Probabilistic Approach in Broadcast-based Cache Invalidation of Location Dependent Data

Y. Kaminishi†, Y. Arakawa‡, S. Tagashira‡, H. Hayakawa‡, T. Kitasuka* and A. Fukuda‡

†FUJITSU Kyushu Network Technologies
2-2-1, Momochihama, Sawaraku, Fukuoka-shi, Fukuoka, 814-8588, Japan
‡Graduate School/Faculty of Information Science and Electrical Engineering, Kyushu University
744 Motooka Nishi-ku, Fukuoka 819-0395, Japan
* Graduate School of Science and Technology, Kumamoto University
2-39-1 Kurokami, kumamoto-shi, Kumamoto, 860-8555, Japan

ABSTRACT

In this paper, we propose a broadcast-based protocol for cache invalidation in location-aware system, wherein each client uses local caches of location dependent data to reduce network accesses. The proposed protocol is designed to avoid sleep fragmentation in clients and reduce waiting time for getting required invalidation information. To realize short waiting time, we use Bloom filter to express and compress invalidation information and shorten a time for accesses to geographically close invalidation information on broadcast schedule. Also we propose three broadcast scheduling methods: FASM (Fragmentation Avoidance Scheduling Method), MBSM (Metrics Balancing Scheduling Method), and MASM (Minimizing Access time Scheduling Method). The effectiveness of the proposed methods is evaluated by simulation.

Keywords: Data caching, broadcast-based consistency protocol, broadcast scheduling, probabilistic approach

1 INTRODUCTION

Wide spread of wireless network makes it possible to provide various promising ubiquitous services everywhere. In particular, current mobile devices such as cellular phone and Ultra-Mobile PC (UMPC) with WiMAX or 3G enable us to access various beneficial information from out of doors, e.g., travel, shopping, event information, and so on. Our target in this paper is the geographical information service that is one of the most useful ubiquitous service in outdoor environment.

Geographical information services are usually made by web technologies such as web server, browser and HTML. Web technology has the advantages that it enables developers to build services easily and allows users skip the step of an install-and-setup. However, it has disadvantages with an indispensable Internet connection due to the underlying client-server architecture, which is main factor for draining of the battery. Therefore, power-saving technology is one of the most important issue for improving a convenience of geographical information service.

Data caching has been known as the effective power-saving technique which stores the frequently accessed data into a local storage on mobile device in advance. It have been shown that data caching can reduce network traffic by accessing already stored data preferentially [1]-[10]. However, caching is impossible to always provide the latest data because it is the pre-stored data. Therefore, we need to invalidate the old cache to prevent users from viewing the past (incorrect) information unconsciously.

In this paper, we consider data caching in geographical information services and invalidation mechanism for the cache through broadcast media. And also we propose a broadcast-based cache invalidation protocol that takes the waiting time and the fragmentation into account. Our proposed protocol adopts Bloom filter to express and compress invalidation information and shorten a time for accesses to geographically close invalidation information on broadcast schedule. In addition, we propose three extended broadcast scheduling methods FASM (Fragmentation Avoidance Scheduling Method), MBSM (Metrics Balancing Scheduling Method), and MASM (Minimizing Access time Scheduling Method). FASM and MASM especially focuses on avoiding the fragmentation or the waiting time respectively. MBSM aims to reduce the waiting time with less fragmentation.

We evaluate the average waiting time and the number of fragmentations by computer simulation. The results indicate that, as compared to FASM, MBSM can reduce 20% of the average waiting time with 25% increase of the number of fragmentations. Also, MASM can reduce 22% of the average waiting time with 125% increase of the number of fragmentation.

2 CACHE INVALIDATION IN GEOGRAPHICAL INFORMATION

In this section, we describe our assumed environment and problems caused by cache invalidation.

2.1 System Model

In this paper, we assume that client devices can cache the geographical information for reducing a power consumption. Traffic information service whose frequency of update is high and biased will be the most applicable for our protocol. In such system, clients get the information of their surrounding or given area via wireless network from the original geographical information that is kept in the server. Client can search and view the information anytime anywhere without

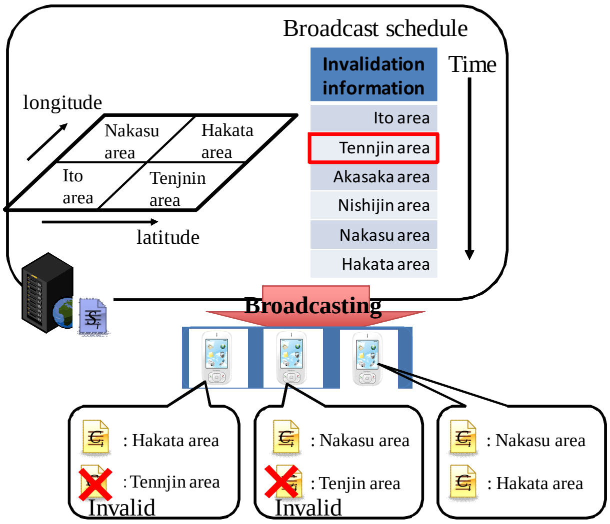

Figure 1: Cache invalidation with broadcast-based protocol

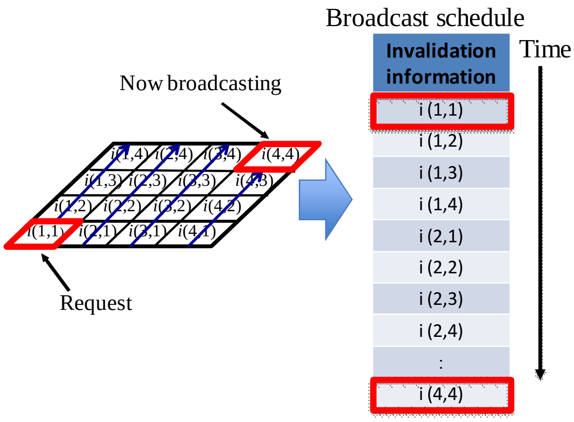

Figure 2: Example of the maximum waiting time

accessing the server since it cached the information in its local storage.

Although cache can decrease the power consumption necessary for the network access, client may unfortunately use old information. Therefore, we need to invalidate the old information. Cache invalidation protocols can be mainly classified into two types: i.e., push type and pull type. A push type protocol can deliver the information with low communication cost regardless of the number of clients. In addition, the power consumption required for checking the invalidation information is low because the client can check only by receiving the information from the server through broadcast media. In general, the power consumption for receiving is lower than that for sending a data. On the other hand, a pull type protocol needs to access the server periodically for checking the invalidation information. Since the load of the server will increase in proportion to the number of clients, the scalability of this protocol is low. We focus on the push type broadcast protocol because the geographical information service is usually used by many users and the number of users is changeable, in addition to the above reason.

In broadcast-based protocol shown in Fig. 1, first, geographical information is divided into some piece of data based on its latitude and longitude. The data of each area is broadcasted in a given order which is called **broadcast schedule**. As one of the dividing method, mesh partitioning has been proposed. When the client uses the cached data, the client needs to monitor the broadcast schedule and check the invalidation information of the areas included in the cache. If the update exists, the client gets and uses the latest information from the server. Meanwhile, the client can use the cached data when there is no update.

2.2 Problems of Cache Invalidation

In this section, we point out several technical issues on broadcast-based cache invalidation.

- In broadcast-based protocol, before accessing the cache, the client has to receive the broadcasted cache invalidation information. However, the cache invalidation information is not always broadcasted at a time when the client accesses the cache. "**A waiting time**" used in this paper is defined as a period from the time the client requests the invalidation information to the time the information is actually broadcasted to the client. In general, the average waiting time is $T/2$ when the broadcast cycle is T. Fig. 2 shows the situation where the waiting time becomes maximum. In this figure, a region is divided into 16 area like mesh. Each area is represented by $i(x, y)$ where x and y is the latitude and longitude respectively. Hence, the client requests the cache invalidation information of $i(1, 1)$ although the server broadcasts those of $i(4, 4)$.

- If the broadcast schedule is delivered in advance, it is not necessary for the client to always monitor the broadcasted information. Since the client can be in a sleeping mode during the period for broadcasting unnecessary information, it can dramatically decrease the power consumption required for receiving the data. However, if the necessary invalidation informations are dispersed on the broadcast schedule, the client needs to repeat the start-up and the sleep-down within one broadcasting cycle. We define the condition that the necessary invalidation informations are dispersively assigned on the broadcast schedule as "**the fragmentation**". Fig. 3 shows that the fragmentation occurs because the invalidation informations on the requested (geographically close) areas (i.e., $i(1, 1)$, $i(1, 2)$, $i(2, 1)$, and $i(2, 2)$) are dispersively assigned on the schedule.

In broadcast-based cache invalidation protocol, it is important to reduce the waiting time and the fragmentation. And thus, we propose a novel cache invalidation protocol that takes both functions into account.

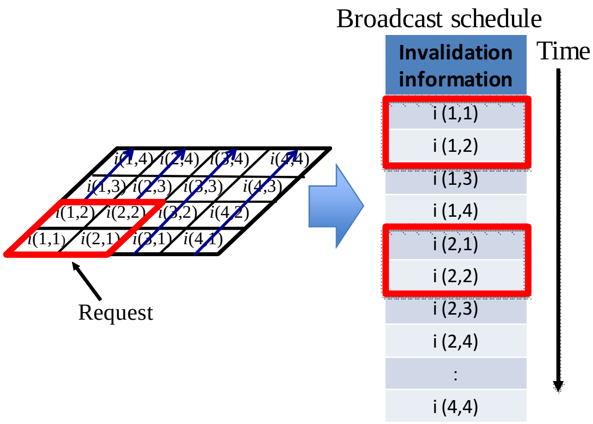

Figure 3: Example of the fragmentation

3 PROPOSED BROADCAST-BASED CACHE INVALIDATION PROTOCOL

Our proposed protocol adopts the efficient data structure for expressing the cache invalidation information. And we propose several broadcast scheduling method for reducing the waiting time and the fragmentation. Key points of our idea are shown as follows;

- We adopts Bloom filter as the data structure for reducing the size of each invalidation information because the broadcast cycle is shortened by making the size of the information smaller, which leads to reduce the average waiting time.
- We propose three scheduling methods for each purpose; i.e., FASM can avoid the fragmentation, MBSM can improve the waiting time with less fragmentations, and MASM can reduce the waiting time.

3.1 Data Structure of Cache Invalidation Information

We adopts Bloom filter as the data structure of expressing the invalidation information. Bloom filter proposed by Burton H. Bloom in 1970 is a stochastic structure with high space efficiency and is used for checking fast whether a given element is included in a set or not.

It can represent a set with a bit sequence of length m. It inputs an element i in a set U into some hash functions, and sets the $h_j(i)$-th bit responded to the obtained hash value $h_j(i)$ to 1. It does this operation for all the elements in a set U. At the checking, it checks whether $h_j(u)$-th bit of the expected element u is 1 or not. A hash value $h_j(u)$ can be obtained by using the same hash function h_j. If all the value at the hash value of all the elements is 1, element u is included in a set U. How to use Bloom filter in our proposed protocol is described as follows.

Let U denote a set of update information in a given area and $bf(U)$ denote a bloom filter for U. The clients can get $bf(U)$ as invalidation information over broadcasting from the server. By using $bf(U)$, each client can check whether each cache u stored in its local storage is updated or not, i.e., u is included in a set U or not. However, by the conflict of hash values, the cache may not be updated even if all the bits are 1. Such false detection is generally called false positive, and its incidence rate is represented as $(1-(1-\frac{1}{m})^{kn})^k$, where the number of hash functions and elements is k and n, respectively. Note that it also depends on a bit length of the bloom filter and the number of elements in a set.

3.2 Broadcast Scheduling Method for Cache Invalidation Information

In this section, we explain three proposed broadcast scheduling method: i.e., FASM (Fragmentation Avoidance Scheduling Method), MBSM (Metrics Balancing Scheduling Method), and MASM (Minimizing Access time Scheduling Method).

3.2.1 FASM

The main purpose of FASM is to avoid the fragmentation. We focus on the client's access pattern to geographical information, and reflect the pattern to the area assignment on broadcast scheduling. In the geographical information service, the client usually accesses the information of arbitrary areas and/or its adjacent areas. Therefore, we schedule such areas' invalidation information to be broadcasted in a continuous manner.

In FASM, a set of invalidation information is mapped on $2^{n-1} \times 2^{n-1} (n \geq 2)$ area. Let $I_0(x,y)$ denote a set of invalidation information mapped on a basic area (x,y) $(x,y = 1,2,\ldots,2^{n-1})$, and $I_p(x,y)$ denote a set of invalidation information mapped on (larger) square area of $2^p \times 2^p (p \leq n-1)$, where each vertex of the square is (x,y), $(x+2^p,y)$, $(x,y+2^p)$, and $(x+2^p,y+2^p)$ respectively, i.e., the square consists of a set of basic areas. For example, we can describe the invalidation information in the whole area of this system as $I_{n-1}(1,1)$.

We denote the broadcast schedule for $I_{n-1}(1,1)$ as an ordered set $S_f(I_{n-1}(1,1))$. In FASM, $S_f(I_{n-1}(1,1))$ is formally defined as follows:

$$S_f(I_q(x,y)) = \begin{cases} [S_f(I_{q-1}(x,y)), & \\ S_f(I_{q-1}(x,y+2^{q-1})), & \\ S_f(I_{q-1}(x+2^{q-1},y)), & \\ S_f(I_{q-1}(x+2^{q-1},y+2^{q-1}))] & \text{if } q>0, \\ I_0(x,y) & \text{if } q=0. \end{cases}$$

Here, we show the behavior of the proposed method by using a concrete example. Let $n=3$; i.e., suppose the broadcast schedule $S_f(I_2(1,1))$ for 4×4 area. Fig. 4 shows the scheduling process of FASM. By using the above equation, $S_f(I_2(1,1))$ is calculated as [$I_0(1,1)$, $I_0(1,2)$, $I_0(2,1)$, $I_0(2,2)$, $I_0(1,3)$, $I_0(1,4)$, $I_0(2,3)$, $I_0(2,4)$, $I_0(3,1)$, $I_0(3,2)$, $I_0(4,1)$, $I_0(4,2)$, $I_0(3,3)$, $I_0(3,4)$, $I_0(4,3)$, $I_0(4,4)$].

Each element in $S_f(I_{n-1}(1,1))$ is formed in a bloom filter and broadcasted to the clients in the order of $S_f(I_{n-1}(1,1))$

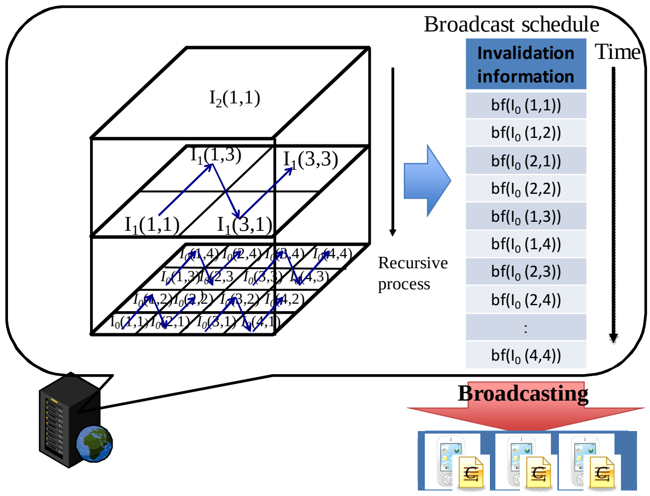

Figure 4: Broadcast scheduling in FASM

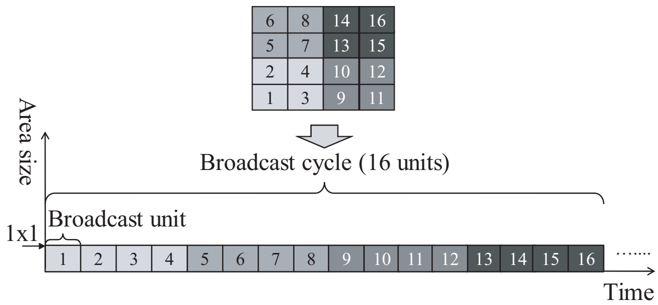

Figure 5: Example of broadcast schedule for 4×4 area in FASM

as shown in Fig. 5. In this figure, each element is transmitted in a broadcast unit and "Area size" represents the size of broadcasted areas included in one broadcast unit, i.e., in this example, all the elements are broadcasted as invalidation information for 1×1 area (i.e., basic area).

By scheduling like this, the client can receive continuously (without fragmentation) all the informations included in a set $\bigcup_{k=0}^{n-1} \bigcup_{l_x=1}^{2^{n-k-1}} \bigcup_{l_y=1}^{2^{n-k-1}} I_k(1+(l_x-1)2^k, 1+(l_y-1)2^k)$, i.e., the set includes $I_2(1,1)$, $I_1(1,1)$, $I_1(1,3)$, $I_1(3,1)$, and $I_1(3,3)$, in addition to the basic areas. For example, when the client requests the invalid information for $I_1(1,1)$, it just gets from $I_0(1,1)$ to $I_0(2,2)$ on the broadcast schedule.

Next, we explain the process of receiving at the client. When the client receives the invalidation information of the expected area, it checks whether update for now requesting cache exists or not by using a bloom filter. If no updates exist, the client uses its cache soon. If it considers that the updates may exist, it directly requests the server to check the updates. When the existence of the updates is confirmed, the client obtains the update information from the server and uses it.

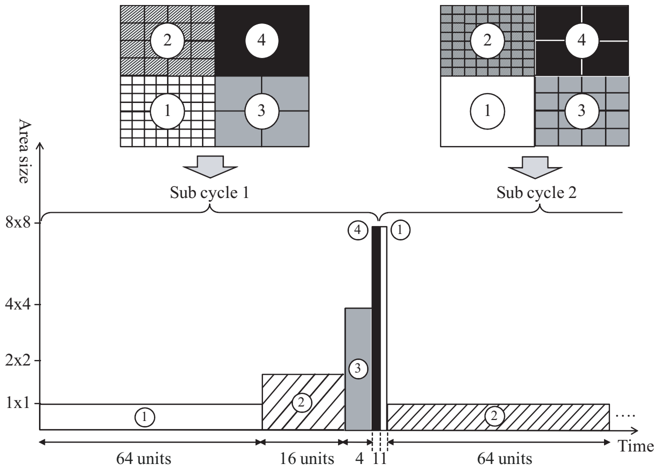

Figure 6: Example of broadcast schedule for 16×16 area in MBSM ($j = 2$)

3.2.2 MBSM

MBSM is the extended protocol of FASM, which can reduce the waiting time and keep the consecutiveness of the broadcast schedule of neighbor areas. MBSM divides the broadcast cycle into $2^j (j \geq 1)$ sub cycles, where j meets $2^j - 1 + \lceil \frac{j}{2} \rceil \leq n-1$. In each sub cycle, all areas' informations are broadcasted as in FASM. However, some areas' informations are assembled as a larger area's information (e.g., 2×2, 4×4, or above), which is transmitted at a broadcast unit, i.e., the aggregation is realized by including larger area's information in one bloom filter. The maximum size can be defined as $2^{2^j-1} \times 2^{2^j-1}$.

Fig 6 illustrates an example of the broadcast schedule for 16×16 area when $j = 2$. In this example, the broadcast cycle is divided into four sub cycles. From this figure, we can find that the invalidation information is broadcasted in different units and different area size; e.g., in the first stage of sub cycle 1, the area size included in each broadcast unit is 1×1 and the total number of broadcast units for this stage is 64. As for the second stage, although the total size of the broadcasted area is the same as that in the first stage, the area size included in each broadcast unit is 2×2 and thus the total number of units is 16, i.e., we can decrease the number of broadcast units. Note that the schedule of areas included in each stage is the same as in FASM, except for the size of basic area.

It is worth noting that, if many larger areas' informations are assembled in one broadcast unit, we can reduce the total number of areas to be broadcasted. As a result, both broadcast cycles of all the area and the waiting time of the clients are reduced. However, it causes the increase of the rate of false-positive because the number of the elements (the invalidation informations) included in one bloom filter increases, as was described in Section 3.1.

In MBSM, broadcast schedule $S_{mb}(I_{n-1}(1,1))$ for area $I_{n-1}(1,1)$ can be formally defined as follows;

1. $\theta \Leftarrow 1, S_{mb} \Leftarrow \phi, O \Leftarrow \phi$

2. Calculate the ordered set $S_f(I_{n-1}(1,1))$ by using the algorithm mentioned in Section 3.2.1.

3. Bring out element e from the head of $S_f(I_{n-1}(1,1))$. Repeat the following operations (a)∼(d) until $S_f(I_{n-1}(1,1))$ becomes vacant.

   (a) Define $i(\geq 1)$ as the order of e in $S_f(I_{n-1}(1,1))$.

   (b) $O \Leftarrow O \cup e$

   (c) $\alpha \Leftarrow |2^j - \theta + \lfloor \frac{i}{2^{2(n-1)-j}} \rfloor + 1| \pmod{2^j}$

   (d) If $i \pmod{4^\alpha}$ is 0, add O to the tail of $S_{mb}(I_{n-1}(1,1))$, and $O \Leftarrow \phi$.

4. $\theta \Leftarrow \theta + 1$

5. If $\theta \leq 2^j$, go back to 2. Otherwise finish.

Next, we explain the behavior of the clients. The process when the client receives the invalidation information of basic area is same as FASM. However, the process is different when the client receives the invalidation information about larger area. In such area, it is considerable that the rate of the false-positive is relatively high compared with those of basic areas. Therefore, if it detects no update information of all areas except those of the basic area, it uses cache soon. However, if it detects the update, it waits the next broadcasting for the smaller area whose false-positive rate is lower without directly requesting the server to check the update. And the client continues this process until receiving the information of the basic area.

MBSM can drastically reduce the waiting time when the no update exists and no update is detected by the bloom filter of all the areas except the basic area. Decrease of the waiting time becomes more effective when no update can be detected by the bloom filter of a larger area. Since the information of the client's requested area has to be broadcasted once, MBSM can improve the detection of the non-update and reduce the waiting time.

3.2.3 MASM

MASM aims to decrease the average waiting time. In general, the average waiting time depends on the broadcast cycle of the client's requested area. Therefore, we propose the scheduling method where the broadcasting periods of the arbitrary areas are uniformly (aggressively) assigned on the timeline. MASM also extends FASM to be the redundancy where the invalidation information of other area and other size is inserted to the schedule as the redundancy part. Fig 7 illustrates an example of the broadcast schedule for 16×16 area when $j = 2$. From this figure, we can find MASM sets the redundancy aggressively as compared to MBSM, although the fragmentation increases.

When the maximum area size of inserting information is $2^{2^j-1} \times 2^{2^j-1}$, the broadcast schedule $S_{ma}(I_{n-1}(1,1))$ of area $I_{n-1}(1,1)$ can be obtained as follows.

1. $c \Leftarrow 1, S_{ma} \Leftarrow \phi$

2. $O_1, O_2, \ldots, O_{2^j} \Leftarrow \phi$

3. Calculate the order set $S_f(I_{n-1}(1,1))$ by using the algorithm mentioned in Section 3.2.1.

4. Repeat the following operations (a)∼(d) for all the $r(= 1, 2, \ldots, 2^j)$.

   (a) $i \Leftarrow 2^{2(n-1)-j}(r-1) + c \pmod{2^{2(n-1)}}$

   (b) If i is 0, set $2^{2(n-1)}$-th element of $S_f(I_{n-1}(1,1))$ to e. Otherwise, set i-th element to i.

   (c) $O_r \Leftarrow e$

   (d) If $c \pmod{4^{r-1}}$ is 0, add O_r to the tail of $S_{ma}(I_{n-1}(1,1))$ and $O_r \Leftarrow \phi$.

5. $c \Leftarrow c + 1$

6. If $c \leq 2^{2(n-1)}$, go back to 4. Otherwise finish.

The process when the client receives the invalidation information is same as MBSM. MASM realize to reduce the average waiting time by setting the redundancy. Although this redundancy is realized by using larger area rather than basic area and thus, reducing the size of information, excessive redundancy makes the total broadcast cycle longer. Therefore, it is important to set optimal redundancy for achieving the good performance with MASM. Also, we need to consider that the FASM's sequentiality is lost by MASM.

4 PERFORMANCE EVALUATIONS

In this section, we evaluate our proposed methods, FASM, MBSM and MASM by computer simulations in terms of the average waiting time and the fragmentation.

4.1 Simulation Environment

We adopt the information of facilities as the broadcasted geographical information. The information of facilities is mapped on the 2-D mesh of latitude and longitude. Each mesh is divided in 32 seconds (about 1 km) scale based on its coordinate. Mesh divided in 32 seconds scale has been defined in [13] as the mesh for searching, and it can cover the whole world with 4 bytes expression. In this simulation, we map $1,052,890$ facilities' informations on 256×256 mesh. We assume facilities' informations in our simulation is distributed uniformly, although they exist disproportionately in the real world. As a result, the number of facilities' informations per area can be expressed by $\frac{shop\_num}{area\_num}$, where the total number of facilities' informations and areas are $shop\_num$ and $area\_num$ respectively. In addition, the number of updates per area can be expressed by $\gamma \times \frac{shop\_num}{area\_num}$, where γ represents the update frequency of facilities's informations. Simulation procedure is described as follows.

1. Determine the invalidation information requested by the client. Requested area is represented by $I_p(x,y)$. We set the three area size, $(1 \times 1,\ 2 \times 2,\ \text{and } 4 \times 4)$, and the ratio of these sizes is $(4:2:1)$.

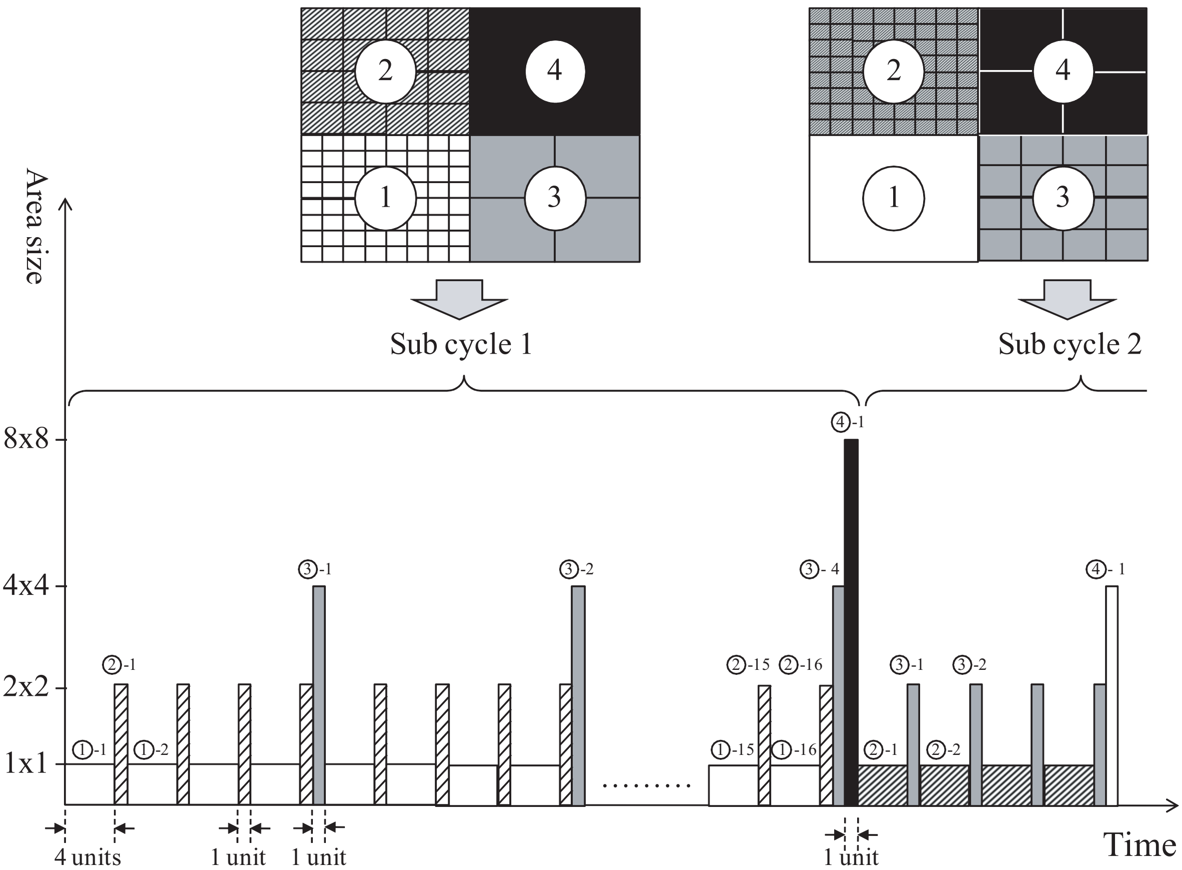

Figure 7: Example of broadcast schedule for 16×16 area in MASM ($j = 2$)

2. Determine the broadcast schedule number properly.

3. Client monitors the broadcast schedule according to the number determined at 2. Simulation will be finished when all the necessary informations are fully received.

We evaluate the fragmentation and the average waiting time by repeating above procedure $10,000$ times.

Since the frequency of update check from the client to the server becomes high when the false-positive of bloom filter of the basic area is high, we adjust the bit length to keep the false-positive 3% or lower.

According to the equation of the occurrence ratio of false-positive in bloom filter, we need at least 28 bit length if the update ratio of facilities' informations is 5%, at least 54 bit length in case of 10%. We use only one hash function in this simulation.

As the metric for evaluating the fragmentation, we employ the frequency of starting up until all the requested information is received. We call this a fragment count. As for the waiting time, we adopt the average waiting period from the time when a client sends a request to the time when all the broadcasted invalidation information is received. Also we define a time unit, called a time slot, as the period necessary for broadcasting a set of invalidation information.

4.2 Results

First, we show the trade-off between the fragmentation and the average waiting time, through the result that MBSM and MASM with more redundancy can improve the average waiting time compared with FASM which aims to avoid the fragmentation without any redundancy. We evaluate both performance of three methods by changing the update ratio and bit length of bloom filter.

MBSM and MASM are evaluated in case of the half broadcast cycle($j = 1$) and the quarter broadcast cycle($j = 2$).

Fig.8 shows the fragment count versus the bit length of a bloom filter when the update ratio is 5%. Fig.9 shows the performance in case the update ratio is 10%.

From Fig.8, we can notice that the fragment count becomes lower when the bit length becomes longer. This is because low false-positive of bloom filter due to a long bit length enables to detect no update with a bloom filter of larger areas. Especially, the performance improvement of MBSM($j = 2$) and MASM($j = 2$) that use bloom filters of larger areas shows that the improvement of false-positive of bloom filter significantly affects the fragment count.

Compared with FASM, the fragment count of MBSM($j = 1$) increases by about 25% but those of MASM($j = 1$) increases by about 110%, in case of the 28 bit length which is the least length.

Fig.9 shows that the fragment count of MASM and MBSM increase when the update frequency is high. The reason is

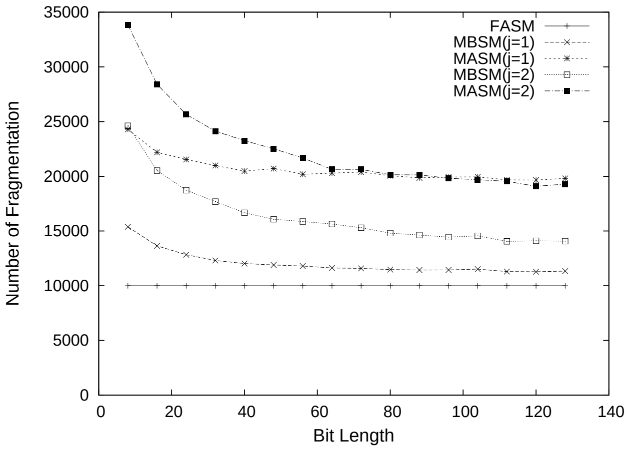

Figure 8: Fragment count of FASM, MBSM, MASM (update frequency:5%)

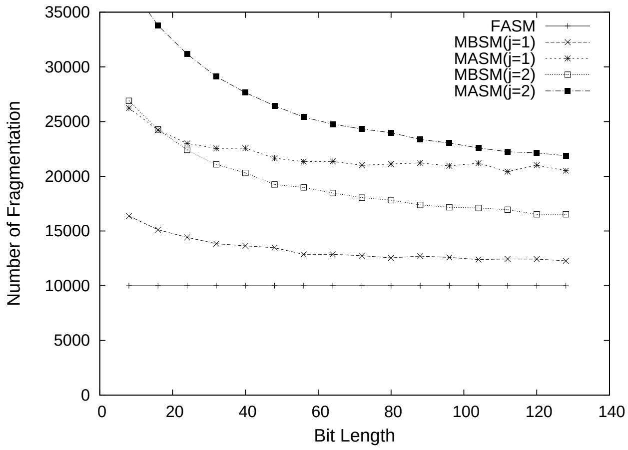

Figure 9: Fragment count of FASM, MBSM, MASM (update frequency:10%)

that the increase of false-positive due to the frequent update prevent the client from detecting the updates with a bloom filter of larger areas. All the performances are similar to those of 5%. Compared with FASM, the fragment count of MBSM($j = 1$) increases by about 45% but those of MASM($j = 1$) increases by about 125%, in case of 54bit.

Fig.10 and Fig.11 show the average waiting time versus the bit length of bloom filter when the update frequency is 5% and 10% respectively. In Fig.10, it is shown that the performance improvement of MASM and MBSM become large. This is because low false-positive of bloom filter due to a long bit length enables to detect non update with a bloom filter of larger areas. Especially in case of the 28 bit length, MASM($j = 2$) achieves the best performance that is 19% better than FASM. MBSM($j = 2$) also improves by about 12%.

The performance of Fig.11 is similar to those of 5%. However, compared with the performance of 5%, the average waiting time of MASM and MBSM increase. The reason is that the increase of false-positive due to the frequent update prevent the client from detecting the updates with a bloom filter of larger areas. Especially in case of the 54 bit length, MASM($j = 2$) achieves the best performance that is 13% better than FASM. MBSM($j = 2$) also improves by about 8%.

5 RELATED WORKS

In this section, we introduce the cache invalidation methods based on push mechanism.

As one of the stateless management approach, TS method which uses a timestamp has been proposed[1]. In TS method, MSS (Mobile Support Station) broadcast the IR (Invalidation Report) to the connected clients in a L cycle. TS method has the problem that, in a worst-case scenario, the client can't receive IR until the next cycle. Since the timestamps of original data which have been updated during $w(w > L)$ seconds is recorded in IR, the client needs to invalidate all the cache data

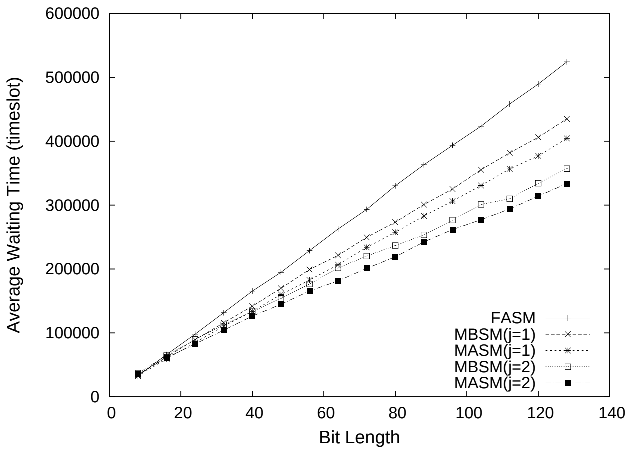

Figure 10: Average waiting time of FASM, MBSM, MASM (update frequency:5%)

if it can't connect to MMS over *w* seconds.

Updated Invalidation Report (UIR) has been proposed[10] for reducing the delay problem of TS method. UIS is also one of the push-based stateless management approach same as TS method. In UIR, the timestamps of original data which have been updated after the last broadcasted IR. By broadcasting UIR more than once between IR cycles, it reduces the delay for receiving the update. However, if the frequency of update becomes high, the size of broadcast message becomes large and it wastes the network capacity. Also, the increase of communication frequency wastes the battery.

6 CONCLUSION

In this paper, we have proposed the broadcast-based cache invalidation method for geographical information service. We adopts the bloom filter to express the invalidation information efficiently. Three methods, FASM, MBSM and MASM are proposed for each purpose. FASM aims to avoid the frag-

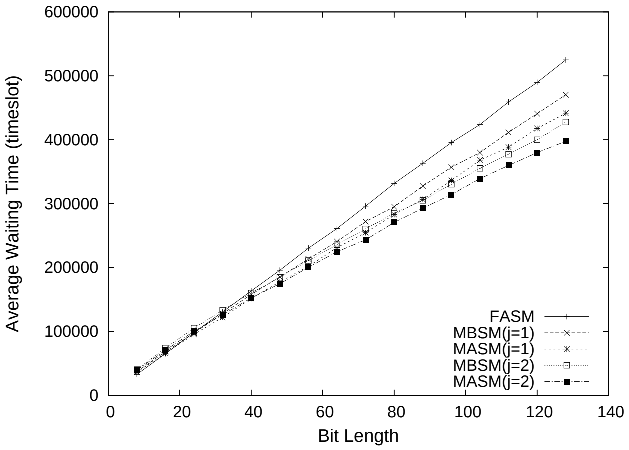

Figure 11: Average waiting time of FASM, MBSM, MASM (update frequency:10%)

mentation. MBSM, the extended method of FASM, aims to reduce the waiting time with small fragmentation. MASM aims to shorten the average waiting time. By computer simulation, we evaluate these three methods in terms of the fragment count and the average waiting time. As a result, compared with FASM, MBSM achieves about 20% improvement of the average waiting time with the increase of 25% fragmentation. MASM reduces the average waiting time by 22% against the 110% increase of the fragmentation.

In the future, we will endeavor to evaluate the proposed methods in more detail, from the view points of sensitivity of update frequency, scalability, and so on. Furthermore, in this paper, we assume the distribution of the facilities' informations is uniform, however, it is nonuniform in live environment. We will address technical issues caused in such live environment.

Acknowledgment

This work was partially supported by "The Kyushu University Research Superstar Program (SSP)", based on the budget of Kyushu University allocated under President's initiative and supported by Grant-in-Aid for Scientific Research, Japan.

REFERENCES

[1] Daniel Barbara, Tomasz Imielinski, "Sleepers and workaholics: caching strategies in mobile environments," ACM SIGMOD Conference on Monagement of Data, pp.1-12, June 1994.

[2] Sunho Lim, Wang-Chien Lee, Guohong Cao, Das, C.R., "Performance comparison of cache invalidation strategies for Internet-based mobile ad hoc networks," Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference, 25-27 Oct. 2004, pp.104- 113.

[3] Sunho Lim, Wang-Chien Lee, Guohong Cao, Chita R.DAS, "Cache invalidation strategies for internet-based mobile ad hoc netwoks," Computer Communications Volume 30, Issue 8, 8 June 2007, pp.1854-1869.

[4] A. Kahol, S. Khurana, S.K.S. Gupta, and P.K. Srimani, "A Strategy to Manage Cache Consistency in a Distributed Mobile Wireless Environment," IEEE Trans. Parallel and Distributed Systems, vol. 12, no. 7, pp. 686-700, July 2001.

[5] Yu Huang, Jiannong Cao, Zhijun Wang, Beihong Jin, Yulin Feng, "Achieving Flexible Cache Consistency for Pervasive Internet Access," Pervasive Computing and Communications, PerCom '07. Fifth Annual IEEE International Conference, 2007, pp.239-250.

[6] Jiannong Cao, Yang Zhang, Guohong Cao, Li Xie, "Data Consistency for Cooperative Caching in Mobile Environments," The flagship magazine of the IEEE Computer Society April 2007(Vol.40, No.4), pp.60-66.

[7] Zhijun Wang , Sajal Das , Hao Che , Mohan Kumar, "A Scalable Asynchronous Cache Consistency Scheme (SACCS) for Mobile Environments," IEEE Transaction on parallel and distributed systems, Vol. 15, No. 11, November 2004, pp.983-995.

[8] Zhijun Wang, Mohan Kumar, Sajal K Das, and Huaping Shen, "Dynamic Cache Consistency Schemes for Wireless Celluar Networks," IEEE Transaction On Wireless Communications, Vol.5, No.2, February 2006, pp.366-376.

[9] Chang, Yeim-Kuan Ting, I-Wei Lin, Tai-Hong, "Dynamic Cache Invalidation Scheme in IR-Based Wireless Environments," Advanced Information Networking and Applications, 2008. AINA 2008. 22nd International Conference, pp.697-704.

[10] G. Cao, "A scalable low-latency cache invalidation strategy for mobile environments," IEEE Transactions on Knowledge and Data Engineering 15 (5) 2003 pp.1251-1265.

[11] Yong Yang, Rocky Dunloap, Michael Rexroad, Brian F. Cooper, "Performance of Full Text Search in Structured and Unstructured Peer-to-Peer Systems," in Proc. IEEE INFOCOM, 2006.

[12] P.Reynolds and A.Valdet, "Efficient peer-to-peer keyword searching," in Proc. International Middleware Conference, 2003.

[13] Kiwi-W Consortium. Input for ISO Physical Storage Format, 28 March 2000. http://www.kiwi-w.org/.