This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2014.2314643, IEEE Transactions on Wireless Communications

FW-DAS: Fast Wireless Data Access Scheme in Mobile Networks

Giwon Lee, Insun Jang, Sangheon Pack, *Senior Member, IEEE*, and Xuemin (Sherman) Shen, *Fellow, IEEE*

***Abstract*—In wireless data access applications, it is essential to reduce both the access latency and the wireless traffic volume. In this paper, we propose a fast wireless data access scheme (FW-DAS) for wireless data access applications in which data objects are frequently updated and fast access to data objects is indispensable. In FW-DAS, different operation modes are defined depending on the data object popularity, and only popular data objects are proactively pushed to the access point/base station in order to minimize the access latency while mitigating the traffic load over the wireless link. An analytical model for the access latency is developed and an operation mode selection algorithm is introduced to reduce the access latency. Extensive simulation results show the effects of access-to-update ratio, data popularity, cache size, data object size, and wireless bandwidth. Analytical and simulation results demonstrate that FW-DAS can reduce the access latency with reasonable traffic load compared with poll-each-read (PER)/callback (CB) and their combinations.**

***Index Terms*—Wireless data access, strong consistency, fast wireless data access scheme, cloud-based services**

I. Introduction

RECENTLY, wireless network traffic is exponentially growing with the popularity of smart phones and pervasive wireless connectivity [2] and it will reach about 11.2 exabytes per month by 2017 [3]. Moreover, computing paradigm is evolving from a client-server model to a cloud computing model [4]–[6]. In the cloud computing model, user data are stored at a set of servers (a.k.a. cloud) and users freely access the data regardless of their locations. As one of the most promising trends in cloud computing, wireless data access applications with strong consistency[1] are becoming more and more popular since they can take a variety of deployment cases in the future [7], [8].

One typical wireless data access application with strong consistency is the stock information in which an up-to-date data object[2] should be always used by a mobile node (MN). An application server (AS) maintains time-sensitive stock information that can be changed every few seconds. Users can cache this information offered by the AS and synchronize it with that at the AS. News report and sports score delivery applications are auxiliary examples of wireless data access with strong consistency. Also, in cloud-based storage services (e.g., Dropbox and Google Drive) and social networking services (SNSs) (e.g., Facebook and Twitter), the consistency of cached data objects should be satisfied by means of wireless data access schemes.

Data caching is a key paradigm for improving the performance of cloud services in terms of end-user latency [9]–[11]. Poll-each-read (PER) and callback (CB) are two well-known strongly consistent wireless data caching schemes [12], [13]. In PER, whenever an MN accesses a data object, it first polls the AS to check the validity of its cached data object. If the cached data object is up-to-date, the AS sends a confirmation message without any data object. Otherwise, the AS sends a reply message with the updated data object to the MN. CB satisfies strong consistency by invalidation procedures. When a data object in the AS is modified, MNs with the corresponding cached data object are notified and the cached data object is invalidated. If an MN wants to access a data object in the AS and a cached data object exists, the MN uses the cached data object without any transmission cost. Otherwise, the MN contacts the AS to obtain the updated data object. In the previous study [12], it is shown that PER can reduce wireless network traffic compared with CB in wireless data access applications when updates for data objects are frequent. This is because the invalidation traffic in CB is significant when data objects are frequently changed. On the contrary, CB can reduce the access latency compared with PER because there is no need to contact the AS for a cache hit. In short, PER and CB have their pros and cons in supporting fast access of frequently updated data objects.

In this paper, we strike a balance between the reduction of wireless network traffic (in PER) and the acceleration of wireless data access (in CB). Specifically, we propose a fast wireless data access scheme (FW-DAS) that reduces both wireless network traffic and data object access latency. Since it is more critical to reduce the network traffic over the wireless link and a network cache can be employed at the access point (AP)/base station (BS) to reduce the access latency [14], [15], FW-DAS introduces two-tier operation: 1) the first-tier between MNs and the AP/BS (i.e., wireless link) and 2) the second-tier between the AP/BS and the AS (i.e., wired link). FW-DAS uses PER in the first-tier to mitigate wireless network traffic and introduces an extended CB scheme in the second-tier to minimize data object access latency. In addition, three

The preliminary version of this paper was presented at the International Conference on Computing, Networking and Communications (ICNC) 2013, San Diego, USA, January 2013 [1].

This work was supported by National Research Foundation of Korea Grant funded by the Korean Government (NRF-2012R1A1B4000894 and NRF-2012K2A1A2032693).

G. Lee, I. Jang, and S. Pack are with the School of Electrical Engineering, Korea University, Seoul, Korea. (e-mail:{goodkw, zerantoss, shpack}@korea.ac.kr)

X. Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. (e-mail: xshen@bbcr.uwaterloo.ca)

[1] Strong consistency requires that the data used by a mobile node should be always up-to-date.

[2] Throughout this paper, a data object represents a unit accessed by an MN in a wireless data access application.

operation modes for the second-tier, 1) invalidation and push mode, 2) invalidation and pull mode, and 3) invalidation only mode, are defined and one of operation modes is selected depending on the data object popularity. An analytical model is derived and an operation mode selection algorithm based on the analytical model is devised. Extensive simulations are carried out to validate the analytical model and to show the effects of access-to-update ratio, data popularity, cache size, data object size, and wireless bandwidth. The analytical and simulation results have shown that FW-DAS can reduce the access latency with reasonable traffic load compared with PER/CB and their combinations.

Main contributions of this paper are two-fold: 1) FW-DAS addresses a fundamental problem in wireless data access, i.e., joint reduction of the access latency and wireless network traffic. Therefore, FW-DAS can be widely used in mobile cloud applications requiring strong consistency; and 2) we develop the analytical model for FW-DAS and investigate how to choose the operation mode in FW-DAS by means of the analytical model. The analytical model is verified by extensive simulations, which can be used for optimizing the performance of FW-DAS.

The remainder of this paper is organized as follows. Related works are summarized in Section II. The system model is described in Section III and the fast wireless data access scheme is illustrated in Section IV. The analytical model and the operation mode selection algorithm are described in Sections V and VI, respectively. Simulation results are presented in Section VII, followed by concluding remarks in Section VIII.

II. Related Work

A number of research efforts have been conducted to satisfy strong consistency in wireless data access. Akon *et al.* [16] propose a strongly consistent and update-aware cache mechanism for wireless data access. Li and Chen [17] introduce an adaptive per-user and per-data object cache consistency management scheme in wireless mesh networks. Since the most popular algorithms in previous works are PER and CB, extensive works on PER and CB have been done in the literature. Lin *et al.* [12] analyze the effects of cache mechanisms on PER and CB. Fang and Lin [7] evaluate PER and CB under more general assumptions. Xiao and Chen [18] propose an optimal CB algorithm with two-level adaptation for wireless data access. Chen *et al.* [19] introduce a server-based PER algorithm in which the AS makes cache replacement decisions and a client-based CB algorithm in which clients make cache replacement decisions. Pack *et al.* [14] investigate wireless data access in mobile hotspots and propose a proxy cache-based PER and CB algorithms. Lee *et al.* [20] propose cooperative PER and cooperative CB wireless data access algorithms with strong consistency in multi-radio wireless networks.

These previous works focus on the cache hit rate or transmission cost while they do not pay attention to the reduction of the data access latency. Although a few works [21], [22] consider the reduction of access latency, they focus on particular applications such as SNSs. Therefore, it is worth investigating a fast data access scheme in generic wireless data access applications, which is the key motivation of this paper.

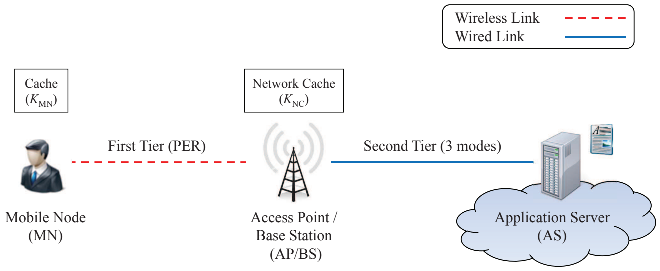


Fig. 1. System model for FW-DAS.

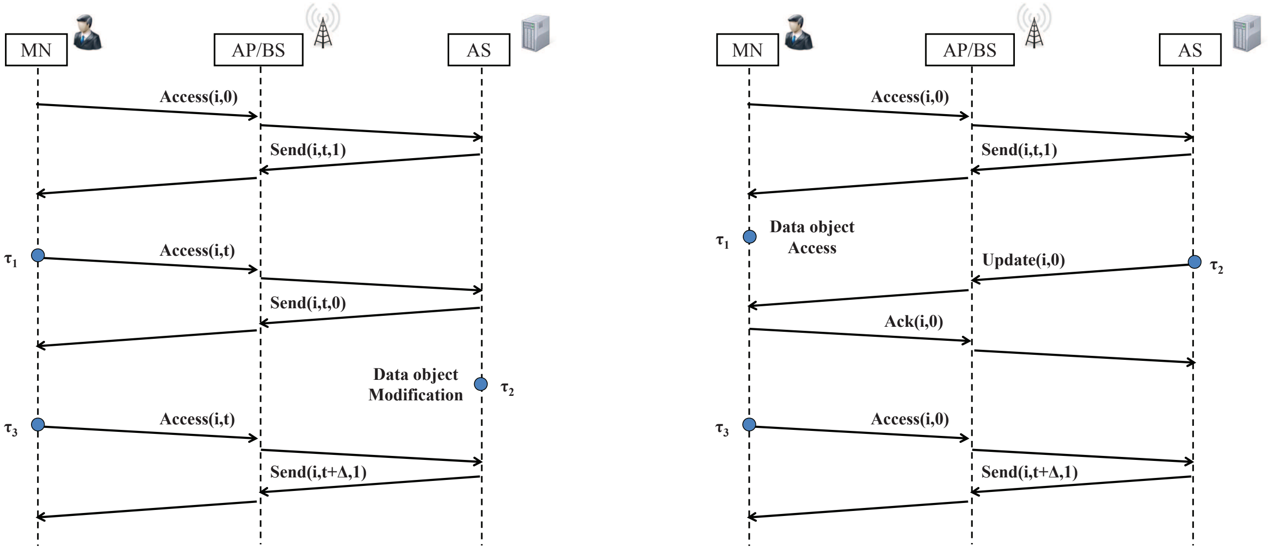
III. System Model

Figure 1 shows the system model for FW-DAS. An MN is connected to an AP/BS through the wireless link, whereas the AP/BS connects to an AS through the wired link. The MN has a cache with a limited size K_{MN} and a network cache with size K_{NC} is installed at the AP/BS. Because the network cache is shared by multiple MNs, K_{NC} is then set to be much larger than K_{MN}. By employing the network cache, it is possible to reduce the transmission cost and the access latency [14]. It is assumed that all modifications to data objects are only made by the AS.

To illustrate the operation of FW-DAS, we have the following terminologies. Let O_i be the ith data object. O_i is associated with a time sequence index t ($t \geq 0$), which is assigned in an increasing order (i.e., O_i with $t + \Delta$ ($\Delta > 0$) is a more recent data object than O_i with t). Four messages are defined for data access schemes [14].

- **Access(**i**,** t**)**: This message requests an access of the data object i. $t > 0$ specifies the current sequence number for the cached data object whereas $t = 0$ represents that there is no data object in the cache.
- **Send(**i**,** t**,** F**)**: This message is used to send the data object or to confirm **Access(**i**,** t**)**. i and t denote the data object index and the sequence number, respectively. F is a flag indicating whether the data object is included in the message or not. That is, when the data 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**,** F**)**: This message invalidates or updates the data object i. If F is 1, the data object is included in **Update(**i**,** F**)** and thus the corresponding data object can be updated. Otherwise (i.e., $F = 0$), **Update(**i**,** F**)** simply invalidates the data object i.
- **Ack(**i**,** R**)**: This message serves as the receipt of **Update(**i**,** F**)**. R is a flag indicating whether the data object is requested from the AP/BS or not. That is, when the AP/BS requests the data object i with this message, R is set to one. Otherwise, R should be set to zero.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2014.2314643, IEEE Transactions on Wireless Communications



(a) PER

(b) CB

Fig. 2. PER and CB operations.

IV. Fast Wireless Data Access Scheme

The main objective of FW-DAS is to reduce the access latency for popular data objects that are frequently updated (e.g., stock information) while mitigating the traffic load over the wireless link. To achieve this goal, we define two-tier operation: the first-tier between the MN and the AP/BS and the second-tier between the AP/BS and the AS. At the first-tier, to avoid high invalidation/update costs over the wireless link, PER is examined. Moreover, a modified CB scheme is devised for the second-tier to accelerate data object access. In this section, we first provide the overview of PER and CB. After that, the operations of FW-DAS at the first and second tiers are illustrated.

A. Overview of PER and CB

In PER, whenever an MN accesses a data object, it first polls the AS to check the validity of its cached data object. If the cached data object is up-to-date, the AS sends a confirmation message without any data object. Otherwise, the AS sends a reply message with the updated data object to the MN.

The message flow of PER is illustrated in Figure 2(a). The MN first tries to access a data object O_i. Since this is the first access to the data object O_i, the MN does not have any cached version. Therefore, the MN sends **Access(**i**,** 0**)** message to the AS and then the AS responds with **Send(**i**,** t**,** 1**)** message that includes the data object. At time τ_1, O_i is cached in the MN and the MN sends **Access(**i**,** t**)** message to check the validity of its cached data object. Then, the AS returns **Send(**i**,** t**,** 0**)** message and the MN uses the cached data object in its local cache. At time τ_2, O_i is modified and therefore the cached data object becomes invalid. After that, an access to the data object O_i is requested at time τ_3 and the MN sends **Access(**i**,** t**)** message for validity check. Then, the AS returns **Send(**i**,** $t + \Delta$**,** 1**)** message, where $\Delta > 0$, with the data object (i.e., F flag is set to one) to the MN, and then the MN updates its local cache.

Unlike PER, CB satisfies strong consistency through invalidation procedures. When a data object in the AS is modified, all MNs with corresponding cached data objects are notified and cached data objects are invalidated. When an MN wants to access a data object in the AS, it first checks the availability of the cached data object. If a cached data object exists, the MN uses the cached data object immediately. Otherwise, the MN contacts the AS to obtain the updated data object.

Figure 2(b) illustrates the CB operation. After the initial access to a data object O_i, the MN maintains the up-to-date data object O_i in its cache. Therefore, no access latency occurs when the cached data object is requested at time τ_1. When the data object O_i is updated at time τ_2, the AS transmits **Update(**i**,** 0**)** message to invalidate the cached data object and the MN responds with **Ack(**i**,** 0**)** message. When the MN accesses O_i at time τ_3 after the invalidation, the MN contacts the AS to get the updated O_i and then the AS responds with **Send(**i**,** $t + \Delta$**,** 1**)** message.

B. First-Tier Operation of FW-DAS

At the first-tier, the conventional PER scheme is used by the MN for data access. That is, the MN should always contact the AP/BS to check the validity of data object. Note that the AP/BS with the network cache maintains only up-to-date data objects by means of the second-tier operation (see Section IV-C). Therefore, if a cached data object exists at the AP/BS, the MN can access it without contacting the AS, which significantly reduces the access latency.

Detailed message flow in the first-tier is illustrated in Figure 3. At time τ_1, the MN first tries to access a data object O_i. Since this is the first access to the data object, the MN does not have any cached version. Therefore, the MN sends **Access(**i**,** 0**)** message to the AP/BS. If the AP/BS does not have

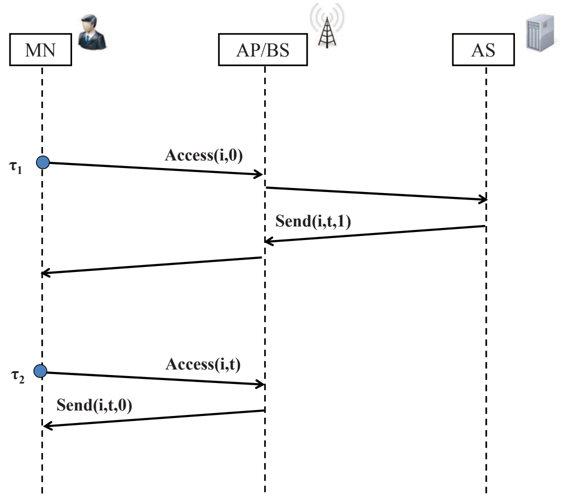


Fig. 3. First-tier operation.

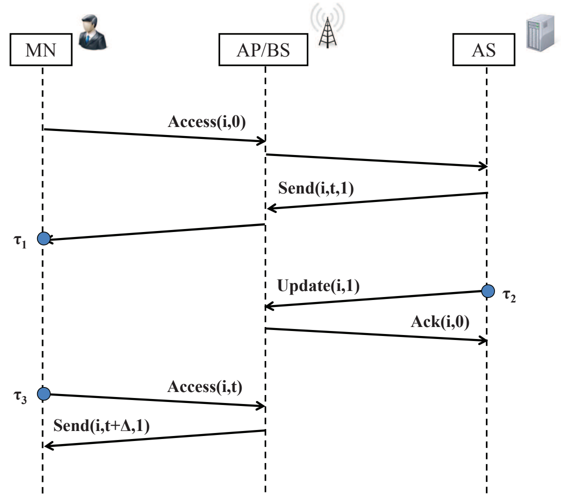


Fig. 4. Second-tier operation: invalidation and push mode.

any cached one, the AP/BS relays **Access(**i**,** 0**)** message to the AS and then the AS responds with **Send(**i**,** t**,** 1**)** message with the requested data object. After that, the AP/BS has a cached data object, which is managed by the second-tier operation.

At time τ_2, another request to O_i, **Access(**i**,** t**)** message, is issued by the MN. Since there is no modification to O_i after τ_1, the AP/BS confirms that the cached version is the most recent one by sending **Send(**i**,** t**,** 0**)** message and then the MN can use the cached data object. If the data object O_i is updated after τ_1, different operations are performed depending on the mode at the second-tier, which will be elaborated in Section IV-C.

C. Second-Tier Operation of FW-DAS

In wireless data access applications, data objects have different popularity and popular data objects can be proactively cached in the AP/BS to enhance the perceived performance [23]. For example, if a data object is popular, the data object can be invalidated and updated by proactively propagating the data object to the AP/BS. Otherwise, the AS simply invalidates without updating the data object at the AP/BS to save the transmission cost. In addition, some data objects may be popular only in a specific area. In such a case, although these data objects are not proactively pushed to the AP/BS, their updates can be explicitly requested by the AP/BS.

To consider these situations, we define three modes at the second-tier: 1) invalidation and push mode; 2) invalidation and pull mode; and 3) invalidation only mode. The invalidation and push mode will be used for globally popular data objects, which are proactively disseminated to the AP/BS when data objects are modified. On the other hand, the invalidation and pull mode is useful for locally popular data objects. Since they are not globally popular, they are not proactively delivered to the AP/BS. However, the AP/BS can request the updated versions if the data objects are popular in the AP/BS's service area. The last invalidation only mode is appropriate to unpopular data objects.

1) Invalidation and Push Mode: In this mode, if a data object is modified at the AS, the AS sends an invalidation message that contains the data object itself. As shown in Figure 4, the MN first accesses a data object O_i and the data object is cached by the MN and the AP/BS at time τ_1. When the data object O_i is updated at time τ_2, the AS checks the popularity of O_i. If the data object is regarded as popular by the AS, the AS sends **Update(**i**,** 1**)** message with the updated data object and thus the AP/BS can maintain the up-to-date data object. Consequently, the MN can check the validity of the data object through the AP/BS at time τ_3 and immediately access O_i from the AP/BS without further contacting the AS. Therefore fast data access for popular data objects is allowed.

2) Invalidation and Pull Mode: Figure 5 shows the message flow in the invalidation and pull mode. At time τ_1, the MN uses a data object O_i and the object can be cached by the MN as well as the AP/BS. When the data object is updated by the AS at time τ_2, the AS regards the data object as unpopular one and thus sends **Update(**i**,** 0**)** message only to invalidate the cached data object at the AP/BS. When the AP/BS receives **Update(**i**,** 0**)** message from the AS, the AP/BS checks whether the data object is locally popular in the AP/BS's service area. If the data object is considered as a locally popular data object, the AP/BS sends **Ack(**i**,** 1**)** message to the AS to retrieve the up-to-date data object and the AS then responds with **Send(**i**,** $t+\Delta$**,** 1**)** message containing the updated data object. Finally, the AP/BS can cache the updated data object and thus the MN can access O_i from the AP/BS at time τ_3 without contacting the AS.

3) Invalidation Only Mode: In the invalidation only mode, the AS sends an invalidation message without any updated data object because the data object is unpopular and is unlikely to be accessed in the near future. As shown in Figure 6, the MN uses a data object O_i and caches it at time τ_1. When O_i is updated at time τ_2, the AS sends **Update(**i**,** 0**)** message to invalidate the cached data object and the AP/BS responds with **Ack(**i**,** 0**)** message. Therefore, for a new access request

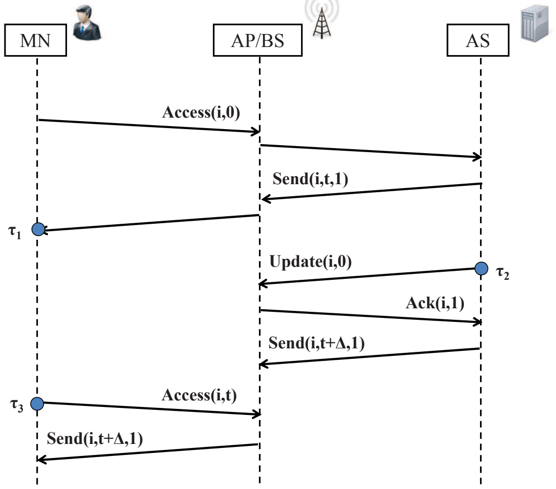


Fig. 5. Second-tier operation: invalidation and pull mode.

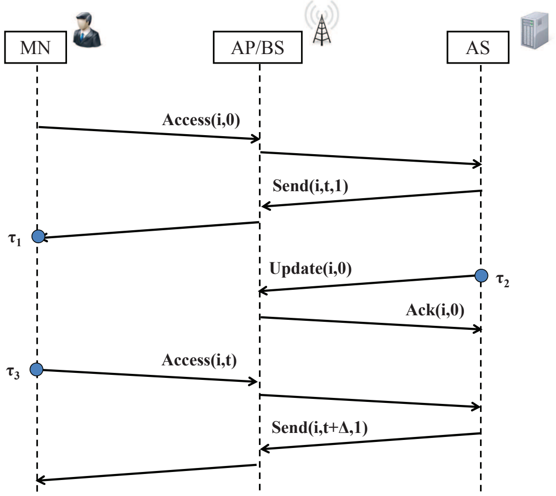


Fig. 6. Second-tier operation: invalidation only mode.

to O_i at time τ_3, the MN should contact to the AS by means of **Access(**i**,** t**)** message.

To attain the optimal performance, the operation mode in FW-DAS should be carefully determined by considering the access latency and traffic volume. Hence, how to choose the operation mode will be described in Section VI.

V. Performance Analysis

In this section, we derive an analytical model for the access latency of FW-DAS, which is defined as the latency for accessing a data object at the AS. In terms of the cache replacement policy, we assume the use of the least recently used (LRU) policy [24], [25], which chooses a data object that has not been used for the longest period of time when the cache replacement is needed. Table I summarizes key notations used in the analytical model.

TABLE I
Summary of notations.

Notation	Meaning
S_{access}	Size of **Access** message
S_{send}	Size of **Send** message without any data object
S_{data}	Size of **Send** message with data object
H	Number of hops in wired link
B	Wired link bandwidth
W	Wireless link bandwidth
M	Number of MNs in the area of AP/BS
α	Probability of the data object update event during the inter-data object access time of the tagged MN
β	Probability of the access event to the updated data object by MNs except the tagged MN during the inter-data object access time of the tagged MN
γ	Probability of the cache replacement at the tagged MN due to cache overflow
L_{push}	Access latency for the invalidation and push mode
L_{pull}	Access latency for the invalidation and pull mode
L_{only}	Access latency for the invalidation only mode
L_{total}	Total access latency
μ_i	Update rate of data object i
λ_i	Access rate of data object i

TABLE II
Latency Computation.

Message	Wired	Wireless
Access	$H \cdot S_{access}/B$	S_{access}/W
Send w/o data object	$H \cdot S_{send}/B$	S_{send}/W
Send w data object	$H \cdot S_{data}/B$	S_{data}/W

A. Access Latency

The access latency is the sum of delays in all procedures. Each procedure delay can be calculated by dividing the message size by the corresponding link bandwidth[3]. Let S_{access} denote the size of **Access** message. The sizes of **Send** message without any data object and with a data object are represented by S_{send} and S_{data}, respectively. The wireless link is one-hop link whereas the wired link is H hops link. In this analysis, WLAN is used as the wireless link; however, it can be extended to other types of wireless links. The wired and wireless links are considered to have symmetrical bandwidths of B and W, respectively. We do not consider any processing latency at the AP/BS and intermediate routers. The latency in each link can be calculated as shown in Table II. Let $access_l$, $send_l$, and $data_l$ denote the latency for **Access**, **Send** without a data object, and **Send** with a data object, respectively, where l is the link index, i.e., the indexes of the wireless and wired links are 1 and 2, respectively. For instance, when **Access** message is sent by the MN and the message is resolved by the network cache using the cached data object, the latency for this case is given by $access_1$ + $data_1$.

Let α be the probability that a data object update event occurs during the inter-data object access time of a tagged MN. Also, β represents the probability that other MNs except a tagged MN access the updated data object before the tagged MN. On the other hand, γ refers to the probability that a

[3] In this work, our main interest is to evaluate and compare the access latency in different wireless data access schemes. Thus, this high-level latency model without the considerations of detailed operations at low layers (e.g., MAC/physical layers) is sufficient for our purpose. In fact, most of the previous works [12], [14], [20], [26] are based on the high-level latency model.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2014.2314643, IEEE Transactions on Wireless Communications

cached data object at the tagged MN is replaced due to cache overflow. To derive the access latency in FW-DAS, we should consider the following four cases: 1) no data object update events occur between two access events and the cached data object at the MN is not replaced (probability of (1 - α) (1 - γ)); 2) no data object update events occur between two access events, but the cached data object at the MN is replaced owing to the cache overflow by other data objects (probability of (1 - α)γ); 3) a data object update event occurs and other MNs access the updated data object and cache it (probability of $\alpha\beta$); and 4) a data object update event occurs and no other MNs access the updated data object (probability of α(1 - β)).

In FW-DAS, an appropriate operation mode is applied to each data object depending on the data object popularity. Therefore, we need to define the access latency of a data object in each mode. L_{push}, L_{pull} and L_{only} represent the access latency for the invalidation and push mode, the invalidation and pull mode, and the invalidation only mode, respectively.

In the invalidation and push mode, the MN contacts the AP/BS for confirmation and receives an acknowledgement message from the AP/BS for case 1). Therefore, the corresponding access latency is given by $access_1$ + $send_1$. For cases 2), 3) and 4), the MN first contacts the AP/BS, and the cached data object at the AP/BS can be delivered to the MN. Then, the access latency can be expressed as $access_1$ + $data_1$. Consequently, the average access latency in the invalidation and push mode can be computed as

$$\begin{aligned}L_{push} = \ & (1-\alpha)(1-\gamma)(access_1 + send_1)\\ & + (1-\alpha)\gamma(access_1 + data_1)\\ & + \alpha\beta(access_1 + data_1)\\ & + \alpha(1-\beta)(access_1 + data_1). \qquad (1)\end{aligned}$$

The access latency in the invalidation and pull mode, L_{pull}, is the same as that in the invalidation and push mode. This is because data objects in the invalidation and pull mode can be cached at the AP/BS in advance as the same as the invalidation and push mode.

In the invalidation only mode, the MN contacts the AP/BS for confirmation and receives an acknowledgement message from the AP/BS for case 1). Therefore, the access latency is $access_1$ + $send_1$. For cases 2) and 3), the MN first contacts the AP/BS and the cached data object at the AP/BS can be delivered to the MN. Hence, the corresponding access latency is $access_1$ + $data_1$. For case 4), since both the MN and the AP/BS have no data object, the access from the AS is needed. Thus, the access latency can be computed as $access_1$ + $access_2$ + $data_1$ + $data_2$. Therefore, the average access latency in the invalidation only mode can be obtained as

$$\begin{aligned}L_{only} = \ & (1-\alpha)(1-\gamma)(access_1 + send_1)\\ & + (1-\alpha)\gamma(access_1 + data_1)\\ & + \alpha\beta(access_1 + data_1)\\ & + \alpha(1-\beta)(access_1 + access_2 + data_1 + data_2).\end{aligned} \qquad (2)$$

It is assumed that the AS has N data objects, and n_1, n_2, and n_3 data objects use the invalidation and push mode, the invalidation and pull mode, and the invalidation only mode, respectively (i.e., n_1 + n_2 + n_3 = N). Let x, y, and z be the portions of data objects using the invalidation and push mode, the invalidation and pull mode, and the invalidation only mode, respectively, i.e., $x = \frac{n_1}{N}$, $y = \frac{n_2}{N}$, and $z = \frac{n_3}{N}$. Then, the expected access latency L_{total} for FW-DAS can be expressed as

$$L_{total} = x \cdot L_{push} + y \cdot L_{pull} + z \cdot L_{only}. \qquad (3)$$

How to select x, y, and z will be elaborated in Section VI.

B. Derivation of α, β, γ

To compute L_{total}, α, β, and γ should be first determined [14]. To this end, we have the following assumptions.

1) The inter-data object update time t_u for O_i follows a general distribution with rate μ_i.
2) The inter-data object access time t_a for O_i by a tagged MN follows an exponential distribution with rate λ_i.
3) The inter-data object access time for O_i by MNs except a tagged MN follows an exponential distribution with rate (M - 1)λ_i, where M is the number of MNs in the area of the AP/BS.
4) The network cache size is sufficiently large, and thus cached data objects in the network cache are not replaced due to cache overflow. On the contrary, cache replacement can occur at the MN's cache.

To derive α and β, we use the timing diagram as shown in Figure 7. At τ_1 and τ_4, two data object update events occur and two data object access events by a tagged MN occur at τ_2 and τ_6. On the other hand, at τ_3 and τ_5, two data object accesses by MNs except the tagged MN are triggered. Let t_o refer to the inter-data object access time by other MNs except the tagged MN. In addition, t_u and t_a represent the inter-data object update time and the inter-data object access time by the tagged MN, respectively. Then, t_u = τ_4 - τ_1 has a general distribution with the probability density function $f(t_u)$ and mean 1/μ_i. Also, its Laplace transform is given by

$$f^*(s) = \int_{t_u=0}^{\infty} f(t_u)e^{-t_u s}dt_u.$$

If the access to O_i follows a Poisson process, τ_2 is a random observer of t_u. From the excess life theorem, t_r = τ_4 - τ_2 has a probability density function

$$r(t_r) = \mu_i \int_{t=t_r}^{\infty} f(t)dt$$

and its Laplace transform is given by

$$r^*(s) = \int_{t_r=0}^{\infty} r(t_r)e^{-t_r s}dt_r = (\frac{\mu_i}{s})[1 - f^*(s)]. \qquad (4)$$

Then, α is derived from

$$\begin{aligned}\alpha = \Pr(t_a > t_r) &= \int_{t_r=0}^{\infty} r(t_r) \int_{t_a=t_r}^{\infty} \lambda_i e^{-\lambda_i t_a} dt_a dt_r\\ &= \int_{t_r=0}^{\infty} r(t_r)e^{-\lambda_i t_r}dt_r = r^*(\lambda_i) = (\frac{\mu_i}{\lambda_i})[1 - f^*(\lambda_i)].\end{aligned} \qquad (5)$$

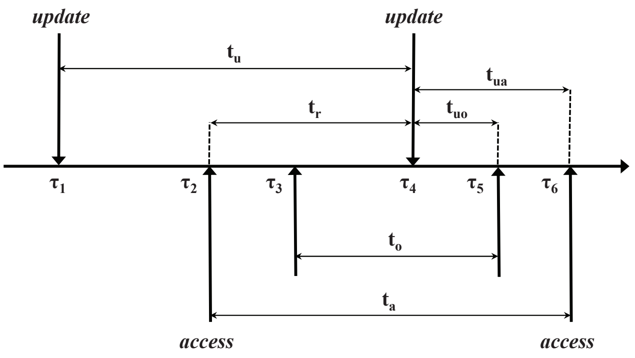


Fig. 7. Timing diagram.

If t_u is exponentially distributed, $f^*(\lambda_i)$ can be computed as

$$f^*(\lambda_i) = \frac{\mu_i}{\lambda_i + \mu_i}.$$

Then, we have

$$\alpha = \Pr(t_a > t_r) = \frac{1}{\rho_i + 1} \tag{6}$$

where $\rho_i = \lambda_i/\mu_i$ is the access-to-update ratio for O_i of the tagged MN.

As shown in Figure 7, let t_{ua} and t_{uo} be the time from the data object update event to the data object access event by the tagged MN and the time from the data object update event to the data object access event by MNs except the tagged MN, respectively. Then, the probability β that there exists an advance data object access event by other MNs except the tagged MN can be computed as

$$\begin{aligned}\beta &= \Pr(t_{ua} > t_{uo}) \\ &= \int_{t_{uo}=0}^{\infty} r(t_{uo}) \int_{t_{ua}=t_{uo}}^{\infty} r(t_{ua}) dt_{ua} dt_{uo}.\end{aligned} \tag{7}$$

If t_u is exponentially distributed, t_{ua} and t_{uo} are also exponentially distributed with rates λ_i and (M - 1)λ_i, respectively, by the random observer property. In this case, (7) can be expressed as

$$\begin{aligned}\beta = \Pr(t_{ua} > t_{uo}) &= \int_{t_{uo}=0}^{\infty} r(t_{uo}) e^{-\lambda_i t_{uo}} dt_{uo} \\ = \int_{t_{uo}=0}^{\infty} (M-1)\lambda_i e^{-M\lambda_i t_{uo}} dt_{uo} &= \frac{M-1}{M}.\end{aligned} \tag{8}$$

The probability that a cached data object in the MN is replaced due to cache overflow, γ, can be approximated as follows [14]. Let λ_i and λ_o be the access rates for a tagged data object O_i and other data objects except O_i, respectively. Then, $\lambda_o = \sum_{j \neq i} \lambda_j$. By the superposition property, the access process for other data objects except O_i follows a Poisson process with rate λ_o. Let $\theta(k)$ be the probability that there are k access events during an inter-data object access time for O_i. For tactical analysis, we consider the average inter-data object access time for O_i, $1/\lambda_i$. Then, $\theta(k)$ is given by

$$\theta(k) = \frac{e^{-\lambda_o/\lambda_i} (\lambda_o/\lambda_i)^k}{k!}. \tag{9}$$

A cached data object is replaced when there are more than K_{MN} access events for other data objects except O_i. Consequently, γ can be obtained from

$$\gamma = \sum_{k=K_{MN}}^{\infty} \theta(k). \tag{10}$$

VI. Operation Mode Selection Algorithm

As mentioned above, FW-DAS defines three operation modes: the invalidation and push, the invalidation and pull, and the invalidation only modes. The operation modes are determined depending on the popularity of data objects. As discussed in Section V-A, the parameter x is the proportion of data objects using the invalidation and push mode, which is given by $\frac{n_1}{N}$. That is, the most popular n_1 data objects in the AS use the invalidation and push mode. On the other hand, y represents the portion of data objects selecting the invalidation and pull mode, which is given by $\frac{n_2}{N}$. In FW-DAS, the most popular n_2 data objects in the AP/BS, except n_1 data objects using the invalidation and push mode, utilize the invalidation and pull mode. The rest of data objects follow the invalidation only mode. For instance, assume that x, y, and N are 0.1 0.05, and 100, respectively. Then, the most popular 10 data objects in the AS are invalidated and pushed to the AP/BS when they are modified. On the contrary, the most popular 5 data objects in the AP/BS, except data objects using the invalidation and push mode, use the invalidation and pull mode while remaining 85 data objects follow the invalidation only mode.

In FW-DAS, the access latency can be significantly reduced as the values of x and y increase because more data objects can be accessed from the AP/BS. However, when larger values of x and y are used, more update traffic can be generated. Consequently, x and y should be carefully determined not to incur significant update traffic while reducing the access latency substantially. To this end, we propose an operation mode selection algorithm that considers both the access latency and the update traffic. The expected saving cost and the increased cost are first derived by the analytical model when a data object uses the invalidation and push/pull modes instead of the invalidation only mode. After that, only when the expected saving cost is sufficiently larger than the increased cost, the data object uses the invalidation and push/pull modes. For the operation mode selection, we introduce another parameter ω that is defined as $x + y$. After determining ω, x and y are accordingly selected.

Without loss of generality, it is assumed that the data object index i is assigned in a descending order depending on the global popularity at the AS, i.e., the most popular data object in the AS is O_1.

Let SC_i be the expected saving cost of the ith data object using the invalidation and push/pull modes against using the invalidation only mode. When the data object update event for the ith data object occurs and no other MNs access the updated data object (with the probability of $\alpha(1-\beta)$), the invalidation and push/pull modes can save the transmission cost from the AS to the AP/BS since the updated data object can be found at the AP/BS. Specifically, the access latencies of the invalidation and push/pull modes and the invalidation only mode are

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2014.2314643, IEEE Transactions on Wireless Communications

$access_1 + data_1$ and $access_1 + access_2 + data_1 + data_2$, respectively. Therefore, the reduced access latency is given by $(access_1 + access_2 + data_1 + data_2) - (access_1 + data_1) = access_2 + data_2$. Consequently, SC_i can be computed as

$$SC_i = \lambda_i \alpha(1 - \beta)(access_2 + data_2) \qquad (11)$$

where λ_i is the access rate to the ith data object. From (6) and (8), (11) is rewritten as

$$SC_i = \frac{\lambda_i}{(\rho_i + 1)M}(access_2 + data_2). \qquad (12)$$

Compared with the invalidation only mode, the invalidation and push/pull modes can increase the update cost when the data object update occurs but no other MNs access the updated data object (with the probability of α(1 - β)). In particular, the invalidation and pull mode has higher update cost than the invalidation and push mode since an additional message should be transmitted in the invalidation and pull mode (see Figures 4 and 5). Therefore, we consider the increased update cost of the invalidation and push mode in choosing ω to select more data objects for the invalidation and push/pull modes. After choosing ω for the invalidation and push/pull modes, we classify the selected data objects into 1) the invalidation and push mode or 2) the invalidation and pull mode by considering the data object popularity.

Let IC_i be the increased update cost of the ith data object when the invalidation and push mode is used rather than the invalidation only mode. $update_2$ and ack_2 denote the latencies of wired link for **Update** without a data object and **Ack**, respectively. Since the update latencies of the invalidation only mode and the invalidation and push mode are respectively given by $update_2 + ack_2$ and $data_2 + ack_2$, IC_i can be defined as

$$IC_i = \mu_i \alpha(1 - \beta)(data_2 - update_2) \qquad (13)$$

where μ_i is the update rate of the ith data object. Then, (13) can be rewritten as

$$IC_i = \frac{\mu_i}{(\rho_i + 1)M}(data_2 - update_2). \qquad (14)$$

Note that the update cost is increased only at the wired link even though the invalidation and push/pull modes are adopted. This is because no data object is transmitted to the MN over the wireless link in FW-DAS even when the data object is updated.

After defining SC_i and IC_i, data objects satisfying $SC_i \geq \delta IC_i$ can be selected for the invalidation and push/pull modes, where δ is a tunable factor ($0 \leq \delta \leq 1$) and its default value is 0.1 in simulations. A smaller value of δ represents that fast access is more important than the reduction of the update cost. On the contrary, if it is more critical to mitigate the update cost, a larger value of δ can be set. Note that δ = 0 represents that FW-DAS does not concern the increased update cost and only focuses on reducing the access latency. Once δ is given, ω can be selected by dividing the largest index among data objects that meet $SC_i \geq \delta IC_i$ by N.

Figure 8 shows the chosen ω under different values of δ. It is assumed that the relative frequency for data objects follows a Zipf-like distribution [27], in which the relative probability of a request to the ith most popular data object ($1 \leq i \leq N$) is proportional to 1/i^κ. κ ($0 \leq \kappa \leq 1$) determines the skewness in the Zipf-like distribution. For instance, for $\kappa = 1$, the access probability of a data object is strictly proportional to its popularity ranking. On the other hand, when κ is 0, the access probabilities for all data objects are the same. Then, p_i is given by

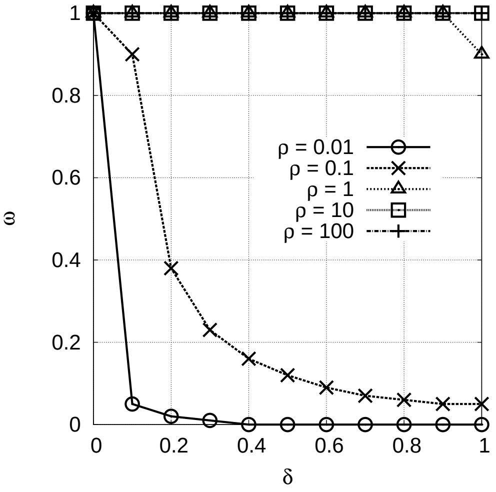


Fig. 8. Effects of tunable factor δ ($\kappa = 0.8$).

$$p_i = \frac{\Omega}{i^\kappa} \qquad (15)$$

where $\Omega = (\sum_{i=1}^{N} \frac{1}{i^\kappa})^{-1}$. It is shown that ω is almost constant when ρ is high, i.e., $\rho = 10$ and $\rho = 100$. When ρ is high, λ_i dominates μ_i and thus SC_i is much larger than IC_i. Therefore, most data objects can use the invalidation and push mode or the invalidation and pull mode without concerning the increased update cost when ρ is high. On the contrary, ω decreases significantly with the increase of δ when ρ is low, i.e., $\rho = 0.1$ and $\rho = 0.01$. When ρ is low, updates occur more frequently and thus higher update cost is expected. Consequently, the impact of δ is more significant under this situation.

When ω is given, x and y can be determined as follows. Apparently, if the data object popularities in the AP/BS (i.e., local popularity) and the AS (i.e., global popularity) are perfectly matched, all data objects of ωN can use the invalidation and push mode by sending the data objects to the AP/BS without any requests. On the contrary, if there is disparity between the local and global popularities, some data objects among ωN data objects should use the invalidation and pull mode by allowing additional requests from the AP/BS. To take the local popularity in the AP/BS into account, the popularity skewness parameter κ can be used based on the following observation. If κ is close to 1, only a few data objects are frequently accessed and therefore the popularities of data objects in the AP/BS and the AS are indistinguishable. In such a case, the consideration of local popularity in the invalidation and pull mode has limited impact. Consequently, most data objects selected by ω can use the invalidation and push mode. On the other hand, when κ is close to 0, each data object has similar popularity and the global popularity (at the AS) may be quite different from the local popularity (at the AP/BS). Hence, the invalidation and pull mode should be more effectively used when κ is low. Based on this rationale, x and y are determined as $x = \omega\kappa$ and $y = \omega(1-\kappa)$, respectively. For instance, if the chosen ω and κ are 0.5 and 0.8, respectively, x and y are set to 0.4 and 0.1, respectively.

Algorithm 1 shows the operation mode selection algorithm. First of all, ω is determined to strike a balance between the increased update traffic and the reduced access latency. Specifically, ω is obtained by dividing the largest index among data objects that meet $SC_i \geq \delta IC_i$ by N (lines 1-2). After that, x and y are determined based on the popularity skewness parameter κ (lines 3-4). Since xN data objects are the most popular ones, they are selected for the invalidation and push mode (lines 5-9). On the other hand, remaining data objects are sorted by the local popularity in the AP/BS (line 10). Then, yN data objects and $(1-x-y)N$ data objects are chosen for the invalidation and pull mode and the invalidation only mode, respectively (lines 11-17). Note that the proposed algorithm has low computational complexity of $O(N)$.

Algorithm 1 Operation mode selection algorithm.

```
1: Assign data object index $i$ to data objects in a descending
   order depending on the global popularity in the AS.
2: $\omega = \frac{\max\{i \,:\, SC_i \geq \delta IC_i\}}{N}$;
3: $x \leftarrow \omega\kappa$;
4: $y \leftarrow \omega(1-\kappa)$;
5: for each data object $i$ do
6:     if $\frac{i}{N} \leq x$ then
7:         Use the invalidation and push mode for data object
           $i$.
8:     end if
9: end for
10: Assign data object index $j$ to unselected data objects in a
    descending order depending on the local popularity in the
    AP/BS.
11: for each data object $j$ do
12:     if $\frac{j}{N} \leq y$ then
13:         Use the invalidation and pull mode for data object $j$.
14:     else
15:         Use the invalidation only mode for data object $j$.
16:     end if
17: end for
```

VII. Simulation Results

In this section, we present the simulation model and present comprehensive simulation results. In simulations, we run the schemes in a discrete event-driven simulator using C++. We compare FW-DAS with PER/CB [12] and PER-PER/CB-CB, where $X - Y$ represents that schemes X and Y (X, $Y \in$ {PER,CB}) are used in the first and second tiers, respectively, with the network cache at the AP/BS. The use of reliable transport and/or data link layer protocols (e.g., transmission control protocol (TCP) and radio link protocol (RLP)) is assumed. Since wireless data access schemes operate at the application layer, the WWAN and WLAN links can be characterized as having negligible packet losses.

TABLE III
Default parameter values for simulations.

S_{access}	S_{update}	S_{ack}	S_{send}	S_{data}	N
60 bytes	60 bytes	60 bytes	60 bytes	180 bytes	100
B	W	H	K_{MN}	K_{NC}	κ
100 Mbps	54 Mbps	10	20	60	0.8

TABLE IV
Total access latency (unit: μsec): simulation versus analytical results ($\nu = 1/{\mu_i}^2$).

ρ	0.01	0.1	1	10	100
Simulation	127.88	118.55	82.10	44.04	36.35
Analysis	127.20	119.70	81.83	43.97	36.47
Error (%)	0.53	0.96	0.32	0.17	0.34

In simulations, the inter-data object access time for O_i follows an exponential distribution with rate λ_i, which is given by $p_i\lambda$ where λ is the net access rate to data objects and p_i is the probability of accessing O_i. The inter-data object update time is drawn from a Gamma distribution with mean $1/\mu_i$ and variance ν. We are particularly interested in the Gamma distribution because the distribution of any positive random variable can be approximated by a mixture of Gamma distributions. For example, when $\nu = 1/{\mu_i}^2$, the Gamma distribution becomes an exponential distribution. Then, the aggregate access-to-update ratio ρ for data objects can be defined as

$$\rho = \frac{\sum_{i=1}^{N} \lambda_i}{N\mu_i} = \frac{\lambda}{N\mu_i}. \tag{16}$$

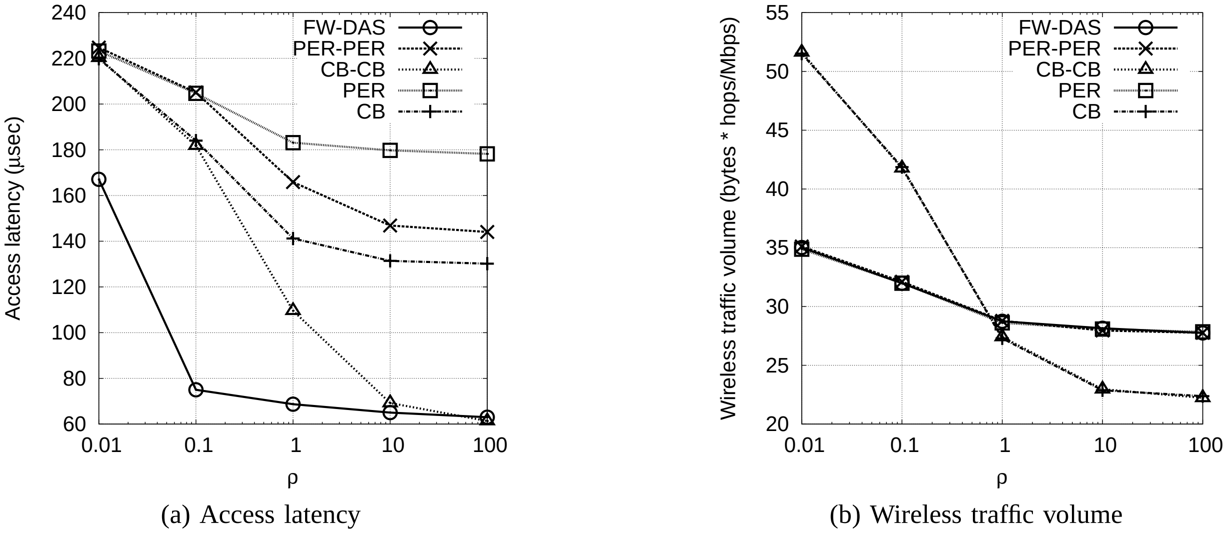
The default parameter values for simulations are derived from [12] and summarized in Table III. S_{update} and S_{ack} represent the sizes of **Update** message and **Ack** message without any data object, respectively. Note that the size of **Update** message with a data object is the same as S_{data}.

From Table IV, it can be seen that the analytical results are consistent with the simulation results and the errors in the analytical results are less than 1% in all cases. Therefore, the accuracy of the developed analytical model is verified. For forthcoming evaluation results, only simulation results are plotted to improve the clarity.

A. Effects of ρ

Figure 9(a) illustrates the effect of the access-to-update ratio ρ on the access latency. It can be seen that the access latency decreases with the increase of ρ. This can be explained as follows. When ρ is low, the update rate dominates the access rate and the possibility that a cached data object becomes stale is high. Thus, longer access latency is observed because the AS should be contacted.

From Figure 9(a), it can be found that the network cache at the AP/BS is effective, i.e., PER-PER and CB-CB are better than PER and CB, respectively. It can be also seen that CB (CB-CB) can reduce the access latency compared with PER (PER-PER). This is because the MN in CB does not need



(a) Access latency

(b) Wireless traffic volume

Fig. 9. Effects of access-to-update ratio ρ.

to contact the AP/BS or the AS when there is a cached data object. In addition, FW-DAS has shorter access latency than other schemes even compared with CB-CB.

On the other hand, CB (CB-CB) may incur more traffic over wireless link due to the invalidation procedure when ρ is low. Thus, we measure the wireless traffic volume, which is the amount of messages divided by the wireless link bandwidth used in each data access scheme, and its unit is bytes * hops/Mbps. As shown in Figure 9(b), if ρ is low, PER (PER-PER) has lower wireless traffic volume than CB (CB-CB). FW-DAS does not incur frequent transmissions of invalidation messages over the wireless link even when ρ is equal to or lower than 1. This is because FW-DAS operates as PER at the first-tier. Therefore, it is shown that the wireless traffic volume of FW-DAS is the same as that of PER. To conclude, FW-DAS can reduce the access latency significantly while maintaining the wireless traffic volume comparable to PER.

When ρ is high, CB (CB-CB) is superior to PER (PER-PER) in terms of the access latency and the wireless traffic volume. Moreover, it can be found that CB-CB has comparable access latency to FW-DAS and lower wireless traffic volume than FW-DAS. This is because CB-CB with the network cache can minimize the wireless traffic volume when the data objects are rarely updated (e.g., ρ is 100). However, as mentioned in Section I, FW-DAS is designed for applications where data objects are frequently updated (i.e., ρ is low) and fast access is needed. Consequently, it can be concluded that FW-DAS is better than CB-CB in terms of both the access latency and the wireless traffic volume for our target environments.

B. Effects of κ

Data access patterns in wireless data access applications are quite diverse. Therefore we analyze the effect of κ that determines the skewness in the data access pattern. As κ approaches 1, only a few data objects occupy a large portion in data access. On the other hand, κ of 0 represents that the access frequency for each data object is identical. Obviously, the performance of data access schemes can be improved when a larger value of κ is used. From Figure 10, it can be found that the performance of FW-DAS can be more apparently improved with the increase of κ. Specifically, when κ = 0.2, FW-DAS can reduce the access latency of PER-PER/CB-CB and PER/CB by 49.3%/46.8% and 49.4%/46.7%, respectively. On the contrary, if κ = 0.8, FW-DAS can save the access latency of PER-PER/CB-CB and PER/CB by 64.2%/59.8% and 64.2%/59.9%, respectively. This is because more popular data objects can be cached or proactively disseminated to the AP/BS in FW-DAS when κ is large.

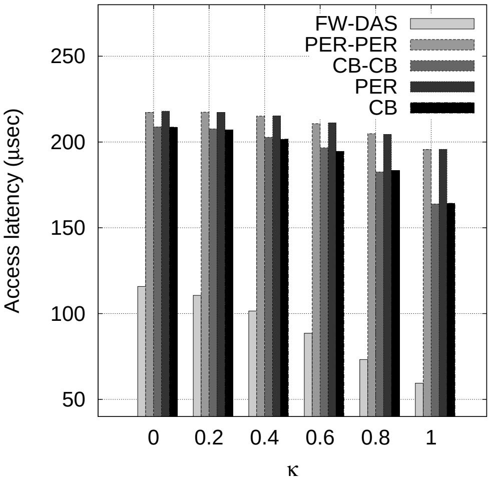


Fig. 10. Effects of popularity skewness parameter κ ($\rho = 0.1$).

C. Effects of K_{MN} and K_{NC}

As shown in Figure 11, the access latency of FW-DAS can be significantly reduced by increasing K_{NC} except ρ = 0.01. For example, if K_{NC} increases from 30 to 90 when ρ = 10, FW-DAS can save the access latency by 68.6%. On the contrary, if K_{NC} increases from 30 to 90 and ρ = 0.01, the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2014.2314643, IEEE Transactions on Wireless Communications

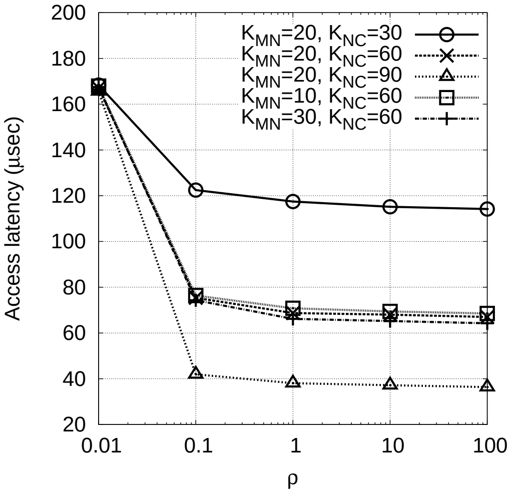


Fig. 11. Effects of cache size K_{MN} and K_{NC}.

access latency does not change significantly. This is because the possibility that a cached data object becomes stale is high and lower ω is chosen to mitigate the excessive update traffic when ρ = 0.01.

It also shows that the effect of K_{MN} on the access latency is not apparent. In FW-DAS, some cache misses at the MN can be resolved by the network cache and therefore the access latency is more sensitive to K_{NC} than K_{MN}. This observation is promising because the network cache has little constraint in the cache size compared with the MN, and therefore a larger K_{NC} can be deployed to improve the performance of FW-DAS.

D. Effects of S_{data}

Since the latency is dependent on the data object size S_{data}, we investigate the performance of FW-DAS over a wide range of data object sizes [14]. Figure 12(a) demonstrates that the gain of FW-DAS is significant regardless of S_{data}. When S_{data} is 180 bytes, FW-DAS can reduce the access latency of PER-PER/CB-CB and PER/CB by 62.7%/58.0% and 62.6%/58.4%, respectively. On the other hand, when S_{data} is 787 bytes, FW-DAS can reduce the access latency of PER-PER/CB-CB and PER/CB by 48.4%/46.7% and 48.2%/46.8%, respectively. The reason why small S_{data} is more effective can be explained as follows. When the data object size is small, the update cost is also low. Therefore, a larger value of ω is chosen (see Figure 12(b)) and more data objects can be proactively pushed to the AP/BS. Consequently, lower access latency can be achieved when the data object size is small. In short, FW-DAS can be applied to applications with various sizes of data object and more effective with small data objects.

E. Effects of Wireless Bandwidth

Table V gives the performance gain of FW-DAS in terms of access latency when three wireless access technologies with different bandwidths are used. The performance gain represents how much access latency can be reduced by using FW-DAS against the scheme Z where $Z \in \{$PER-PER, CB-CB, PER, CB$\}$. For example, FW-DAS can reduce the access latency compared with PER-PER by 37.4% when HSUPA is employed. As mentioned before, the default wireless bandwidth in simulations is 54Mbps, which is based on IEEE 802.11a/g WLANs. On the other hand, the downlink and uplink bandwidths of LTE-Advanced are set to 300Mbps and 75Mbps, respectively [28]. Also, the downlink and uplink bandwidths of HSUPA are set as 14.4Mbps and 5.76Mbps, respectively [29]. From Table V, it can be found the gain of FW-DAS is more apparent when LTE is assumed. On the other hand, the performance gain of FW-DAS with HSUPA is minimal. This is because the access latency is directly affected by the wireless bandwidth. This result is promising because the wireless bandwidth is continuously increasing with the advance of wireless communications technologies (e.g., multiple input and multiple output (MIMO), cooperative communications, and so on).

TABLE V
EFFECTS OF WIRELESS BANDWIDTH (UNIT: %, $\rho = 0.1$).

Network type	PER-PER	CB-CB	PER	CB
HSUPA	37.4	28.6	37.3	29.2
WLAN	62.8	58.0	62.7	58.5
LTE-Advanced	70.2	66.2	70.1	66.7

VIII. CONCLUSION

In this paper, we have proposed a fast wireless data access scheme (FW-DAS) for wireless data access applications where fast access is required with low wireless traffic load. We have developed an analytical model for the access latency, and investigated how to choose the appropriate operation mode in FW-DAS for better performance. From analytical and simulation results, it is demonstrated that FW-DAS can reduce the access latency significantly while maintaining the wireless traffic volume comparable to PER. For our future work, we will try to improve FW-DAS performance by means of node cooperation.

REFERENCES

[1] G. Lee, I. Jang, and S. Pack, "Fast wireless data access scheme in wireless networks," in *Proc. ICNC*, January 2013.

[2] E. Halepovic, C. Williamson, and M. Ghaderi, "Wireless data traffic: A decade of change," *IEEE Network*, vol. 23, no. 2, pp. 20-26, March 2009.

[3] Cisco Systems Inc., "Cisco visual networking index: Global mobile data traffic forecast update, 2012-2017," February 2013.

[4] B. Hayes, "Cloud computing," *Communications of the ACM*, vol. 51, no. 7, pp. 9-11, July 2008.

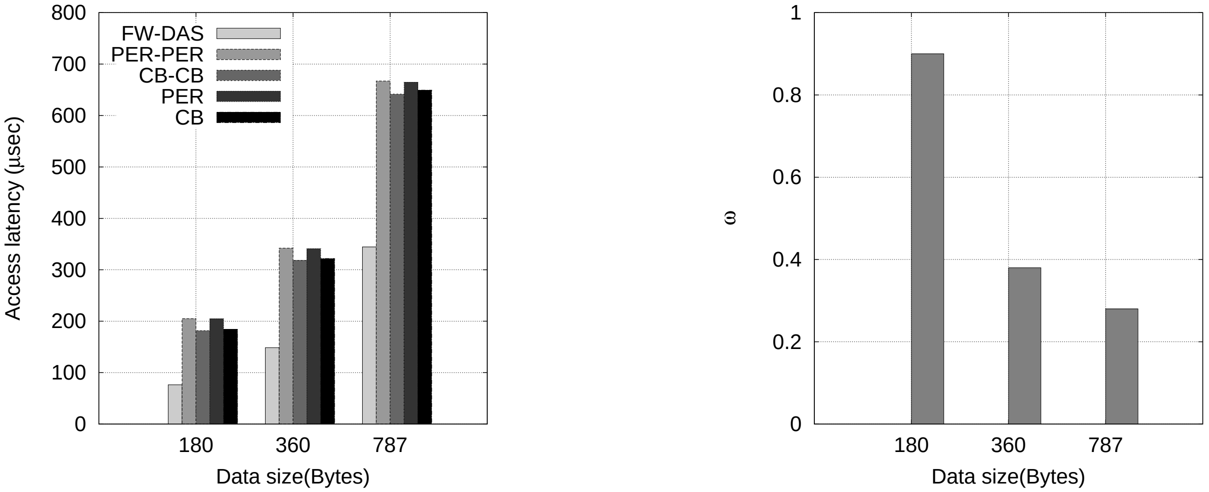
[5] S. Azodolmolky, P. Wieder, and R. Yahyapour, "Cloud computing networking: Challenges and opportunities for innovations," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 54-62, July 2013.

[6] M. Miller, *Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online*. Que, 2008.

[7] Y. Fang and Y. Lin, "Strongly consistent access algorithms for wireless data networks," *ACM Wireless Networks*, vol. 11, no. 3. pp. 243-254, May 2005.

[8] Z. Wei, G. Pierre, and C. Chi, "CloudTPS: Scalable transactions for web applications in the cloud," *IEEE Transactions on Services Computing*, vol. 5, no. 4, pp. 525-539, 4th qtr. 2012.

[9] G. Chockler, G. Laden, and Y. Vigfusson, "Design and implementation of caching services in the cloud," *IBM Journal of Research and Development*, vol. 55, no. 6, pp. 422-432, November 2011.



(a) Access latency

(b) ω

Fig. 12. Effects of data object size S_{data} ($\rho = 0.1$).

[10] M. Bjorkqvist, L. Y. Chen, M. Vukolic, and X. Zhang, "Minimizing retrieval latency for content cloud," in *Proc. IEEE INFOCOM*, April 2011.

[11] J. Dai, Z. Hu, B. Li, J. Liu, and B. Li, "Collaborative hierarchical caching with dynamic request routing for massive content distribution," in *Proc. IEEE INFOCOM*, March 2012.

[12] Y. Lin, W. Lai, and J. Chen, "Effects of cache mechanism on wireless data access," *IEEE Transactions on Wireless Communications*, vol. 2, no. 6, pp. 1247-1258, November 2003.

[13] H. Chen and Y. Xiao, "Cache access and replacement for future wireless internet," *IEEE Communications Magazine*, vol. 44, no. 5, pp. 113-123, May 2006.

[14] S. Pack, H. Rutagemwa, X. Shen, J. W. Mark, and K. Park, "Proxy-based wireless data access algorithms in mobile hotspots," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 5, pp. 3165-3177, September 2008.

[15] H. Che, Y. Tung, and Z. Wang, "Hierarchical web caching systems: Modeling, design and experimental results," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 7, pp. 1305-1314, September 2002.

[16] M. Akon, M. T. Islam, X. Shen, and A. Singh, "A bandwidth and effective hit optimal cache scheme for wireless data access networks with client injected updates," *Elsevier Computer Networks*, vol. 56, no. 7, pp. 2080-2095, May 2012.

[17] Y. Li and I. Chen, "Adaptive per-user per-object cache consistency management for mobile data access in wireless mesh networks," *Elsevier Journal of Parallel and Distributed Computing*, vol. 71, no. 7, pp. 1034-1046, July 2011.

[18] Y. Xiao and H. Chen, "Optimal callback with two-level adaptation for wireless data access," *IEEE Transactions on Mobile Computing*, vol. 5, no. 8, pp. 1087-1102, August 2006.

[19] H. Chen, Y. Xiao, and X. Shen, "Update-based cache access and replacement in wireless data access," *IEEE Transactions on Mobile Computing*, vol. 5, no. 12, pp. 1734-1748, December 2006.

[20] K. Lee, I. Jang, S. Pack, and W. Lee, "Design and analysis of cooperative wireless data access algorithms in multi-radio wireless networks," *ACM Wireless Networks*, vol. 19, no. 1, pp. 17-29, January 2013.

[21] C. Chuang, Y. Lin, and Y. Yeh, "Performance of linear-type mobile data transmission," *IEEE Transactions on Wireless Communications*, vol. 10, no. 8, pp. 2451-2455, August 2011.

[22] H. Ko, S. Pack, and W. Lee, "Timer-based push scheme for online social networking services in wireless networks," *IEEE Communications Letters*, vol. 16, no. 12, pp. 2095-2098, December 2012.

[23] J. Zander and P. Mahonen, "Riding the data tsunami in the cloud: Myths and challenges in future wireless access." *IEEE Communications Magazine*, vol. 51, no. 3, pp. 145-151, March 2013.

[24] J. Wang, "A survey of web caching schemes for the internet," *ACM SIGCOMM Computer Communication Review*, vol. 29, no. 5, pp. 36-46, October 1999.

[25] A. Balamash and M. Krunz, "An overview of web caching replacement algorithms," *IEEE Communications Surveys and Tutorials*, vol. 6, no. 2, pp. 44-56, 2nd qtr. 2004.

[26] K. Fawaz and H. Artail, "DCIM: Distributed cache invalidation method for maintaining cache consistency in wireless mobile networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 4, pp. 680-693, April 2013.

[27] L. Berslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proc. IEEE INFOCOM*, March 1999.

[28] A. Ghosh, R. Ratasuk, B. Mondal, N. Mangalvedhe, and T. Thomas, "LTE-advanced: Next-generation wireless broadband technology," *IEEE Wireless Communications*, vol. 17, no. 3, pp. 10-22, June 2010.

[29] M. A. Khairy and M. A. El-Saidny, "Uplink quality measurement reporting mechanism for HSUPA," *IEEE Communications Magazine*, vol. 46, no. 6, pp. 68-75, June 2008.

Giwon Lee received the B.S. and M.S. degree from Korea University, Seoul, Korea, in 2009 and 2011, respectively. He is currently an Ph.D. course student in School of Electrical Engineering, Korea University, Seoul, Korea. His research interests include mobile cloud computing, software defined networking, vehicular networks, and future internet.

Insun Jang received the B.S. degree from Korea University, Seoul, Korea, in 2011. He is currently an M.S. and Ph.D. integrated course student in School of Electrical Engineering, Korea University, Seoul, Korea. His research interests include content delivery networks, multimedia networking, software defined networking, and future internet.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2014.2314643, IEEE Transactions on Wireless Communications

Sangheon Pack received the B.S. and Ph.D. degrees from Seoul National University, Seoul, Korea, in 2000 and 2005, respectively, both in computer engineering. In 2007, he joined the faculty of Korea University, Seoul, Korea, where he is currently an Associate Professor in the School of Electrical Engineering. From 2005 to 2006, he was a Postdoctoral Fellow with the Broadband Communications Research Group, University of Waterloo, Waterloo, ON, Canada. He was the recipient of KICS (Korean Institute of Communications and Information Sciences) Haedong Young Scholar Award 2013, IEEE ComSoc APB Outstanding Young Researcher Award in 2009, LG Yonam Foundation Overseas Research Professor Program in 2012, and Student Travel Grant Award at the IFIP Personal Wireless Conference (PWC) 2003. From 2002 to 2005, he was a recipient of the Korea Foundation for Advanced Studies Computer Science and Information Technology Scholarship. He was a publication co-chair of IEEE INFOCOM 2014, a co-chair of IEEE VTC 2010-Fall transportation track, a co-chair of IEEE WCSP 2013 wireless networking symposium, a TPC vice-chair of ICOIN 2013, and a publicity co-chair of IEEE SECON 2012. He is an editor of Journal of Communications Networks (JCN) and a senior member of the IEEE. His research interests include Future Internet, SDN/ICN/DTN, mobility management, mobile cloud networking, multimedia networking, and vehicular networks.

Xuemin (Sherman) Shen received the B.Sc.(1982) degree from Dalian Maritime University (China) and the M.Sc. (1987) and Ph.D. degrees (1990) from Rutgers University, New Jersey (USA), all in electrical engineering. He is a Professor and University Research Chair, Department of Electrical and Computer Engineering, University of Waterloo, Canada. Dr. Shen's research focuses on resource management in interconnected wireless/wired networks, wireless network security, wireless body area networks, vehicular ad hoc and sensor networks. Dr. Shen serves/served as the Editor-in-Chief for IEEE Network, Peer-to-Peer Networking and Application, and IET Communications; a Founding Area Editor for IEEE Transactions on Wireless Communications; an Associate Editor for IEEE Transactions on Vehicular Technology, Computer Networks, and ACM/Wireless Networks, etc.; and the Guest Editor for IEEE JSAC, IEEE Wireless Communications, IEEE Communications Magazine, and ACM Mobile Networks and Applications, etc. Dr. Shen is a registered Professional Engineer of Ontario, Canada, an IEEE Fellow, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, and a Distinguished Lecturer of IEEE Vehicular Technology Society and Communications Society.