Monitoring Cost Reduction in Sensor Networks using Proximity Queries

Md Rakibul Haque
Dept. of Computer Science & Engineering, University of Asia Pacific, Bangladesh
Email: rakibul_haq@yahoo.com

Mahmuda Naznin
Dept. of Computer Science & Engineering, BUET, Bangladesh
Email: mahmudanaznin@cse.buet.ac.bd

Abstract—Event detection and notification is a common task in a Wireless Sensor Networks (WSN). Efficient data aggregation and the minimization of energy consumption are the great research challenges in WSN. In WSN, normally the events are detected by more than one node and it is more reliable if the consistent event information is received from more than one node. Therefore, aggregated event information is more important than individual event information for energy saving and reliability. Proximity queries can be used to reduce the complexity of data aggregation and energy consumption. This paper presents an efficient and scalable hybrid framework for processing spatial and temporal proximity queries in WSN which we call Spatial and Temporal Processing (STP). STP builds tree structure with less overhead and reduces the event propagation cost through proximity queries. STP reduces energy consumption by reducing the number of aggregator nodes, which ultimately increases the network life time. STP eliminates the unnecessary aggregation of events using a tunable temporal proximity threshold. We compare STP’s performance with another spatial query processing method and we show that STP performs better.

Index Terms—Aggregation, proximity, sensor network, spatial and temporal query.

I. INTRODUCTION

Advances in wireless technology, micro-fabrication and integration, embedded microprocessors, ad-hoc nature and easy deployment have established sensor network as a very popular network in commercial and military applications. One of the major applications of sensor network is monitoring task. An event is generated when a particular condition is satisfied by a sensor node. Proximity query allows reporting events those are observed by sensor nodes placed within a certain distance from each other. We can reduce the unnecessary event notifications to the base station and increase the network life time by in-network aggregation of the events through proximity queries. This paper presents STP (Spatial and Temporal Proximity query processor), which can efficiently aggregate proximity events and provide alarms to the base station with low energy consumption. STP provides an aggregation mechanism, which selects small number of aggregator nodes based on the spatial and temporal proximity of the nodes.

Combining the proximity events carefully, STP can simultaneously handle a large number of proximity queries without flooding the large portion of the network, which ultimately keeps sensors less busy and saves the energy of the sensors.

The rest of the paper is organized as follows. Section II presents related work. We define the spatial and temporal proximity query in Section III. The framework of STP is explained in Section IV. In Section V, we present the performance evaluation of STP and compare with another proximity query processing method [21]. Finally, we conclude and outline our future research goals in Section VI.

II. RELATED WORK

One of the fundamental issues in proximity query processing is data aggregation. Due to inherent redundancy in raw data collected from sensors, data aggregation can reduce communication cost by eliminating redundancy and forwarding only the useful information extracted from the raw data. For these reasons, it is crucial for a sensor network to support in-network data aggregation [8]. Various aggregation approaches ([3], [8], [10], [22] and [19]) have been proposed for data gathering applications and event raised applications. Most of these approaches use tree based structure for aggregation and utilize the multi-hop communication links to reduce the computation of expensive queries ([15], [16]). Some methods try to reduce cost of data aggregation using probabilistic techniques [18] or decentralized algorithms ([6], [10] and [11]). Some research proposals ([2], [7], [4] and [14]) advance the in-network data aggregation mechanism. None of these research works considered the concept of proximity query.

The concept of ‘proximity query’ was introduced in [21], which does not consider the timing of event generation for event aggregation and send unnecessary alarms to the base node when the event generating target does not move rapidly. Proximity alarm is unnecessary if the events X and Y are aggregated at node z after a long time or the time difference between the detection of X and Y is very high. For example, in a battle field application.
proximity alarm should be sent to the base node when an enemy and an army cross two locations n1 and n2, respectively at nearly same time where n1 and n2 are within proximity spatial threshold. The alarm should not be sent if the time difference between the detection of the two events is very high or the events information is aggregated after a long time. In these cases, temporal proximity threshold can solve the problem which is not considered by Y. Kotidis [21]. Again, an enemy may stay at a location for long time (few minutes). In this case, unnecessary alarms will be sent to the base node by the RI (Routing Index) method [21].

In this paper, we redefine the concept of ‘proximity query’ considering the ‘temporal proximity’ and propose aggregation mechanism for reducing the unnecessary alarms to the base node for both rapidly and slowly moving targets.

III. PROXIMITY QUERY

Given a set of predefined event types \( \Psi = \{A, B, C, \ldots\} \), proximity query [21] can be defined as \( Q = (X, Y, d, t) \) where \( X \) and \( Y \) are members of \( \Psi \), and \( d, t \) is the spatial and temporal proximity threshold, respectively. When two events \( X \) and \( Y \) are occurred within a distance \( d \) (hops) and within the time \( t \), then a proximity event is generated and the base station should be alarmed. The idea can be extended to multiple events scenario. A set of proximity queries is registered on the sensor nodes from the base station. When a node detects or hears an event, it propagates the event information to its neighbor nodes with a hope for potential proximity event in the other nodes of the network. Figure 1 shows a scenario where a proximity event is generated due to two events \( X \) at node \( n_1 \) and \( Y \) at node \( n_2 \), where the distance (hop) between \( n_1 \) and \( n_2 \) is not greater than the defined proximity threshold, \( d \). The potential aggregator nodes hear about the events within very short time after events (event \( X \) and \( Y \)) detection propagation by the source nodes \( n_1 \) and \( n_2 \). By ‘aggregator’, we refer to the node which is selected to send the proximity event to the base node.

IV. FRAMEWORK OF STP

In this section, we have explained the framework of STP. The framework of STP consists of two main phases: query registration and event propagation. The first phase is responsible for establishing the ‘routing structure’ which is used in the second phase. By ‘routing structure’, we refer to the tree formed in the network in ‘query registration’ phase. We have used ‘routing structure’ with ‘inverse routing tree’ interchangeably. Figure 2 shows an inverse routing tree formed in the network. We assume that, the sensor network architecture has one base station which may be a powerful sensor node, or may be located outside of the sensor network that can communicate with a subset of the sensors in the network.

A. Query Registration

The base station acts as an initiator of the proximity query processing. The user registers the proximity queries in the base station. The base station informs all the nodes in the sensor network about these queries. The simplest approach is to broadcast the packet to the nodes in the sensor network. This approach leads to flooding which will increase unnecessary message transmissions and reduce the energy of the sensors. To minimize the impact of broadcasting, STP adopts random waiting time [9]. It is a simple technique, for achieving temporal convergence, in which when a node receives a packet it waits for a random period of time. During this time period, if it receives another query packet, it combines both queries into a single packet and forwards the combined packet to other nodes. In order to further reduce the impact of broadcasting, if a node receives a packet more than once, it discards the later packets. In this approach, queries will be reached to all of the nodes in the network.

Each query packet contains proximity events, distance threshold, hop count and time threshold. If a node receives query packet from multiple neighbors, it should select one of them as a next hop to reach to the base station. The hop count (hp) is used as metric for this purpose. This process forms an inverse routing tree which is depicted in Figure 2. In inverse routing tree, a node in the
network knows its next neighbor for forwarding a message to the base node in the shortest path. This inverse routing tree is used for sending proximity alarm.

B. Event Propagation

Algorithm 1 processEvent(\(X, n_1, t_X, h_p\))

Parameters:
\(X\): Event needs to be processed  
\(n_1\): Source node detecting the event \(X\)  
\(t_X\): Detection time of \(X\)  
\(h_p\): Hop counter

Begin
Update eventHistory
If hasSeen \((X, n_1, t_X, h_p)\) Then Return
End If
If \(h_p > 0\) Then Broadcast eventDetected Message \((X, n_1, t_X, h_p, -1)\)
End If
For all event \(y\) in the eventHistory do
If there is a proximity query like \(Q(X, y, d_{XY}, t_0)\) Then
If distance \((n_1, n_2) \leq d_{XY}\) and \(|t_X - t_Y| \leq t_0\) and \(|t_Y - min (t_X, t_Y)| \leq t_0\) Then Raise Proximity Event \(PE(X, n_1, y, n_2)\)
End if
End if
End if
End for
End

Here, we assume that target generating events does not move rapidly from one place to another in the network in some applications like environment monitoring, border surveillance, fire or pollution detection, etc. When a node detects an event, normally, it informs the base station about the event. In general, events are propagated independently to the base station. The number of event notification to the base station is reduced by the processing of proximity queries efficiently. STP takes the advantages of in-network aggregation of events which requires less effort (in terms of bandwidth, battery power, etc.) than the general method. When a node detects an event, it sends an ‘EventDetected’ message to its neighbors hoping that the event would be aggregated in other nodes. An ‘EventDetected’ message contains about the event information, source node, event time and hop counter \(h_p\), etc. For limiting the event propagation, hop count \(h_p\) can be used as a bound. The source node initializes the \(h_p\) for each event \(y\) registered with event \(x\) as follows:

\[
h_p = \max\left(\frac{d_{XY}}{2}\right)
\]

When another node receives the ‘EventDetected’ message, it runs the process specified in Algorithm 1. The node updates its event history with the received message and checks whether the event will cause aggregation or not. If the \(h_p\) received in the message is not zero, it decreases the \(h_p\) by one and transmits the message to its neighbors. For optimizing the number of messages, Random-Walk [5] or modified-BFS [20] can be used.

C. Event History

We assumed that, each node maintains a cache, ‘EventHistory’. Each entry of the cache corresponds to a proximity event detected or heard from neighboring nodes. Each node refreshes its cache after a certain period of time. For example, consider the following proximity queries \((X, Y, d_1, t_1), (X, W, d_2, t_2), (Y, Z, d_3, t_3), (X, Z, d_4, t_4)\) and in the network, all nodes receive above proximity queries from the base station. If a node receives an event \(X\), then it stores this event information for the amount of time that is the maximum of the threshold times among the registered queries where event \(X\) is involved. We call this time as \(T_{\text{refresh},X}\) for \(X\) and it is calculated as follows:

\[
T_{\text{refresh},X} = \max(t_1, t_2, t_3, t_4)
\]  (2)

D. Aggregation

In this section, we describe the mechanism of event aggregation. For example, a node \(n_1\) detects an event \(X\) and propagates the ‘EventDetected’ message to its neighbors. Similarly, node \(n_2\) detects an event \(Y\) and it also propagates the message. These two messages may meet each other at some nodes in the network. From Figure 1, the nodes within the bounded area may be potential aggregator if the event detection time within the temporal proximity threshold. To become a potential aggregator

\[
\text{Distance} (n_1, n_2) \leq d_{XY}
\]  (3)

\[
|t_x - t_y| \leq t_{xy}
\]  (4)

\[
|t_x - \min (t_x, t_y)| \leq t_0
\]  (5)

Equations (3) and (4) are required to meet the conditions of proximity query, and (5) is used to avoid the unnecessary proximity alarms. In (5), \(t_0\) is the current time at the potential aggregator node and \(t_0\) is the time threshold, which refers to the maximum allowed time for event propagation.

E. Selection of Aggregator

STP selects aggregator efficiently to reduce the redundant proximity alarms to the base station which causes less energy, bandwidth consumption than RI method [21]. The aggregator selection problem is similar to the leader selection problem in WSN. There are many research ([13], [12]) are performed to resolve this problem. We adopt the leader selection mechanism from [12] to select aggregator node efficiently. In [12], leaders are selected efficiently based on the some parameters like available energy, number of neighbors, distance from the source and base node, etc. The rotation of aggregator node among the potential aggregator nodes can be used to save the battery power of the nodes and proper load distribution.

F. Sending Proximity Alarm

The selected aggregator nodes send the proximity alarm to the base node using the inverse routing tree which is depicted in Figure 2. The aggregator node sup-
presses the same proximity event for a small amount of
time to avoid the redundant proximity alarms.

G. Route Maintenance

As long as the inverse routing tree is maintained, ag-
ggregator node can easily send alert to the base station. The
inverse routing tree should be fault resilience to the
node failure. If a node in the inverse routing tree fails, the
ancestor in the failed path to the base station can easily
determine the link failure through periodic neighbor dis-
covering protocol. To repair the link, the ancestor broad-
casts a packet requesting for an alternative path. When a
neighboring node receives the packet it sends an Ac-
knowledgement packet to the ancestor node. It may hap-
pen that several nodes having the routes to the base sta-
tion send the acknowledgement to the ancestor. The an-
cestor will selects the node which has the minimum hop
count distance to the base station. Figure 3 describes the
link failure and alternative path establishment procedure.

H. Rapidly Movement of Event Sources

This section explains the proximity query processing
 technique when event generating target moves rapidly. Some research activities ([24], [25] and [23]) are per-
formed to track moving targets. When a target moves,
based on its direction and speed, an aggregation tree is
formed in STP like E-SPAN [1] (shown in Figure 4).

Figure 4 shows an event propagation scenario based
on the direction and speed of event sources. The node
detecting target, identifies the direction and speed of the
source and propagates ‘EventDetected’ messages only
towards the direction of target, instead of forwarding to
all of its neighbors as explained in Section IV-B. Similarly,
node detecting the target1 propagates the event to-
wards the direction of target1. Therefore, the event infor-
mation is aggregated on the nodes towards the direction
of target1 and target2. If the target, moves away from the
base station, the event message is propagated to its direc-
tion. If the event is aggregated with the event generated
by target2 in some nodes, then they will try to become
aggregator. If the two targets are in opposite sides of base
station and moves towards base station, the event infor-
mation is aggregated in some nodes nearby base station
only if the targets are in spatial threshold distance. In all
the cases, the number of potential aggregator nodes be-
comes less and ultimately, reduces the cost for informing
the base station.

V. SIMULATION RESULTS AND EVALUATION

In this section, we compare the performance of STP
with RI method [21] with the simulation results obtained
by simulation. In simulation, we have varied the spatial
and temporal proximity threshold and network sizes to
compare the total cost and number of aggregator nodes.
The total cost [21] is calculated as follows:
Total Cost = 1.41 x (No. of transmitted messages)
+ (No. of received messages)
+ (No. of idle listened messages)
The total cost includes all the cost handling each event
in the network except the cost for query registration
phase.
A. Simulation Setup

We have developed a simulator using Java to compare our method STP to RI method on various aspects. In this experiment, we have modeled the network as two dimensional $(n \times n)$ grids for simplicity. We assume that, all sensor nodes have similar properties.

B. Effect of Spatial Proximity

We have measured the impact of varied spatial proximity threshold on the total cost which is depicted in Figure 5(a). We consider the hop distance between nodes as the spatial proximity. From Figure 5(a), it is evident that total cost increases as the spatial proximity threshold increases in both STP and RI method. The total cost increases because, the event propagation cost and the number of aggregator nodes increases when the spatial proximity increases. The rate of increase of total cost in our STP method is less than the rate of increase in RI method [21]. The reason behind this is that STP selects aggregator node (explained in section IV-E) among the potential aggregator nodes efficiently and the number of selected aggregator nodes is less than the number of aggregators in RI method (depicted in Figure 5(b)).

C. Effect of Temporal Proximity

We have also measured the effect of temporal proximity threshold on the total cost and number of aggregator nodes which are depicted in Figure 6(a) and Figure 6(b), respectively. From these two figures, it is observed that the total cost and number of aggregator in RI method is almost constant since RI method does not consider the concept of temporal proximity. In STP, the total cost and the number of aggregator increase slightly as the temporal proximity threshold increases, and become constant for higher value of proximity threshold. The reason behind this is that the number of proximity events and number of potential aggregator node increases when the temporal proximity threshold increases (see Equation 4). From Figure 6(a) and Figure 6(b), it is evident that the total cost and the number of aggregator node is less in STP than those in RI method.

D. Network Size

We have estimated the total cost in the networks of different sizes varied from $(5 \times 5)$ to $(25 \times 25)$ nodes, which is depicted in Figure 7. From this figure, it is observed that the rate of increase of total cost is less in STP than in RI method. The reason behind is that the number of aggregator nodes increases more in RI method when the network size increases.
E. Number of Neighbors

We varied the number of neighbors of each node from four to twenty and measured the total cost. From Figure 8, it is observed that the total cost in RI method increases rapidly and in STP increases slightly when the number of neighbors increases since the event propagation cost and the number of aggregator node increases when the number of neighbors increases.

F. Effect of Monitoring Time

In our simulation, we measure the total cost varying the monitoring time of an event source (not rapidly moving) from 30 sec to 5 minutes, which is depicted in Figure 9. By ‘monitoring time’, we refer to the time duration for which an event source is detected by the same sensor node. From Figure 9, it is observed that the total cost in RI method increases rapidly and in STP increases slightly when the monitoring time increases. The reason behind is that the same selected aggregator nodes route the proximity event and uses the same paths to the base station in STP.

G. Number of Concurrent Queries

We compared the performance of STP and RI method for varied number of concurrent proximity queries from ten to fifty which is depicted in Figure 10. It is observed from this figure that the total cost increases slightly with as concurrent queries increases in both STP and RI method. However, the cost is less in STP because the number of aggregator nodes in STP is smaller than in RI method.

H. Effect of Target Movements

We have also measured the total cost changing the speed of the targets. In our simulation, we assumed the spatial and temporal proximity thresholds as 3 hops and 100 ms, respectively. We varied the speed of the targets from 10 hops/ms to 50 hops/ms for both the RI and STP methods and measure the total costs, which are plotted in Figure 11.

From Figure 11, it is observed that cost in STP increases slowly when target moves slowly and increases dramatically for high speed of target movement. The reason behind this is that when target moves faster, the formed aggregation tree (see Section IV-H) out weight the benefit in STP. In RI method, cost grows slowly when speed increases. From Figure 11, it is evident that STP performs better than RI method when target moves slowly.
VI. CONCLUSION AND FUTURE WORK

In this paper, we have presented STP, a framework for in-network aggregation of events through proximity queries in a sensor network where aggregated event information is more important than the individual event information. Our method STP reduces the cost for query registration and event propagation. STP eliminates unnecessary proximity events through temporal proximity threshold. STP reduces the cost for sending proximity alarms to the base node by selecting small number of aggregator nodes. Simulation results show that STP performs better than another popular spatial query processing method called RI [21].

In this paper, we have implemented STP and RI methods assuming the network of nxn grids and similar type sensor nodes. We plan to measure the effect of concurrent events in STP and compare the performance of STP with other aggregation methods like E-SPAN [1] using NS2 [17]. We desire to extend STP for specific applications (such as environment monitoring, border surveillance, fire or pollution detection) of sensor networks where the proximity query is efficiently applicable for in-network aggregations.

REFERENCES


Md Rakibul Haque received BSc. degree in Computer Science & Engineering from Military Institute of Science and Technology (affiliated with University of Dhaka), Dhaka, Bangladesh in 2005. He received MSc. degree in Computer Science & Engineering from Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh in 2009. He is currently an Assistant Professor of Computer Science & Engineering at the University of Asia Pacific, Dhaka, Bangladesh. His research interests include wireless sensor networks, wide area service discovery and content sharing peer-to-peer networks with focus on search flexibility, efficiency and robustness.

Mahmuda Naznin received BSc. and MSc. degrees in Computer Science & Engineering from Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh in 1996 and 2000, respectively. She received PhD. degree in Computer Science from North Dakota State University, USA in 2007. She is currently an Associate Professor of Computer Science & Engineering at BUET, Dhaka, Bangladesh. She has served as a reviewer for many international conferences and journals. Her research interests include wireless ad-hoc and sensor networks, graph algorithms, combinatorial algorithms and bioinformatics.