

An efficient continuous range query processing scheme in mobile P2P networks

Jongtae Lim¹ · Kyoungsoo Bok¹ · Jaesoo Yoo¹ 

Published online: 23 February 2017
© Springer Science+Business Media New York 2017

Abstract With the development of mobile devices, interest in the mobile P2P-based services has increased. However, the existing schemes spend too much cost to process a continuous range query in mobile P2P network environments. In this paper, we propose a new efficient range query processing scheme in mobile P2P network environments. The proposed scheme consists of two phases. In the query distribution phase, it prunes the peers that are impossible to be included in the query result. In the monitoring phase, the proposed scheme updates the query result incrementally. It is shown through performance evaluation that the proposed scheme outperforms the existing scheme in terms of the number of messages.

Keywords Continuous range query · Mobile P2P network · Location-based service

1 Introduction

With the development of location-aware techniques and mobile devices such as smartphones, tablet PCs, and laptop computers, location-based services (LBS) have been studied [1]. To provide location-based services, we need efficient query processing

✉ Jaesoo Yoo
yjs@chungbuk.ac.kr

Jongtae Lim
jtlim@chungbuk.ac.kr

Kyoungsoo Bok
ksbok@chungbuk.ac.kr

¹ School of Information and Communication Engineering, Chungbuk National University, 1 Chungdaero, Seowon-gu, Cheongju 28644, Korea

methods for various query types such as a point query, a range query [2], a k -nearest neighbor (KNN) query [3], a top- k query [4], and a skyline query [5].

Recently, most of the location-based services have been provided in the client–server environment. In the client–server environment, the server processes all queries. When the client requests the service, the server receives and processes the request. The server sends the query results to the client. Therefore, the role of the server is very important in the client–server environment. However, the client–server-based services have some disadvantages. In the client–server environment, when the server does not work well, the service is not provided because the server processes all of the queries. By the same reason, the bottleneck occurs in the client–server environment.

With the development of mobile devices, interest in the mobile P2P (peer-to-peer)-based services has increased. In the mobile P2P-based service, each device is called a peer. We call a peer that issues a query the query peer in the P2P-based service. To process the query, the query peer sends and receives the data with other peers [6–18]. There are no servers in the mobile P2P network environments. Therefore, the mobile P2P-based service solves the problem of a client–server-based service.

To process a continuous range query in mobile P2P network environments, the naive method processes the range query periodically and frequently. However, it has too much cost and low accuracy. Therefore, continuous range query processing schemes have been studied [6–18]. There are two approaches. In the first approach, when the query issues, the neighbor peers process the query using their local information. They send the results to the query peer. Finally, the query peer merges the results of the neighbor peers. ExtRange [6] is a typical example of this approach. In the second approach, when the query issues, the neighbor peers send their local information to the query peer. The query peer merges the information of the neighbor peers and processes the query using the merged information. SMQ [7] is a typical example of this approach. However, these approaches are still too expensive to process the continuous range query.

In this paper, we propose a new efficient range query processing scheme in mobile P2P network environments. The proposed scheme consists of two phases. In the first query distribution phase, it prunes the peers that are impossible to be included in the query results. In the second monitoring phase, the proposed scheme updates the query results incrementally.

2 Related works

2.1 Total Dominant Pruning (TDP)

In the P2P network environments, the message distribution performance is hostage to set a fixed routing path. However, the fixed routing path changes from time to time, because the peers move and the connection status of the peers changes from time to time in the P2P network environments. To process the continuous query in mobile P2P network environments, previous continuous query processing methods must use the flooding scheme. Figure 1 shows the flooding scheme. In the flooding scheme,

Fig. 1 Flooding scheme

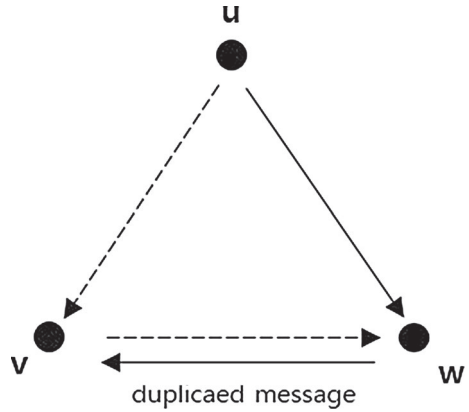
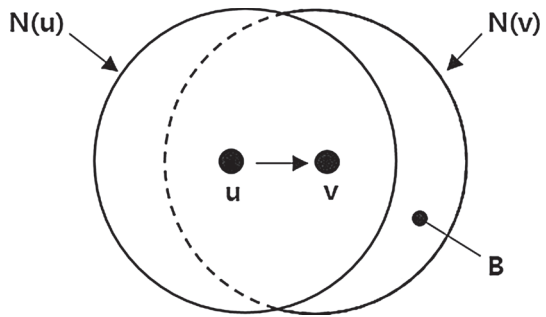


Fig. 2 Dominant Pruning



the peers send the messages to all neighbor peers. Therefore, duplicate messages are occurred when messages are sent from u to v and from w to v in Fig. 1.

To reduce the duplicated messages, Dominant Pruning (DP) and Total Dominant Pruning (TDP) have been studied [8]. Figures 2 and 3 show DP and TDP. In this scheme, $N(x)$ means neighbor peers of x . In Dominant Pruning, the peers send the message without the neighbor peers of the neighbor peers (B in Fig. 2) to prevent sending messages to the peers that already receive the messages. By the same token, in Total Dominant Pruning, the peers send the message without the two-step neighbor peers of the neighbor peers (B and U in Fig. 3). To reduce the duplicated messages, we use the Total Dominant Pruning in query distribution.

2.2 ExtRange

Efficient continuous range query processing methods have been studied. There are two approaches such as ExtRange and Spatial Monitoring Query Processing (SMQ). In ExtRange, when the query issues, the neighbor peers process the query using their local information. And then they send the results to the query peer. Finally, the query peer merges the results of neighbor peers.

Figure 4 shows the ExtRange. In ExtRange, each peer knows the location information of its own data. The query message contains information such as $\langle qid, qo, r, er,$

Fig. 3 Total Dominant Pruning

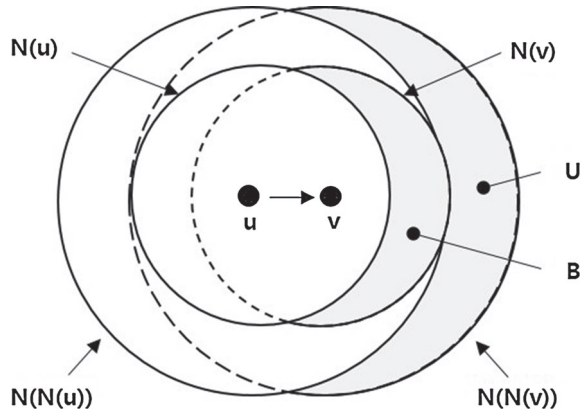
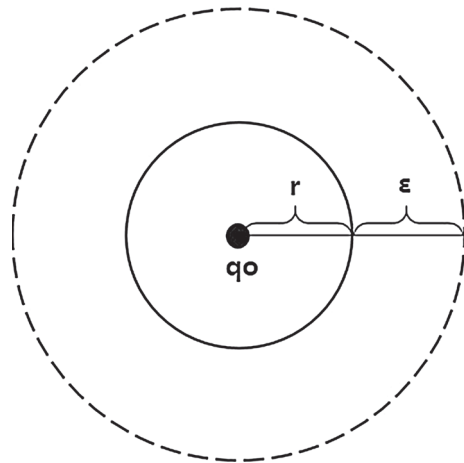


Fig. 4 ExtRange



ts, dur >. qid is a query id, qo is the object that issue the query, r is the query range, er is the extended range. ts is the query issue time, and dur is the time that passed from ts . Each peer set Safe Period (SP) and Wait Time (WT) for effective monitoring in ExtRange. SP means that the result of the query does not need to be updated. SP is calculated by Eq. 1. WT means that the query needs to be reprocessed. WT is calculated by Eq. 2. They are calculated by the maxspeed of an object and an extended range. The extended range is selected by a user. According to the extended range, the performance of ExtRange is changed excessively.

$$SP = \frac{|d_{to}(o_i, qo) - r|}{2 \times \text{Maxspeed}} \tag{1}$$

$$WT = \frac{\epsilon}{2 \times \text{Maxspeed}} \tag{2}$$

2.3 Spatial Monitoring Query Processing (SMQ)

In SMQ, when the query issues, the neighbor peers send their local information to the query peer. And the query peer merges the information of its neighbor peers and processes the query using the merged information.

Figure 5 shows SMQ. In SMQ, there are four kinds of query processing methods. They are Stationary Range Monitoring Query Processing methods (S-RMQ), Mobile Range Monitoring Query Processing methods (M-RMQ), Stationary KNN Monitoring Query Processing methods (S-KNNMQ), and Mobile KNN Monitoring Query Processing methods (M-KNNMQ). M-RMQ is a continuous range query processing method. In M-RMQ, the query peer such as N in Fig. 5 makes an inner bounding rectangle and an outer bounding rectangle. The inner bounding rectangle means boundary that the query peer can move without result update. The outer bounding rectangle means boundary that objects can move without result update. In case that the query peer leaves the inner bounding rectangle or some objects pass the outer bounding rectangle, this method reprocesses the query.

However, the existing approaches are too expensive to process the continuous range query when peers have low speed. Since they use the physical maximum speed of a peer to set SP (monitoring boundary) and WT, they suffer from flexibility.

3 The proposed query processing scheme

In this paper, we propose a new efficient range query processing scheme in mobile P2P network environments. The proposed scheme consists of two phases. In the query distribution phase, the proposed scheme prunes the peers that are impossible to be included in the query result. TDP is used for the query distribution. In the monitoring phase, the proposed scheme updates the query result incrementally.

Figure 6 shows the proposed query processing scheme. The proposed query processing scheme consists of two phases. First phase is initial query processing phase. In the initial query processing phase, the proposed query processing scheme calculates an initial result of the range query. The proposed scheme solves that ‘how to reduce

Fig. 5 SMQ-M-RMQ

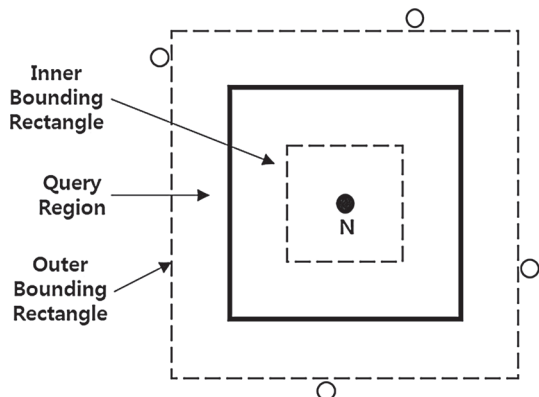
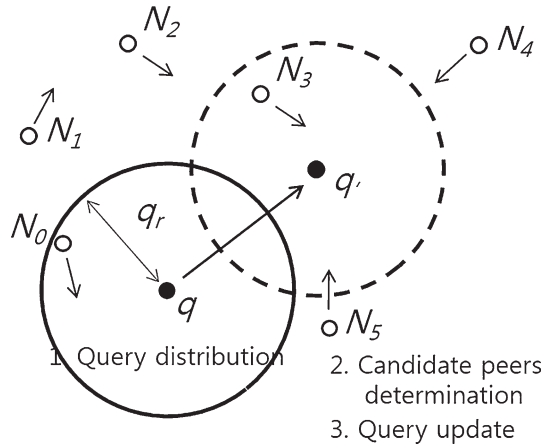


Fig. 6 The proposed query processing scheme



the duplicated messages?’ and ‘which peers will be forwarding the message?’ in query distribution step of initial query processing phase. And the proposed scheme solves that ‘which peers will be processing the query?’ in candidate peers determination step of initial query processing phase. Second phase is monitoring phase. In the monitoring phase, the proposed query processing scheme monitors and updates the result of the range query. The proposed scheme solves that ‘how to update the result of query efficiently?’ in query update step of monitoring phase.

3.1 Query distribution

A query has information such as $\langle qid, qo, qr, expiretime, keywords \rangle$, where qid is the query identifier, qo is an peers that issues the query, qr is its range, $expiretime$ is its expiration time, and the $keywords$ is the extended information for processing the query. Figure 7 shows the proposed query distribution method. In this figure, an arrow means the flow of a message, and a circle means the communication range. The query distribution method uses TDP [8] to prevent the duplicated messages. For example, query peer q issues a query. q sends the query message to its neighbor peers such as N_1 and N_2 . However, q does not send the query message to neighbor peers of N to prevent duplicated messages. The neighbor peers of N mean the peers within the communication range of peer N ($(N_1, N_2, \dots) \in N$).

In the proposed scheme, a query message is sent to peers that can be included in the range query results. To determine the candidate peers, we make the query distribution range. Figure 8 shows how to make the query distribution range. In Fig. 8, the ellipse means the range which contains the candidate peers which can be included in the range query result. In this scheme, we draw a circle C to cover candidate peers because we have too much cost to draw the ellipse E .

The coordinate (x, y) that represents the center point of the circle C and the radius r are computed by Eqs. 3, 4, and 5, respectively. These parameters are only used in order to draw a circle in Fig. 8. In this equation, $expiretime$ means the valid time of a

Fig. 7 Query distribution method

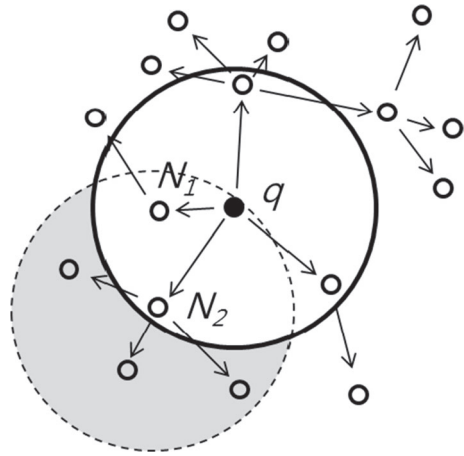
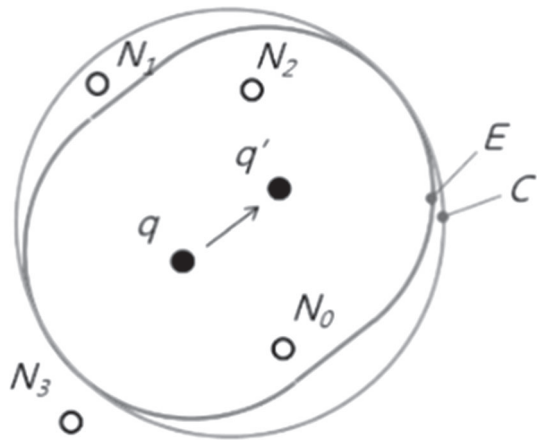


Fig. 8 Query distribution range



query. q_x and q_y are the initial coordinate of the query peer when the query is issued. qv_x and qv_y are the movement vectors x and y of the query peer.

$$x = q_x + \left(qv_x \times \frac{expiretime}{2} \right) \tag{3}$$

$$y = q_y + \left(qv_y \times \frac{expiretime}{2} \right) \tag{4}$$

$$r = q_r + \frac{\sqrt{(qv_x^2 + qv_y^2)} \times expiretime}{2} \tag{5}$$

3.2 Candidate peers determination

All peers that can be included in the range query result receive the query message by the query distribution strategy. However, all peers that receive the query message are included in the range query results. We propose an efficient continuous query processing scheme. Figures 9 and 10 show how the proposed query processing scheme computes the relative vectors and determines the candidate peers. All peers that receive the query message process the query by using their local information. To process the continuous range query, peers compute the relative vectors of the query peer.

Figure 9 shows how the proposed scheme computes the relative vectors. The peers know the vector of the query peer. For example, a vector between query peer q and N_0 is $(qv_x - nv_x, qv_y - nv_y)$. nv_x and nv_y are the movement vectors x and y of the peers such as N_0 . The peer determines the candidate peers by the relative vector and *expiretime* of query.

Figure 10 shows how the proposed scheme determines the candidate peers. The proposed scheme determines the candidate peers by effective range and dq . dq is minimum distance between peers and linear movement of relative vectors. For example, N_0 is the candidate peer of this range query because N_0 can enter an effective range as shown in Fig. 10. However, N_1 is not the candidate peer of this range query. As a result, the candidate peer such as N_0 needs to be monitored to process the continuous range query.

Figure 11 shows the proposed candidate peers determining algorithm. The inputs are query peer q and peer set $N.queue$. The output is *CANDIDATE_SET*. First, when a peer receives the query message, the peer loads the information of q (Line 01). Second, the peer calculates the relative vectors v_x and v_y (Line 04). Third, the peer calculates the distance between l and n using the relative vectors (Lines 05–06). Finally, the peer compares d_q and q_r . If $d_q < q_r$, the peer sends the message that this peer is candidate to query peer q (Line 8).

3.3 Query update

The naïve scheme that processes the range query at all the times periodically is not efficient. It spends much cost to update the query result. Therefore, most of the existing

Fig. 9 Computation of the relative vectors

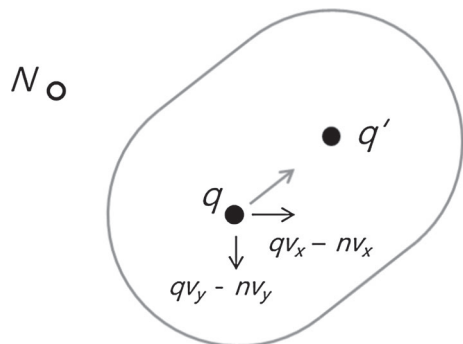
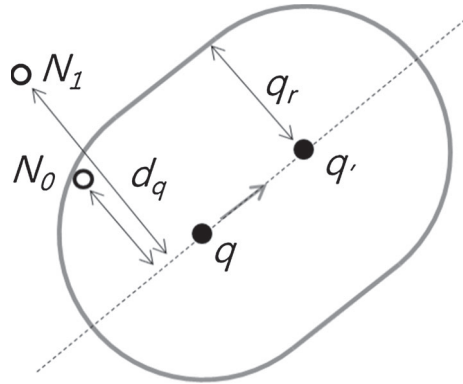


Fig. 10 Determination of the candidate peers



DetermineCandidatePeers(q, N.queue)

Input : query peer q , peer set $N.queue$

Output : the set of candidate peers, $CANDIDATE_SET$

```

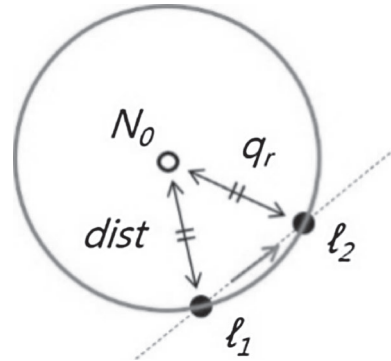
01: load the query information from query peer  $q$ ;
02: while ( $N.queue$  is not empty) {
03:    $n =$  a peer in  $N.queue$ ;
04:    $v_x = q_{vx} - n_{vx}$ ;  $v_y = q_{vy} - n_{vy}$ ;
05:   line  $l \rightarrow v_y = av_x + b$ ;           //  $l$  is the line that include  $q$  and  $q'$  in fig10
06:    $d_q = \text{CalculatDistance}(n, l)$ ;     // Equation to calculate the distance
07:   if ( $d_q < q_r$ ) {
08:      $CANDIDATE\_SET \leftarrow n$ ;       // Send the message to query peer
09:   }
10: }
```

Fig. 11 The candidate peers determining algorithm

methods make safe regions such as Safe Period, inner bounding rectangle, and outer bounding rectangle.

Figure 12 shows how to update the query result. In order to update the query result in the proposed scheme, N_0 draws the monitoring range as shown in Fig. 11. The monitoring range is drawn by the location of N_0 and the query range q_r . For example, N_0 can be included in the query result. When the query peer q is at location ℓ_1 , N_0 enters the monitoring range. This means that N_0 is included in the range query result. When the query peer is at location ℓ_2 , N_0 leaves the monitoring range soon. This means that N_0 is removed in the range query result. To compute the intersected location, we use the equation of the circle and the linear equation. By using the monitoring phase, the proposed scheme updates the query result incrementally.

Figure 13 shows the proposed result update algorithm. The inputs are query peer q and peer n . The output is $RESULT_SET$. First, when the peer is determined to a candidate peer, the peer draws the monitoring range using q_r . And the peer calculates

Fig. 12 Query result update

UpdateRangeQueryResult(q, n)

Input : query peer q , peer $n \leftarrow \text{CANDIDATE_SET}$

Output : the result of a continuous range query, RESULT_SET

01: $\ell[i] = \text{drawMonitoringRange}(q, n)$

02: $qt = \text{computeQueryTime}(q)$;

03: *while* ($qt > \text{current_time}$) {

04: *for* ($j=0; j<i; i++$) {

05: *if* ($\text{current_time} == \ell[j]$) {

06: $\text{updateResult}(\text{RESULT_SET})$;

07: }

08: }

09: }

Fig. 13 The result update algorithm

the intersection time between monitoring range and line l (Line 01). Next, the peer monitors the time. When current time reaches intersection time, the result of range query is updated (Lines 03–09).

4 Performance evaluation

Performance evaluation was conducted on the desktop using the Windows7 operating system, Pentium 3.0GHz processor, and 4GB main memory. Table 1 shows the performance parameters in this performance evaluation. In these environments, peers locate in a two-dimensional space that has 1024×1024 coordinates. The number of peers is 10,000, and the number of queries is 1000. The communication range is 100. In performance evaluation, we compare the number of messages of peers. The number of messages means the number of accessed peers when a query is processed. The attribute values of peers and queries are randomly generated. In order to show the efficiency of the proposed scheme, we compare it with one of the existing schemes,

Table 1 Performance parameters

| Parameters | Value |
|---------------------|-----------------|
| Number of peers | 10,000 |
| Number of queries | 1000 |
| Size of space | 1024 m × 1024 m |
| Communication range | 100 m |

ExtRange in terms of the number of messages. The other existing schemes have high communication costs because peers send the whole information for query processing in these existing schemes. So, we compare the performance of the proposed scheme with only Extrange.

Figure 14 shows the performance comparison according to the query speed. The query speed means the average moving speed of the query peer. The query speed changes from 0 to 30 m/s. As a result, the proposed scheme achieves about 396% better performance than ExtRange. In low query speed environments, ExtRange has poor performance because SP and WT are calculated by the maximum speed of the peers. This means that SP and WT are not suitable in low query speed environments because the result of query is unnecessarily often updated. Therefore, ExtRange achieves better performance when the query speed increases. However, even in the maximum query speed environments, the proposed scheme still has better performance than ExtRange. We confirm that there is no reversal phenomenon of performance in results of extended evaluations according to the increase in speed.

Figure 15 shows the performance comparison according to the peer speed. The peer speed means the maximum moving speed of a peer. The peer speed changes from 10 to 70 m/s. As a result, the proposed scheme achieves about 161% better performance than ExtRange. In the low peers speed environments, ExtRange has poor performance because SP is very small. Therefore, the result of the query is updated often around the extended range. By the same token, ExtRange achieves better performance when the speed of peers increases. However, even in the maximum peers speed environments, the proposed scheme still achieves better performance than ExtRange. We confirm that there is no reversal phenomenon of performance in results of extended evaluations according to the increase in speed.

Figure 16 shows the performance comparison according to the expiretime. The expiretime means the average expiretime of query. As a result, the proposed scheme achieves about 179% better performance than ExtRange. In the high expiretime environments, ExtRange has poor performance because WT is very small. Therefore, ExtRange often reprocesses the query. The proposed scheme and ExtRange have poor performance when the expiretime increases. However, the proposed scheme has better performance than ExtRange regardless of the *expiretime*. We confirm that there is no reversal phenomenon of performance in results of extended evaluations according to the increase in speed.

Figure 17 shows the performance comparison according to the number of queries. As a result, the proposed scheme achieves about 400% better performance than ExtRange. With increasing the number of queries, the number of messages is increased

Fig. 14 Performance comparison according to speed of a query peer

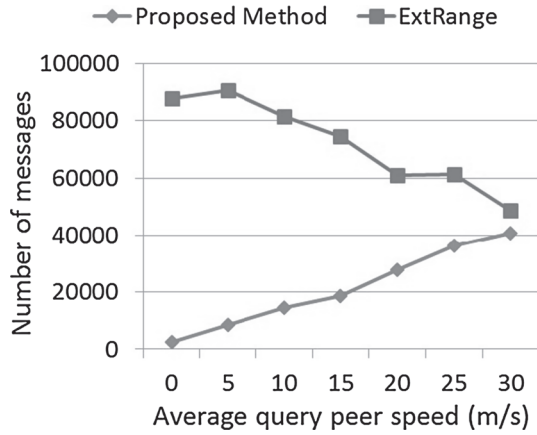
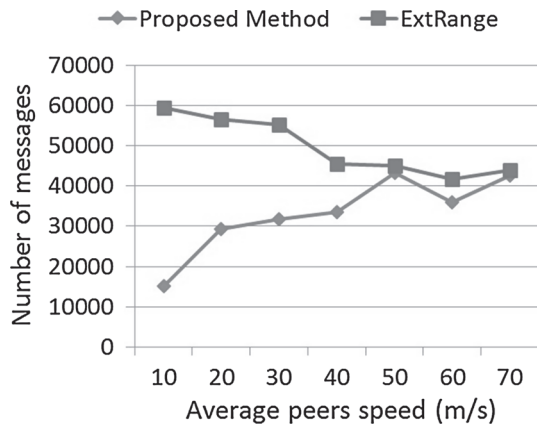


Fig. 15 Performance comparison according to average speed of peers



in the proposed scheme and ExtRange. As shown in Fig. 15, the proposed scheme achieves better performance because the proposed scheme has lower number of messages of peers in initial query processing and query update.

Figure 18 shows the performance comparison according to the number of peers. As a result, the proposed scheme achieves about 500% better performance than ExtRange. With increasing the number of peers, the number of messages is increased in the proposed scheme and ExtRange. Increasing the number of peers means that peers are more densely located in limited spaces. As shown in Fig. 16, the proposed scheme achieves better performance because ExtRange has more duplicated messages in high density peers environments. On the other hand, the proposed scheme has lower duplicated messages in high density peers environments because the proposed scheme uses the TDP in query distribute step.

Figure 19 shows the performance comparison according to the communication range. As a result, the proposed scheme achieves about 650% better performance than ExtRange. With increasing the communication range, the number of messages

Fig. 16 Performance comparison according to the valid time

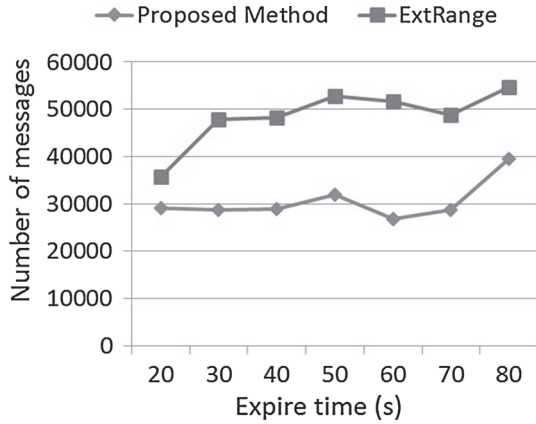
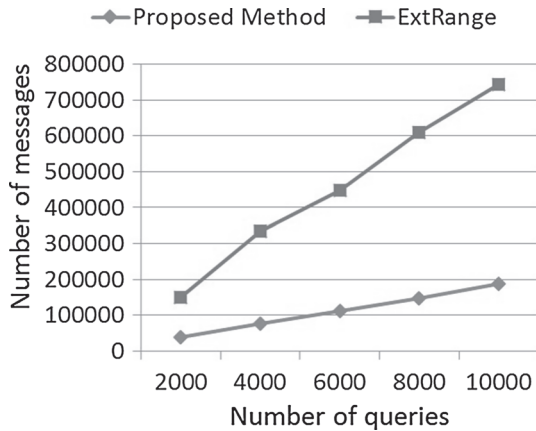


Fig. 17 Performance comparison according to the number of query



is increased in the proposed scheme and ExtRange. Increasing the communication range means that there are more receive peers when a peer sends a message. As shown in Fig. 19, the proposed scheme achieves better performance because ExtRange has more duplicated messages in large communication range environments. On the other hand, the proposed scheme has lower duplicated messages in large communication range environments because the proposed scheme uses the TDP in query distribute step.

5 Conclusion

In this paper, we have proposed a new efficient continuous range query processing scheme in mobile P2P network environments. The proposed scheme consists of two phases. In the query distribution phase, it prunes the peers that are impossible to be included in the query result. In the monitoring phase, the proposed scheme updates

Fig. 18 Performance comparison according to the number of peers

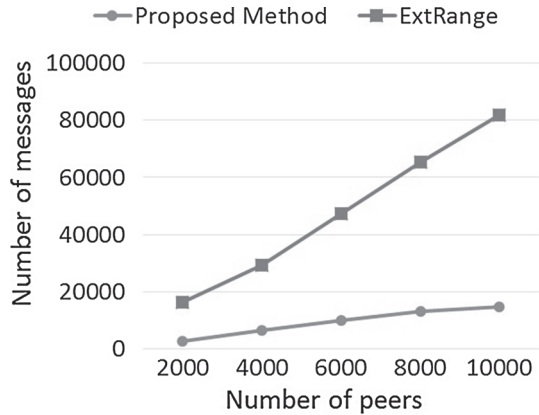
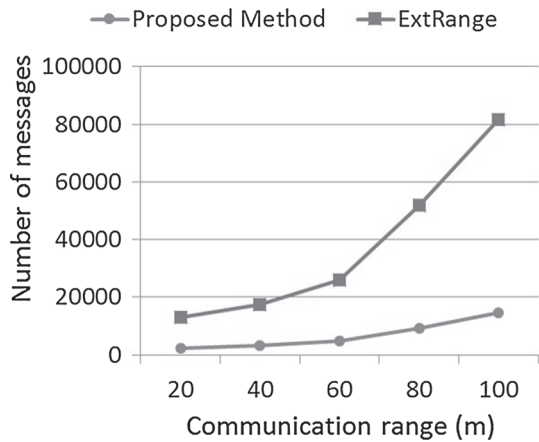


Fig. 19 Performance comparison according to the communication range



the query result incrementally. It was shown through performance evaluation that the proposed scheme outperforms the existing scheme in terms of the number of messages. In our future work, we will study the various query processing methods such as KNN, top- K , and skyline query in mobile P2P networks.

Acknowledgements This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2016-H8501-16-1013) supervised by the IITP (Institute for Information and communication Technology Promotion) and the National Research Foundation of Korea (NRF) Grant funded by the Korea Government (MSIP) (No. 2016R1A2B3007527).

References

- Schiller J, Voisard A (2004) Location based services. Morgan Kaufmann Publishers Inc., San Francisco
- Wu KL, Chen SK, Yu PS (2005) On incremental processing of continual range queries for location-aware services and applications. In: Proceedings of the Annual International Conference on Mobile and Ubiquitous Systems, pp 261–269

3. Mouratidis K, Hadjieleftheriou M, Papadias D (2005) Conceptual partitioning: an efficient method for continuous nearest neighbor monitoring. In: Proceedings of the ACM Conference on Management of Data, pp 634–645
4. Ilyas IF, Aref WG, Elmagarmid AK (2003) Supporting top-*k* join queries in relational databases. In: Very Large DataBase (VLDB), pp 1–12
5. Borzsonyi S, Kossmann D, Stocker K (2001) The skyline operator. In: Proceedings of the IEEE International Conference on Data Engineering, pp 1–20
6. Do TT, Hua KA, Lin C (2009) ExtRange: continuous moving range queries in mobile peer-to-peer networks. In: Proceedings of the International Conference on Mobile Data Management(MDM), pp 317–322
7. Galdames P, Kim K, Cai Y (2010) A generic platform for efficient processing of spatial monitoring queries in mobile peer-to-peer networks. In: Proceedings of the International Conference on Mobile Data Management (MDM), pp 1–10
8. Lou W, Wu J (2002) On reducing broadcast redundancy in ad hoc wireless networks. *IEEE Trans Mob Comput* 1(2):111–123
9. Scholl M, Thielliez M, Voisard A (2005) Location-based mobile querying in peer-to-peer networks. In: Proceedings of the OTM 2005 Workshop on Context-Aware Mobile Systems, pp 1–10
10. Ahmed DT, Shirmohammadi S (2007) Design issues of peer-to-peer systems for wireless ad hoc networks. In: Proceedings of the IEEE International Conference on Networking (ICN), pp 26–31
11. Hara T (2010) Energy efficient data access in mobile ad hoc networks. In: Proceedings of the International Conference on Mobile Data Management (MDM), pp 326–330
12. Bae J, Lee S, Jung E, Kim S (2008) A mobile peer-to-peer query in a social network. In: Proceedings of the International Conference on Advanced Language Processing and Web Information Technology (ALPIT), pp 450–453
13. Huang B, Chen H (2009) Popularity based mobile peer-to-peer search mechanism. In: Proceedings of the International Conference on Wireless Communications Networking and Mobile Computing (WiCom), pp 1–4
14. Ma Y, Jamalipour A (2009) An epidemic P2P content search mechanism for intermittently connected mobile ad hoc networks. In: Proceedings of the IEEE Conference on Global Telecommunications (GLOBECOM), pp 697–702
15. Xu B, Vafae F, Wolfson O (2009) In-network query processing in mobile P2P databases. In: Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS), pp 207–216
16. Chow CY, Mokbel MF, Leong HV (2011) On efficient and scalable support of continuous queries in mobile peer-to-peer environments. *IEEE Trans Mob Comput* 10(10):1473–1487
17. Koldehove B, Ottenwalder B, Rothermel K, Ramachandran U (2012) Moving range queries in distributed complex event processing. In: Proceedings of the In Distributed Event-Based Systems, pp 201–212
18. Zhu Q, Lee DL, Lee WC (2011) Collaborative caching for spatial queries in mobile P2P networks. In: Proceedings of the IEEE International Conference on Data Engineering, pp 279–290