t**)** is sent to the AS without conversion and no modification occurs after t, the PC cannot cache the recent O_i. Therefore, the PC sends **Access(**i**,** 0**)** instead of **Access(**i**,** t**)**. When the AS receives **Access(**i**,** 0**)**, it responds with **Send(**i**,** t**,** 1**)** or **Send(**i**,** $t{+}\Delta$**,** 1**)**. For both cases, the PC caches the received data object. On the other hand, the reception of **Send(**i**,** t**,** 1**)** represents that the MT has the up-to-date object and hence the PC sends **Send(**i**,** t**,** 0**)** to the MT, which saves the transmission cost between the AP and the MT. For **Send(**i**,** $t + \Delta$**,** 1**)**, since there is a modification to O_i after the MT's data access, the PC delivers the recent data object O_i via **Send(**i**,** $t + \Delta$**,** 1**)**.

4) There are cached objects in both the MT and the PC: Figure 5 indicates the P-PER operation when both the MT and the PC have cached objects for O_i. Let t_{PC} and t be the time sequences of the cached object in the PC and the MT, respectively. Intuitively, t_{PC} is always equal to or larger than t, so that the PC should check whether t_{PC} is the latest time sequence of O_i. Therefore, when the PC receives **Access(**i**,** t**)** from the MT, it converts **Access(**i**,** t**)** into **Access(**i**,** t_{PC}**)** and delivers **Access(**i**,** t_{PC}**)** to the AS. If O_i is modified after t_{PC}, **Send(**i**,** $t_{PC}+\Delta$**,** 1**)** will be returned; otherwise, **Send(**i**,** t_{PC}**,** 0**)** is sent to the PC. If the PC receives **Send(**i**,** $t_{PC} + \Delta$**,** 1**)**, it caches the latest O_i and then relays the data object to the MT. On the other hand, if the PC receives **Send(**i**,** t_{PC}**,** 0**)** and $t_{PC} = t$, the PC does not need to deliver the data object to the MT. Hence, **Send(**i**,** t**,** 0**)** is sent to the MT and then the MT uses its cached object for its data access. However, if $t_{PC} > t$ and **Send(**i**,** t_{PC}**,** 0**)** is received, the PC delivers the up-to-date O_i from its cache to the MT using **Send(**i**,** t_{PC}**,** 1**)**.

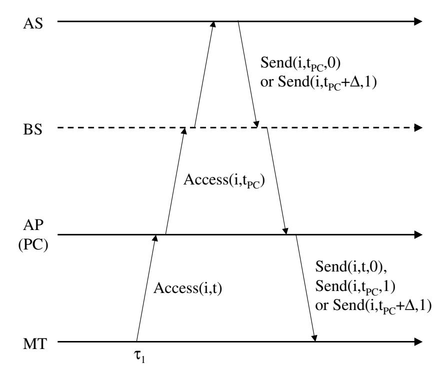


Fig. 5. P-PER operation: There are no cached objects in both the MT and the PC.

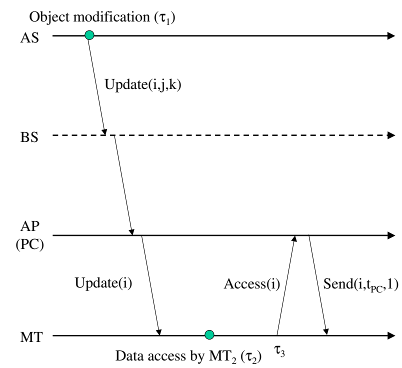


Fig. 6. P-CB operation: a cached object in the PC.

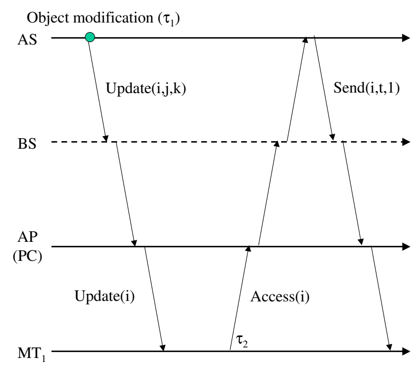


Fig. 7. P-CB operation: no cached object in the PC.

B. PC-based Callback (P-CB)

We describe the P-CB operations by considering two cases: 1) the PC has a cached object for O_i; and 2) the PC does not have any cached object for O_i.

1) The PC has a cached object: Figure 6 illustrates the operation of P-CB when the PC has a cached object for O_i. Let j and k be the indexes of the MT and the vehicle where the MT resides, respectively. At time τ_1, an object O_i is modified and hence the AS sends **Update**(i**,** j**,** k) to invalidate O_i maintained by MT j. When the PC receives **Update**(i**,** j**,** k), it also invalidates its cached object. In addition, the PC invalidates the cached object at the MT by sending **Update**(i). In WLAN, link layer broadcast can be supported. Therefore, the PC broadcasts **Update**(i) within WLAN where the corresponding MT locates. Since broadcast is used, the indexes of the MT and the vehicle (i.e., j and k) are not needed. The broadcast-based invalidation is more effective when multiple MTs in vehicle k have cached objects. After the invalidation, MT j sends an acknowledgement message **Ack**(i**,** j**,** k), which is omitted in Figure 6. For reliable transmission in invalidation, the PC re-broadcasts **Update**(i) if it does not receive any acknowledgement message from MT j. At time τ_2, another MT MT_2 accesses O_i. As a result, the PC has a cached object with the time sequence t_{PC}. Therefore, the PC can resolve **Access**(i) sent by the MT at time τ_3 by referencing its cache, and then the PC sends its cached object to the MT via **Send**(i**,** t_{PC}**,** 1). Consequently, the data object transmission occurs only in the WLAN link. This is a representative advantage that can be achieved by the PC-based wireless data access algorithms. In other words, even though an MT does not access an object after the object update, the MT can reduce the transmission cost by contacting the PC if there is a data access by other MTs. This benefit becomes significant as the popularity of the object increases, which will be investigated in Section IV.

2) The PC has no cached object: Figure 7 shows the P-CB operation when the PC has no cached object for O_i. This situation can happen if: 1) there is no MT within the vehicle accesses O_i after the invalidation at time τ_1; or 2) there is any MT within the vehicle accesses O_i after τ_1, but the cached object is replaced. Since both the MT and the PC have no cached objects, **Access**(i) is transmitted to the AS and the AS replies with **Send**(i**,** t**,** 1). Then, the PC and the MT maintain the data object O_i in their caches until it is invalidated or replaced.

IV. SIMULATION RESULTS

To evaluate the performance of P-PER and P-CB, we develop an event-driven simulator and run extensive simulations. N data objects are assumed to be in the AS and the relative frequency for data objects follows a Zipf-like distribution [11]. Let O_i be the ith most popular object ($1 \leq i \leq N$). The probability of access to O_i is given by

$$p_i = \frac{\Omega}{i^\alpha}, \tag{1}$$

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

We consider two types of events: Access and Update [5]. 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. The inter-Update event arrival time is drawn from a Gamma distribution with mean $1/\lambda$ and variance ν. As an important performance parameter,

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	H	α
384 Kbps	96 Kbps	100 Mbps	11 Mbps	10	0.8

we define the access-to-update ratio ρ as

$$\rho = \frac{\sum_{i=1}^{N} \mu_i}{N\lambda} = \frac{\mu}{N\lambda} \tag{2}$$

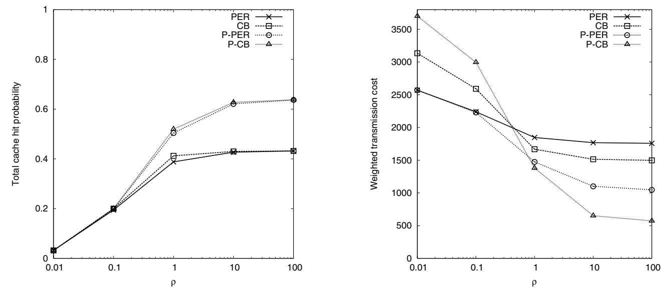
to culminate two events in a single parameter.

The performance metrics are the total cache hit probability (p) and the weighted transmission cost (C_T). The total cache hit probability is defined as the probability that a data access can be resolved by a cached object at the MT cache or the PC. Let p_{MT} and p_{PC} be the MT cache hit probability and the PC cache hit probability, respectively. p_{MT} is computed as N_{hit}^{MT}/N_a, where N_{hit}^{MT} is the number of data accesses resolved by the MT cache and N_a is the number of total data accesses. p_{PC} is determined by N_{hit}^{PC}/N_a, where N_{hit}^{PC} is the number of data accesses that cannot be resolved by the MT cache but can be resolved by the PC. Since only the MT cache is employed in PER and CB, p_{MT} is the total cache hit probability. On the other hand, the total cache hit probability in P-PER and P-CB can be expressed as $p = p_{MT} + (1 - p_{MT}) \cdot p_{PC}$.

The transmission cost can be found by considering the traffic volume for a data access event. As in [12], it is calculated as the product of the message size and the hop distance. Since ITS-based mobile networks are characterized with heterogeneous links (i.e., wired, WWAN, and WLAN links) with different bandwidths, the aforementioned approach cannot be used as is. In this study, we define the weighted transmission cost as the transmission cost divided by the corresponding link bandwidth, and its unit is given by $bytes * hops/Mbps$. Let S_{access}, S_{update}, and S_{ack} denote the sizes of **Access**, **Update**, and **Ack** messages, respectively. S_{send} and S_{data} represent the size of **Send** message without a data object and the size of **Send** message with a data object, respectively. The WWAN and WLAN links are one-hop links whereas the wired link is H hops. The weighted total transmission cost C_T can be computed by the sum of the weighted transmission costs in the wired, WWAN, and WLAN links. For instance, when an **Access** message is sent by the MT and the message is resolved by the PC using the cached object, the transmission cost for this case is $S_{access}/L + (S_{send} + S_{data})/L$. The default parameter values for the simulations are derived from [5] and summarized in Table I.

A. Effect of Access-to-Update Ratio (ρ)

Figure 8 illustrates the effect of the access-to-update ratio (ρ). It can be seen that the total cache hit probability increases as ρ increases, but the weighted transmission cost decreases as ρ increases. This can be explained as follows. When ρ is low, the update rate dominates the access rate. Therefore, the possibility that the MT cache or the PC has a stable data object is high. This reduces the effectiveness of cache. On the other hand, when ρ is high, the access events occur more frequently than the object update events, and thus the cache can be actively referenced. As shown in Figure 8, P-PER yields the best performance when ρ is low and P-CB gives the best performance for high ρ. In other words, there is no single algorithm that always yields the best performance for all possible values of ρ. This means that it is possible to minimize the weighted transmission cost by integrating P-PER and P-CB, i.e., a hybrid approach.



(a) Total cache hit probability (b) Weighted transmission cost

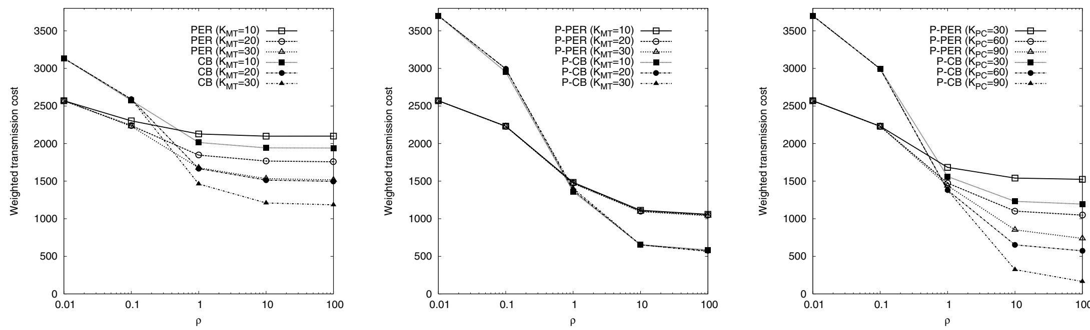
Fig. 8. Effect of ρ ($K_{MT} = 20$ and $K_{PC} = 60$).

B. Effect of Cache Size

From Figure 9, the performance of PER and CB can be significantly improved by increasing K_{MT}. In PER and CB, if the cached object in the MT is not valid, the MT accesses the AS and therefore incurs high transmission cost. Hence, if a large K_{MT} is used, the total cache hit probability can be increased and the weighted transmission cost can be significantly reduced. The performance of P-PER and P-CB can also be improved as K_{MT} increases. However, as shown in Figure 9, it can be seen that the effect of K_{MT} is not significant in P-PER and P-CB. In P-PER and P-CB, some cache misses at the MT cache can be resolved at the PC. Therefore, the total cache hit probability considering both the MT cache hit and the PC cache hit is not highly sensitive to K_{MT}. On the other hand, Figure 9 indicates that the total cache hit probability and the weighted transmission cost are significantly affected by K_{PC} in P-PER and P-CB. Namely, as K_{PC} increases, the total cache hit probability increases and the weighted transmission cost decreases. Unlike the MT cache, the PC has little concern in the cache size because it is installed in a vehicle with sufficient processing power. Therefore, a larger K_{PC} can be easily deployed and improve the cache performance of P-PER and P-CB in ITS-based networks with multi-hop wireless links.

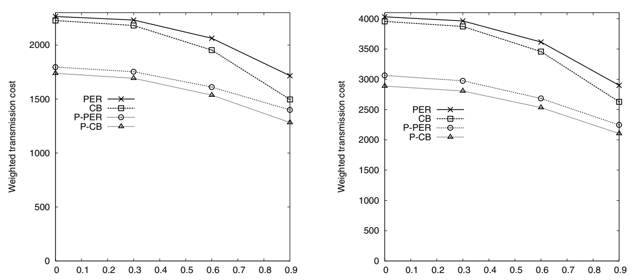
C. Effect of Data Object Size (S_{data})

It is expected that ITS-based networks will support different types of wireless data access applications, e.g., traffic/road/weather information search, real-time news report, multimedia contents delivery, etc. These heterogeneous applications have different data object sizes. Since the weighted



(a) PER and CB (K_{MT}) (b) P-PER and P-CB (K_{MT}) (c) P-PER and P-CB (K_{PC})

Fig. 9. Effects of K_{MT} and K_{PC}.



(a) $S_{data} = 727$ bytes (b) $S_{data} = 1500$ bytes

Fig. 10. Effect of S_{data}.

transmission cost is largely dependent on the data object size S_{data}, we investigate the performance of P-PER and P-CB in different data object sizes. Figure 10 demonstrates that the gains of P-PER and P-CB increase as S_{data} increases. This is because P-PER and P-CB can reduce the transmission cost over WWAN with the least radio resource, by introducing the PC installed in a vehicle (i.e., AP). Therefore, P-PER and P-CB are more suitable for applications with large size objects, e.g., multimedia data objects.

V. Conclusion

In this paper, we have introduced a proxy cache (PC) installed in ITS-based network and have proposed two proxy-based data access algorithms, namely PC-based poll-each-read (P-PER) and PC-based callback (P-CB), for ITS-based networks with multi-hop wireless links. By utilizing the PC, P-PER and P-CB can improve the cache hit performance and reduce the transmission cost significantly. The effects of access-to-update ratio and cache size have been investigated through extensive simulations. It is shown that P-PER and P-CB can be effectively used under diverse environments. Consequently, P-PER and P-CB are viable solutions for improving the performance of data access applications in ITS-based networks. A trade-off between P-PER and P-CB in response to access-to-update ratio suggests that an adaptive approach can further improve the performance of wireless data access applications in ITS-based networks.

The proposed algorithms can be easily extended to vehicular ad-hoc networks or vehicle-to-vehicle communications. In these situations, a group of vehicles forms a WLAN and a proxy cache can be installed at an anchor vehicle to maintain connectivity with the BS. In our future work, we will introduce hybrid proxy-based wireless data access algorithms in ITS-based networks and investigate their performances over a wide range of network scenarios.

Acknowledgment

This work has been supported in part by a Strategic Grant from the NSERC of Canada under Grant No. STPGP 257682.

References

[1] S. Pack, H. Rutagemwa, X. Shen, J. Mark, and L. Cai, "Performance Analysis of Mobile Hotspots with Heterogeneous Wireless Links," *IEEE Transactions on Wireless Communications*, to appear.
[2] V. Persone and V. Grassi, "Performance Analysis of Caching and Prefetching Strategies for Palmtop-Based Nagigational Tools," *IEEE Transactions on Intelligent Transportation Systems*, 4(1), pp. 23-34, March 2003.
[3] L. Wischhof, A. Ebner, and H. Rohling, "Information Dissemination in Self-Organizing Intervehicle Networks," *IEEE Transactions on Intelligent Transportation Systems*, 6(1), pp. 90-101, March 2005.
[4] T. Munaka, T. Yamamoto, and T. Watanabe, "A Reliable Advanced-Join System for Data Multicasting in ITS Networks," *IEEE Transactions on Intelligent Transportation Systems*, 6(4), pp. 424-438, December 2005.
[5] Y. Lin and W. Lai, and J. Chen, "Effects of Cache Mechanism on Wireless Data Access," *IEEE Transactions on Wireless Communications*, 2(6), pp. 1247-1258, November 2003.
[6] Y. Fang and Y. Lin, "Strongly Consistent Access Algorithms for Wireless Data Networks," *ACM Wireless Networks*, 11(3), pp. 243-254, May 2005.
[7] H. Chen, Y. Xiao, and X. Shen, "Update-based Cache Replacement Policies in Wireless Data Access," *IEEE Transactions on Mobile Computing*, 5(12), pp. 1734-1748, December 2006.
[8] Y. Xiao and H. Chen, "An Adaptive Callback Cache Access for Wireless Internet," in *Proc. IEEE Globecom 2005*, December 2005.
[9] J. Howard, M. Kazar, S. Menees, D. Nichols, M. Satyanarayanan, R. Sidebotham, and M. West, "Scale and performance in a distributed file system," *ACM Transactions on Computer Systems*, 6(1), pp. 51-58, Feburuary 1988.
[10] M. Nelson, B.Welch, and J. Ousterhout, "Caching in the sprite network file system," *ACM Transactions on Computer Systems*, 6(1), pp. 134-154, Feburuary. 1988.
[11] L. Berslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web Caching and Zipf-like Distributions: Evidence and Implications," in *Proc. IEEE INFOCOM 1999*, March 1999.
[12] S. Lo, G. Lee, W. Chen, and J. Liu, "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.