

AIM - Adaptive Invalidation Mechanism for Wireless Networks

Mangal Sain*, Srikanth Varanasi**, Young-Jin Kang*, Hoon Jae Lee*

* Department of Computer and Information Engineering, Dongseo University, Busan, South Korea

**Computer science, Texas Tech University, Lubbock, Texas, USA

mangalsain1@gmail.com, srikanth.varanasi@ttu.edu, rkddudwls55@gmail.com, hjlee@dongseo.ac.kr

Abstract— With the advent and advancement of mobile technology, there has been a significant increase in accessing the Internet services and information wirelessly. One of key optimization techniques is to cache frequently accessed data items in a local cache. In Invalidation report (IR) based cache invalidation mobile nodes access the cached data items to answer a query only after receiving an IR that is periodically broadcasted by a server. This may unnecessarily increase the query delay for waiting towards next IR. To provide broadcast support to nodes distributed over different wireless cells, IR incurs communication overhead.

In this paper, we propose a data access scheme with low cache consistency scheme based on the poll-each-request (PER) cache invalidation framework, called Poll Each Request-Weak Consistency (PER-WC). Unlike prior approach, where every node has the same consistency level with the server, each node is able to set its own target consistency level independently. We designed a customized simulation framework to conduct our experiments using the CSIM, which is a popular development toolkit for discrete-driven simulation and modeling. We vary key simulation parameters to measure the performance. Our results show that the proposed scheme can reduce the query delay.

Keywords— IR, Cache Consistency, Query Delay, Cache Invalidation, Data access scheme

I. INTRODUCTION

This is a Digital Era with 7.2 Billion smart gadgets and 2 Billion smart phones [1] exceeding human population. It just took 3 decades for such exponential development. As of 2015, the rate of growth was estimated to be 5 times as compared to the rate of increase in world population which accounts to 7 Billion [2] (2015).

Wireless Networks are classified into 2 types namely single-hop networks and multi-hop networks. The transmission nature is of 3 types namely – Broadcast, Unicast and Multicast. Caching is an important feature in today's applications. To cache efficiently, a node has to decide upon the following 3 parameters such as i) cache admission policy ii) cache invalidation policy iii) cache replacement policy. Our focus is on the cache invalidation policy.

A. IR/UIR Approach

For a caching scheme, consistency of data items is very essential. Strong consistency denotes that the data items in client cache are always valid with respect to server. Such high consistency does not required all the time with the time-insensitive data. Here, server maintains a table of mapping

between clients and their requested data items. When such a mapping is provided, server tracks updates to data items and proactively sends to registered clients. One such mechanism which efficiently utilizes cache to answer user queries is Invalidation Report (IR). In IR approach, the servers broadcast the data periodically containing data items and their most recent timestamps in key-value pair structure. The IR also contains the update history information witnessed during the last X intervals. In Updated Invalidation Report (UIR) mechanism, the data items that are updated between two consecutive IRs are only broadcasted. Compared to IR, the UIR does not contain the update history. The server inserts several UIRs into each IR interval. Due to the periodic broadcast nature of IR/UIR, the nodes can operate in sleep state and save energy. In IR/UIR server is stateless, strong consistent client-initiated scheme.

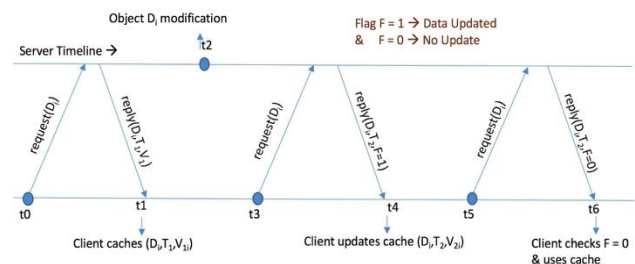


Figure 1. Poll Each Request

B. PER Approach

To overcome the limitations of broadcasting in IR/UIR, we adopted other strong consistent protocols such as Poll-Each-Read (PER) that are widely researched and adapted. PER is a stateless, client-initiated strong consistent scheme. In PER, clients request data as and when required. There is no server broadcast as in IR. When a client queries for a data object, the client polls server to check the validity of its cached data item. If the data is up-to-date, the server sends a confirmation message without any data object. Otherwise, the server sends a reply with the updated data item.

C. Motivation

IR works effectively for a single cell environment. However, as the number of cells within single Base Station (BS) increase, the effective numbers of IRs/cell decrease. This

causes the nodes to receive inappropriate data update notifications for long periods of time. Also, every node in IR waits for a minimum of half IR interval in average to answer a query of cached data items. This incurs a long query delay and less effective ratio if cells increase. Therefore, we propose a weak consistency approach based on PER mechanism. Our contribution is as follows:

- Developed PER-WC, a user defined consistency sensitive cache Invalidation strategy.
- Each client independently and flexibly decides on the consistency. Based on this condition, a validity window is derived which specifies the total number of cached items that can be answered without an uplink request.
- As a consequence of having a validity window, a client can request multiple data items within a single request.

II. RELATED WORKS

In this section, we briefly review various research areas related to data access with an emphasis on the efforts related to the work presented in this paper. Most of the cache invalidation strategies proposed and developed fall under the following three categories. They are (1) Co-operative based models (2) Adaptive Consistent models (3) Replacement models.

In this section, we give an introduction and an insight into various approaches and then we specifically focus on three key areas: (1) Consistency level (Strong or Weak) (2) Consistency control (Push, pull and hybrid) and (3) Cache status maintenance (stateful or stateless).

A. Strong or Weak

Since previously studied popular techniques were PER and CB (inherently strong consistency based approaches), extensive research was conducted on them in literature.

Chen et al. [4] developed cache replacement policy based on updates. PER and CB were modified and refined to Server Based PER (SB-PER), Revised CB (R-CB) where server keeps information about all nodes. In SB-PER, the server records all activities of the client (Original PER is stateless) and maps all clients with the requested data items. This enables the server to decide cache replacements based on the updated values.

Coming to weak consistent approaches, Lim et al. [5] proposed ConSens a target consistency based approach with underlying assumption that not every application requires a 100% consistency. It's based on IR/UIR approach. Proposed were the following three suggestions to the existing approach. They are i) Opportunistic access, ii) Lazy request and iii) Multi-data transmission.

B. Push, Pull or Hybrid

There are 3 ways of accessing data depending on which entity initiates the communication. They are push, pull and hybrid models.

Lee et al. [3] proposed FW-DAS. In second-tier, push mode is utilized to actively receive updates on BS aggressively and serving clients with updated information. In pull mode, the updated value of an object is notified to BS and

it requests the item only on local popularity. While in pure invalidation mode, the updated object is notified to BS and BS requests it upon client's demand only. This approach can be categorized as a hybrid approach.

C. Stateful or Stateless

A server can be stateful (knowing) or stateless (unknowing). A stateful server keeps track of its clients. On the other hand, a stateless server does not track its clients.

Tiwari et al. [6] explored a stateless and synchronous server scheme based on sending only hot data items in IR increasing the effective cache hit ratio. It is true in general that stateless servers are scalable while stateful servers require adequate resources to become scalable and handle scalability with difficulty. In our point of view, having enough knowledge about clients will enhance server capabilities in taking better decisions about cache admission, control and replacement policies.

III. PROPOSED SCHEME

A. System Model

We consider a wireless network with clients having single-hop access to server. The server communicates with clients in one-to-one manner. We use server and BS interchangeably as BS acts as source of information and is transparent to database server from node's point of view. Also, the server maintains a queue for client requests separately.

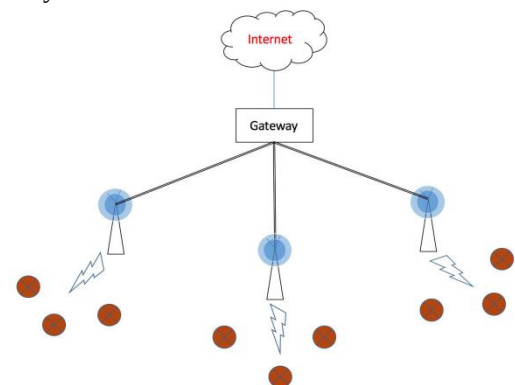


Figure 2. System Model

Every node has a cache and stores frequently requested data with it. Every cached data item has a time stamp (TS) and value associated with it. A client request is uniquely identified by the ID of the item/items requested. The BS updates as well as tracks updated data items whereas clients are only given read permissions on them. Finally, clients maintain a record of previously answered queries in a window called consistency window for inner operations.

In our approach, every node subscribes to a server with its own consistency level and accepts occasional inconsistencies. The consistency is a threshold value that is assigned by the user which indicates the update tightness between the source data item stored at the server and its cached local copy at the client. The consistency is considered strong and weak based on assigned threshold

value which ranges between 1 and 0. The current consistency of the system is strongly influenced by the consistency window. Lastly, every node is eligible to input its threshold value as a system configuration parameter to enable consistency for various applications.

Previous research on PER emphasized on investigating the client's energy problem, to let the clients operate in sleep state when there is no request or query. It caused high query delay and long disconnection period which effected the overall performance of the system. Thus, we made a few assumptions and found that strong consistency was not always a mandatory criteria and users not getting effected critically when this assumption was relaxed. Also we believe users possesses adequate knowledge on how frequently and rigorously they needed potential updates. Hence, we propose a user-defined consistency sensitive cache invalidation scheme, called PER-WC, to support diverse consistency requirements of users and applications.

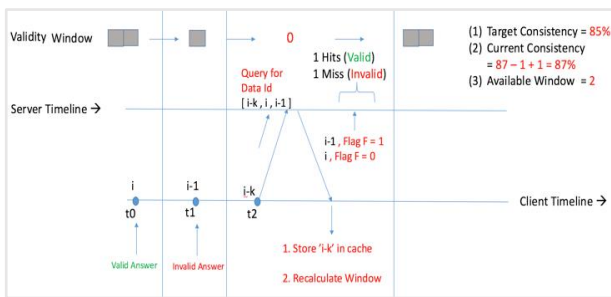


Figure 3. Validity Window Mechanism

B. Consistency Sensitive Data Access Algorithms

We first discuss the operation of PER-WC in detail. An ideal scenario is explained including setting up of validity window (W_v) and calculating current consistencies appropriately. The number of queries answered through cache is quantified by the validity window. Through consistency window (W_c), number of cache hits and/or number of misses are derived. In the proposed technique, client decides the necessary actions for themselves.

C. The Proposed Scheme - PER-WC

Initially a node tries to access i^{th} data item. Since it would be the first access to the item, the node has no cached value of it. It sends request ($i, 0$) (where $t=0$ represents new request) and server responds with a reply (i) with the new data. Similarly, a subsequent request for previously cached i^{th} data item is request (i, t) where t is last accessed time. The server verifies time stamp ' t ' and send appropriate reply. If the requested data item is already updated at the server, then it automatically sends the updated value to the client. Otherwise, it attains null value indicating the client to use its own cache. The presence or absence of the data value is denoted by flag being 1 and 0.

1) Consistency Condition: Every node defines a consistency condition either = 1 or a value less than 1. If it is set to 0.7, then 7 out of 10 queries are guaranteed to be valid answers (as adopted from [6]). In order to achieve consistency in PER, an adaptive validity window (w_v) is proposed that controls the consistencyAcknowledgment" and "References".

2) Validity Window: Without validation from server there is a chance that cache becomes invalid and eventually the consistency reduces below the assigned target consistency. Therefore, it is necessary to have server validation after certain number of cache attempts. Our adaptive validity window defines a limit on the number of opportunistic cache attempts.

To realize validity window, each node counts the number of invalid items (N_{inval}) and valid items (N_{val}) in a given constitutes our consistency window(W_c).consistency window (W_c). The sum of valid and invalid items

Subsequently, the current consistency(τ_c) is derived based on the above mentioned parameters as following:

$$(W_c) = (N_{\text{inval}}) + (N_{\text{val}}) \text{ and } (\tau_c) = (N_{\text{val}}) / (W_c)$$

An item is considered valid if it was cache hit , invalid if cache miss. More the valid items, more is the current consistency(τ_c). More the current consistency(τ_c), more is the validity window(w_v).

```

Calculate_Velocity_Window()
Limit = Nval/Wc
While ( Tc >= Tt )
    Nval -- = 1;
    New window ++;
    Limit = Nval/Wc;
    Wv = New window ++;

```

Figure 4. Pseudo Code of Validity Window

An item is considered valid if it was cache hit , invalid if cache miss. More the valid items, more is the current consistency(τ_c). More the current consistency(τ_c), more is the validity window(w_v). The validity window is the outcome of the difference between current and target consistency(τ_t) i.e.,

$$(w_v) = (\tau_c) - (\tau_t)$$

The validity window (w_v) is inversely proportional to the target consistency (τ_t). If targetconsistency (τ_t) reaches 1, the validity window (w_v) will become 0. This is because current consistency (τ_c) cannot be > 1 and if it is 1, the difference between current and target would be 0.

IV.RESULTS

A. Simulation Testbed

The proposed approach is carefully tested with the below test bed. The query and update inter arrival rate follow exponential distribution. The model is realistically designed with respect to data access; the entire data items in the

database are classified into two types: hot and cold data items. Based on probability, 80% and 20% of query requests are for hot and cold data items, respectively. 33% of update requests are for hot data items but they are uniformly distributed within the hot subset. Likewise, 67% update requests are for cold subset. Similar data query and update patterns are found in [14]. Here, we do not consider multiple update frequencies for data items. Each node caches 5% of the data items in the database, i.e., 50 cache slots. We assumed N (1000) data objects at the server and the relative frequency for data objects follows a Zipf-like distribution. In terms of cache replacement policy, we have used Least Replacement Policy(LRU). The target consistency ranges from 0.7 to 0.95 with 0.05 steps, but we use 0.8 and 0.9 in most experiments unless otherwise stated to clearly see the performance differences. The consistency window is set to the last 100 queries.

B. Simulation Results

We have performed extensive simulations on PER-WC. To evaluate PER-WC with PER, we have developed schemes on a discrete event-driven simulator based on C, CSIM. The key simulation parameters varied are: query rate, update rate, target consistency, validity window and consistency window.

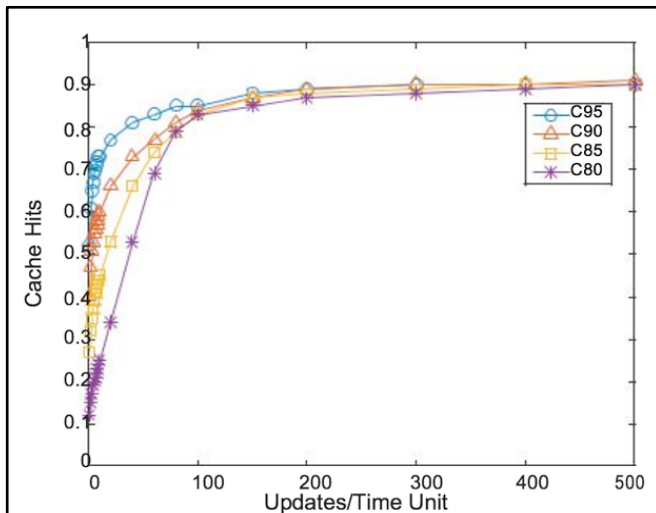


Figure 5. Cache hits (for $W_v = 5, 10, 15, 20$)

We first evaluate the performance of proposed PER-WC schemes in terms of query delay, cache hits, number of PER-WC uplink requests, adaptive consistency.

TABLE 1. SIMULATION PARAMETERS

Parameters	Values
Number of nodes	100
Database Size	1000 items
Hot data items	5% of DB
Cold data items	95% of DB
Mean Query Interval time(t_q)	50 – 100s
Mean update Interval time (t_u)	10 – 1000s
Adaptive Window	5 – 30
Target Consistency (T)	0.7 – 0.95
Consistency window (wc)	100 queries

The existing scheme based on PER has been modified to become aware of sensitive consistency: PER-weak consistency (represented as PER-WC). We have included PER scheme as the base case for comparisons.

C. Cache hit ratio

We evaluate the cache hit as a function of update interval. In the PER-WC scheme, a cache hit occurs in the following cases: (i) The consistency condition is not satisfied, and the validated local cached data (against server's copy) is identified to be still valid and (ii) The consistency condition is satisfied, and a cached data item used in advance turns out to be valid after polling the server.

In Fig. 5, at the beginning of the simulation, the number of valid cached items is higher (0.2-0.6) for higher consistencies (0.8-0.95) but eventually they do not affect the cache hit much. PER-WC shows little variation in expected results until update rate = 100 and henceforth has the similar cache hits as PER regardless of the query intervals. As shown in Fig. 5, the PER-WC scheme shows slightly lower cache hit than the PER scheme depending on the target consistencies. As the target consistencies decreased to 0.8, validity of cached data items decreased and cache hit ratio decreased. The cache hit ratio is ranged from (85% - 90%).

V. CONCLUSIONS

In this thesis, we developed PER-WC scheme, which is a consistency sensitive approach to existing PER mechanism. Specifically, we have evaluated the effects of adaptive consistency on query latencies. Our proposed model extends PER scheme and improves cache hit ratio by aggressively adapting and using cached data items. Another feature of PER-WC is its smooth integration with user-defined consistency. We introduced a window to adaptively change the current consistencies of the clients. From the simulation results it is also evident that our algorithm outperforms the existing approach.

We would like to perform simulations against various other parameters namely query intervals, consistency window to verify the correctness of this approach.

ACKNOWLEDGMENT

This research was financially supported by the Ministry of Trade, Industry and Energy(MOTIE) of the Republic of Korea and Korea Institute for Advancement of Technology(KIAT) through the Regional Specialized Industry Development Program(Grant Number : R0005598).

Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology supports this work. (Grant number: NRF-2011-0023076).

REFERENCES

- [1] (2015) Number of mobile subscribers. [Online]. Available: <https://gsmainelligence.com/>
- [2] (2015) Population in the world. [Online]. Available: <http://www.census.gov/popclock/world>

- [3] G. Lee, I. Jang, S. Pack, and X. Shen, Fw-das: Fast wireless data access scheme in mobile networks, *Wireless Communications, IEEE Transactions on*, vol. 13, no. 8, pp. 42604272, 2014.
- [4] H. Chen, Y. Xiao, and S. V. Vrbisky, An update-based step-wise optimal cache replacement for wireless data access, *Computer Networks*, vol. 57, no. 1, pp. 197212, 2013.
- [5] S. Lim, Y. Lee, J. Cheon, M. Min, and W. Wang, User-defined consistency sensitive cache invalidation strategies for wireless data access, *Computer Communications*, vol. 41, pp. 5566, 2014.
- [6] R. Tiwari and N. Kumar, An adaptive cache invalidation technique for wireless environments, *Telecommunication Systems*, pp. 117, 2015.



Mangal Sain received the M.Sc. degree in computer application from India in 2003 and the Ph.D. degree in computer science in 2011. Since 2012, he has been an Assistant Professor with the Department of Computer Engineering, Dongseo University, South Korea. His research interest includes wireless sensor network, cloud computing, Internet of Things, embedded systems, and middleware. He has authored over 30 international publications including journals and international conferences. He is a member of TIIS

and a TPC member in several international conferences



Srikanth Varanasi completed his M.S (2016) in computer science at Texas Tech University. He received the Bachelor's degrees in computer science from Vellore Institute of Technology in 2010. His area of research includes wireless networks, computer networks and data communications. He is currently working a software engineer in Texas, USA.



Young Jin Kang

2013: BS at Dongseo University, Republic of Korea
 2015: MS at Dongseo University, Republic of Korea
 2015 ~ current: doctor's course Dongseo University, Republic of Korea
 Research Interests: Wireless Sensor Networks, Cryptography and Network Security, Side Channel Analysis



Hoon-Jae Lee received his BS, MS, and PhD. degrees in Electrical Engineering from Kyungpook National University, Daegu, South Korea, in 1985, 1987, and 1998, respectively. He is currently a professor in the Department of Information and Communication Engineering at Dongseo University. From 1987 to 1998, he was a research associate at the Agency for Defense Development (ADD). His current research interests include developing secure communication system, side-channel attack, and

ubiquitous sensor network/radio frequency identification security.