Reducing Signaling Overhead in Distributed Mobility Management via Bloom Filter

Haneul Ko, Giwon Lee, and Sangheon Pack
School of Electrical Engineering, Korea University, Korea
Email: {st_basket, goodkw, shpack}@korea.ac.kr

***Abstract*—Distributed mobility management (DMM) is a promising technology to address the mobile data traffic explosion problem. However, DMM can incur significant signaling overhead to search or distribute the location information. To reduce the signaling overhead, we introduce a timer-based Bloom filter aggregation technique for distributing the location information. Preliminary simulation results show that the timer should be carefully selected to attain the optimal performance.**

I. Introduction

Recent report [1] predicts that mobile data traffic will hit an annual run rate of 130 exabytes by 2016 and high volume video will account for most of the data traffic. More specifically, the compound annual growth rate in the data traffic from 2011 to 2016 is 78 percent and the number of personal devices connected to mobile networks by 2016 exceeds 10 billion. Due to this mobile data explosion, current mobile networks suffer from scalability and performance degradation problems, and research on new mobile network architecture is ongoing. In particular, flat and distributed network architecture is perceived as a promising solution since it is possible to handle explosive mobile data traffic without processing at the core systems in mobile networks. Moreover, long term evolution (LTE) and beyond LTE systems aim to accomplish flat and distributed network architecture [2], [3]. In addition, the standardization issue on distributed mobility management (DMM) is being actively discussed in internet engineering task force (IETF).

In DMM, each access router (AR) conducts location/mobility management. In other words, when a mobile node (MN) enters the service area of the AR, the MN sends a location update message to the AR and the AR manages the location information of ARs under its service area. Therefore, location update messages do not need to be forwarded to any central server, which can significantly reduce the traffic overhead.

Even though DMM has benefit in reducing the location update traffic, it should address the issue how to setup a connection between corresponding node (CN) and MN because the location information of MNs are distributed at ARs. The simplest way to resolve this issue is to distribute the location information maintained at each AR to all other ARs or to multicast the connection setup request to all ARs (i.e., paging). However, these approaches result in significant signaling overhead, which may diminish the advantage of distributed mobility management.

To address this issue, we introduce the use of Bloom filter for distributed mobility management. The Bloom filter is a simple space efficient randomized data structure that can used to test the membership [4]. To add some elements in the Bloom filter, the elements are hashed by using one or more hash functions. Then, the bits at the position resulted from the hash functions are set to 1. On the other hand, the original Bloom filter cannot delete elements without false negative, e.g., the Bloom filter indicates that some present elements are absent. Therefore, several schemes are suggested to delete some elements by employing additional bits [5]. Since the Bloom filter manages a large number of elements with the limited number of bits, the false positive probability exists. However, the probability of false positive can be made sufficiently small by adjusting the number of hash functions and the size of Bloom filter. Hence, the Bloom filter is a attractive method for distributed systems, which need to share information with low overhead.

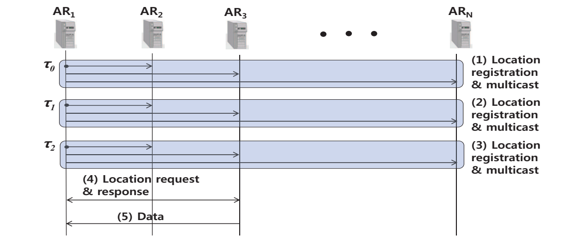
In DMM, when the MN is attached to an AR, the AR updates its Bloom filter and propagates the Bloom filter to other ARs. Since the propagation of Bloom filter for every incoming MN leads to high signaling overhead, a timer is maintained by the AR and the updates of Bloom filter are aggregated and propagated when the timer expires. Therefore, the tradeoff between consistent location information and low signaling overhead can be balanced by adjusting the value of timer.

II. Timer-based Bloom Filter Aggregation

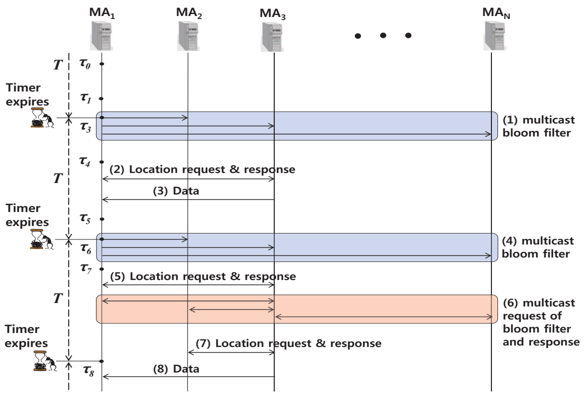
Figure 1 compares the location update and query procedures in 1) conventional DMM with multicast and 2) DMM with Bloom filter. τ_k represents the location update time epoch of the MN or the timer expiration epoch where k is an integer number. Note that we do not consider the situation of the false positive of Bloom filter for simplicity.

As shown in Figure 1(a), when MNs move into AR_1, each MN sends the location registration message to AR_1 at τ_0, τ_1 and τ_2, respectively. After registering the location of MNs, AR_1 multicasts the updated location information to all other ARs. Therefore, every AR knows the up-to-date location information of MNs and thus a session can be established without additional lookup procedure. When a CN located at AR_3 tries to send the packets to the MN at AR_1, AR_3 already knows the current location of the MN and thus it requests the location information of the MN to AR_1. After receiving the location response message, AR_3 sends the packets to AR_1.

Figure 1(b) shows the location update and query procedure when the timer-based Bloom filter aggregation technique is used. By means of this technique, multiple location update events are aggregated during the predefined timer and the aggregated Bloom filter is disseminated to other ARs in a batch manner. Consequently, the signaling overhead for location



(a) Conventional DMM with multicast.



(b) DMM with timer-based Bloom filter aggregation.

Fig. 1. The location update and query procedure.

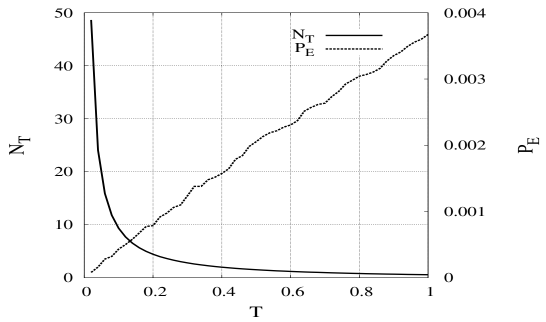


Fig. 2. N_T and P_E vs. T.

update can be significantly reduced. However, incorrect location information can be kept at ARs since location update events are notified periodically.

In Figure 1(b), the timer expires at τ_3 and AR$_1$ disseminates the aggregated Bloom filter, which includes the location update events at τ_0 and τ_1. When the CN at AR$_3$ sends the packets to the MN, the packets arrives at AR$_3$, which checks its received Bloom filters. After that, AR$_3$ requests the location information of the MN to AR$_1$ (i.e., flow (2)), and AR$_3$ sends the packets to AR$_1$ and the MN.

On the other hand, we assume that the MN moves from AR$_1$ to AR$_2$ at τ_7 and AR$_3$ has the packets to be delivered to the MN before the next timer expiration (i.e., τ_8). Since AR$_3$ has out-of-date location information (i.e., the MN is attached to AR$_1$), AR$_3$ requests the location information of the MN to AR$_1$ and therefore AR$_3$ receives the negative response from AR$_1$, which indicates that the MN moves away from AR$_1$. After that, AR$_3$ multicasts the Bloom filter request message to refresh Bloom filters. After receiving the Bloom filter request message, every AR sends its up-to-date Bloom filter to AR$_3$. Finally, AR$_3$ can send the location request message to appropriate AR (i.e., AR$_2$) and the packets can be routed to AR$_2$.

III. Preliminary Simulation Results

As mentioned earlier, the performance of the timer-based Bloom filter aggregation technique is dependent on the value of timer. Therefore, to show the effect of timer T on the performance, we conducted preliminary simulations by developing an event-driven simulator with C++ [6]. In simulations, the location query and update arrival rate are set to 1. On the other hand, the number of ARs and the number of hosts per AR are set to 8 and 100, respectively.

From simulations, we measured the average number of timer expiration events between two consecutive location query messages (denoted by N_T) and the probability that the AR has incorrect location information (denoted by P_E). As shown in Figure 2, as T increases, N_T drastically decreases whereas P_E linearly increases. In other words, lower location update cost can be achieved when T is large; however, it can increase the possibility of miss-routing for location query due to higher P_E. On the contrary, when smaller T is employed, each AR maintains up-to-date location information and reduce the chance of miss-routing for location query by disseminating the aggregated Bloom filter more frequently. However, this feature can be obtained at the expense of higher signaling overhead due to frequent Bloom filter disseminations. To conclude, there is a tradeoff between N_T (or location update cost) and P_E (or location query cost), and more work on the selection of the appropriate T is needed.

IV. Conclusion

In this paper, we proposed a timer-based Bloom filter technique to reduce the signaling overhead in DMM. Preliminary simulation results demonstrate that an appropriate timer should be carefully selected for the optimal performance. In future works, we will develop the analytic model to derive the optimal timer and investigate how to further mitigate the overhead of Bloom filter in DMM.

Acknowledgement

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

References

[1] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2011-2016, http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.pdf.

[2] B. A. Bjerke, "LTE-advanced and the evolution of LTE deployments," *IEEE Wireless Communications*, vol. 18, no. 5, pp. 4-5, 2011.

[3] Mobility Management Entity in LTE/EPC Networks, http://squiz.informatm.com/lteglobal/__data/assets/it_exhibitor_attachment/0003/335982/Tellabs_White_paper_MME_in_LTE_EPC_Networks.pdf.

[4] A. Z. Broder, *et al.*, "Network Applications of Bloom Filters: A Survey," *Internet Mathematics*, vol. 1, no.4, pp. 485-509, 2004.

[5] C. E. Rothenberg, *et al.*, "The deletable Bloom filter: a new member of the Bloom family," *IEEE Communications Letters*, vol. 14, no. 6, pp. 557-559, 2010.

[6] K. Lee, *et al.*, "Design and analysis of cooperative wireless data access algorithms in multi-radio wireless networks," *ACM Wireless Networks*, vol. 19, no. 1, pp. 17-29, 2013.