A Query Processing Method in Multi-user Scenarios

Xingcheng Liu, Zaili Ma, Xutao Cao, Shaochong Liu
Department of Electronic and Communications Engineering
Sun Yat-sen University
Guangzhou, Guangdong, China
E-mail: isslxc@mail.sysu.edu.cn

***Abstract* – An application scenario, the multi-users scenario in wireless sensor networks (WSNs), was defined in this article, where a large number of queries were generally sent from thousands of users in a large-scale monitored region and they would be a very heavy load for query processing. To deal with this multi-user scenario, the Network Event Report (NER) query processing method is proposed. With this method, the queries are sent to the exact event regions based on the information reported by the event reporting mechanism employed in the NER. In order to restrict the distribution of queries within the even region, a special subnet, the embedded network (EN), is employed in the NER. Simulation results show that the proposed NER approach can work effectively in multi-users scenario.**

Keywords- wireless sensor networks; query processing; network events report (NER); embedded networks

I. INTRODUCTION

Wireless Sensor Network (WSN) is the sensing network comprised of a large number of sensor nodes distributed in monitored regions. As a new type of technology, WSNs have been well applied to security scrutinizing, environmental monitoring, wildfire detection, buildings monitoring, chemicals leak warning, medical care assistance, and many other fields [1].

In this paper we will define an application scenario for WSN, the multi-user scenario, in which the applications provide services to the public. In the multi-user scenario, the users are referred to those who just focus on their interests but know little about the deployment of the network. And, their queries are event-based, where the event is referred to the combination of specific values of several inter-related physical properties. And users query the network by the event names to determine whether the events are in the network and where they are in the network.

To give an example of multi-user scenario, we assume that there is a wide land to be monitored and many people who are interested in it. For instance, a tremendous number of tourists visit a large national forest park, where there are elephants, monkeys, squirrels, birds, et al.. So, the tourists need to know where the animals are and how many of them there are. Then, they send their queries to the base station (BS) within a network. For a large number of users, one of the benefits from the WSN is that the average cost of per user can be reduced. At the same time, it may result in many technical problems to be solved. In this paper we concentrate on how to process the large number of queries in such application scenario of WSN.

When the WSN provides service to a large number of users, the distribution of queries takes on some characteristics that must be taken into consideration when designing a query processing method. The distinct characteristic of this scenario is large-scale monitored region with a large number of queries. Usually the application provided for a large number of users is performed in a large-scale monitored region. The distinct characteristic is that the number of queries increases as the number of users grows. Moreover, when users have no a priori knowledge about the network, they raise queries just based on their interests instead of on geographic location. This is different from the most cases in the traditional approaches, where the region to be queried is specified. On the contrary, in the case of our interests, the queries just indicate which events and the time periods to be queried.

Many researches have been exploring data query methods in WSN based on spatio-temporal query processing [2], [3] [4], in which users are required to indicate the time attributes and the regions to be queried. However, in multi-user scenario, requesting ordinary users to determine the exact query regions is an inappropriate demand for them. It is more natural for users to query the network directly based on their interests, in which case only the event names and the time periods to be queried are provided, rather than the exact location information. Therefore, the spatio-temporal query processing methods can not work very well because the queries have to be distributed all over the network in this case. As has been discussed before, the traditional approaches can not work well in multi-user scenario. So, it is necessary to propose a new approach to process data query for this scenario. In this paper a suggested query processing method, the Network Events Report (NER), is proposed. Simulations show that it works effectively in the multi-user scenario.

II. NER APPROACH

A. NER Introduction

We designed a special query processing method called NER for multi-user scenario. It mainly focuses on the problems aroused by the large-scale monitored region with a large number of queries.

For a large-scale network, it is not reasonable to distribute queries to the entire network. To make the distribution of queries be restricted in the event region, a distributed self-organizing algorithm called the embedded network algorithm is designed to run in the event regions. In each region, a local

The project was supported by the National Natural Science Foundation of China (60673086, 60970041).

CHINACOM 2010, August 25-27, Beijing, China

DOI 10.4108/chinacom.2010.124

sub-network, called embedded network (EN), can be constructed to cover it. The ENs can adjust their structures to cover the regions as the event regions move or change in shape. So, when queries sent from BS reach the EN, the distributing of queries is restricted in the ENs.

To disseminate query from base station (BS) to the event region exactly, an event reporting mechanism is introduced to solve this problem. In the EN, one or more nodes that have a better position will be selected to report events to the BS. The 'better' position in the algorithm is defined as the node that has a minimal hop count from the BS. In other words, the nodes close to the BS in the event region will be more likely to be selected. This kind of node is designated as the access point of the EN, which reports events of its EN to the BS, with a forward path between BS and EN being created. Along the path, the BS can send queries to the ENs. Other routing protocols [5] [6] can also be employed for BS sending queries to ENs.

Based on the information obtained from the event reporting mechanism, the BS can know what and where the on-going events are in the network. When the queries arrive at the access points along the paths set up before, the queries will be distributed just within the ENs from the access point.

B. Embedded Network Algorithm

In NER, the entire network is assumed to be a fixed tree-type network topology, and the low-level mechanisms of scheduling, time synchronization, and the like, can be ignored or manipulated in other algorithms and protocols. We also assume that nodes in the network know their node ID and the hop count from BS, denoted by *NodeID* and *Hop* respectively. In addition, several events monitored in the network have been defined in advance, denoted by *EventID*.

To cover the event region and update the EN when an event is moving geographically, a distributed algorithm of high energy-efficiency is required. There have been some self-organizing algorithms [7], [8]. However, these algorithms are inapplicable for our application scenario in the WSNs because they are not designed for dealing with the event moving issues. Therefore, we designed an algorithm, Fast Mini-Cost Coverage (FMCC) algorithm, which is specialized for the EN.

FMCC runs in event region, and an event region is divided into a changing region and a static one. The changing region is the one, where there are nodes that detect new events (including the case that no event is detected), while the rest is the static region, where nodes do not detect new events. The FMCC mainly runs in the changing region.

Other four variables are maintained by a node in FMCC, which are defined as follows:

Ufather: the *NodeID* of the parent node in the tree-type topology of its local EN. However, it may not be the parent node in the topology of the entire network, and the default value is the *NodeID* of itself.

Ufather_hop: the *Hop* of its *Ufather*. The default value is the *Hop* the node has.

Retorted: a Boolean variable that indicates whether the node has a child or not in its EN. It will be "True" if the node sends any control messages but the *Quit*, which to be defined later.

Ustatus: the state of node, one of the three defined states, *Speaker*, *Adherent* and *Guarder*.

A node in the FMCC has one of the three different states, which are defined as: *Speaker*, *Adherent* and *Guarder*.

Speaker: the state of the access point of EN. The access point is also the root node of the EN, node of this state is responsible for sending reports of events happened in the EN to BS.

Adherent: the state of the node that has not finally determined its *Ufather*.

Guarder: the state of the node that finally determined its *Ufather*. The node of this state always prevents its neighbors from becoming *Speaker* by sending a *Retort* message that denies the declaration from a candidate *Speaker*.

In the case of running the FMCC, nodes determine their states by comparing *Hop* among their neighbors.

In FMCC there are four control messages defined as follows:

Self-elect: the message that node sends to become a *Speaker* competing with its neighbors, the message contains the sender's *EventID*, *Hop* and *NodeID*.

Speaker-declare: the message that node delivers to confirm its *Speaker* state among its neighbors, the message includes the sender's *EventID*, *Hop* and *NodeID*.

Retort: the message that node sends to reject the *Self-elect* or *Speaker-declare* message to remove the inappropriate *Speaker*, the message contains the sender's *EventID*, *Hop*, *NodeID*, *Ufather* and *Ustatus*.

Quit: the message that the node with children in its EN and without detection of any event sends to quit from its primary EN. The message contains *EventID* and *NodeID*.

The FMCC algorithm is performed in two phases:

In the first phase, the nodes in the changing region, which have not detected any events and have children nodes in their primary EN, send *Quit* message to quit from their primary EN. The children of such nodes become *Speaker* again; the nodes in the changing region that detected new events become *Speaker*, and the *Guarder* nodes in the static region change to *Adherent*. Then, the *Speakers* broadcast *Self-elect* messages after a random back-off delay, in order to apply for the *Speaker* status among their neighbors. Responses from their neighbors are shown in Figure 1. A few *Speakers* in the changing region have been eliminated after this phase, and some of them will be further eliminated in the next phase.

In the second phase, the *Speakers* survived in the first phase broadcast *Speaker-declare* messages to confirm their *Speaker* status among their neighbors. Some of these declarations will be denied by *Retort* messages from their neighbors while others will not be denied, the details of the FMCC are shown in Figure 1.

```
PSEUDOCODE:FMCC
Input: Messages, Nodes Output: Local topology records of the EN
Annotation: All the messages involved have the same EventID; otherwise, the messages will be treated as the Quit message; node cancels its message that will be sent for the same purpose as the messages received;

Ⅰ. Node N received a Quit message M:
If (N.Ufather = M.NodeID)
   N.Ustatus ← Speaker, N clear the EN information about M

Ⅱ. Node N received a Self-elect message M:
Switch (N.Ustatus)
 case Speaker:
   If (N.Ufather_hop > M.Hop)
     N.Ustatus ← Guarder, N.Ufather ← M.NodeID
   Else If (N.Ufather_hop = M.Hop)
     N.Ustatus ← Adherent, N.Ufather ← M.NodeID
 case Adherent:
   If (N.Ufather_hop > M.Hop)
     N.Ufather ← M.NodeID
     If (N.Hop >= M.Hop)
       N.Ustatus ← Guarder
   Else If (N.Ufather_hop = M.Hop && N.Hop >= M.Hop)
     N.Ufather ← M.NodeID
 case Guarder:
   If (N.Ufather != M.NodeID)
     send a Retort message to M.NodeID

Ⅲ. Node N received a Retort message M:
If (N.Ustatus = Speaker && N.Ufather != M.NodeID)
 Switch (M.Ustatus)
   case Adherent:
     If (N.Hop >= M.Hop)
       N.Ustatus ← Adherent, N.Ufather ←M.NodeID
   case Guarder:
     N.Ustatus ← Adherent, N.Ufather ← M.NodeID

Ⅳ. Node N received a Speaker-declare message M:
Switch (N.Ustatus)
 case Speaker: ERROR // impossible condition
 case Adherent:
   If (N.Ufather_Hop >= M.Hop && N.Hop >= M.Hop )
     N.Ufather ← M.NodeID
   Else
     send a Retort message to M.NodeID
 case Guarder:
   send a Retort message to M.NodeID
```

Figure 1. FMCC pseudocode

When the running of the FMCC is completed, in most cases only one tree-type network topology is formed in a event region, which is called the EN, while in other cases more than one such EN might be built. In the later cases, the number of ENs formed depends on the distributions of nodes and the shapes of the event region. The topology of the EN formed is shown in Figure 2. The finally-survived *Speakers* (the root nodes of the topology) become the access points of the EN and send the reports of events to the BS. The *Speaker* reports event to the BS immediately when it becomes an access points, then it reports event periodically until it is not an access point any more for the purpose of keeping "alive", which means that the event is still there.

During the period of constructing an EN, all the nodes that have sent *Self-elect*, *Speaker-declare* and *Retort* messages are the *Ufather* of some nodes, it means that these nodes have children and its *Retorted* is set as "True". When a query is received by a node in EN, the query will be forwarded to its neighbors if the node's *Retorted* is "True". Otherwise, the query will not be forwarded by this node. This kind of nodes form a Connected Dominating Set [7], [8] of the EN, which construct the query forwarding paths, along which the query distribution is restricted within the EN.

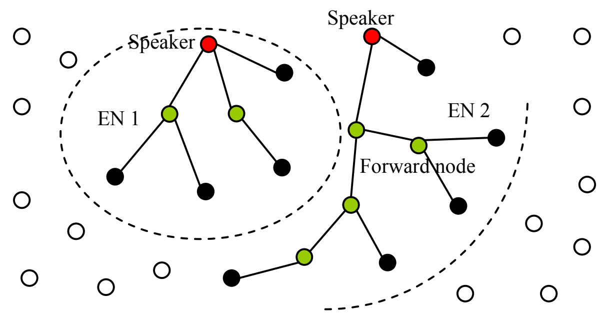

Figure 2. Topology of EN

III. SIMULATION EXPERIMENTS

The above proposed NER is implemented using the Visual C++ 6.0. Some of the experiments are oriented to compare the performance of the suggested approach with that of the other two methods, GHT [9] and FullFlood [10].

We construct a multi-user scenario with the Following sensor network model and parameters:

1. The nodes are uniformly deployed in a 1000m×1000m square region;
2. The BS is located and fixed at the position (0, 0);
3. All sensor nodes are immobile and have the same fixed communications capacity, besides, all communications links are bidirectional;
4. The signal interference in the wireless channel is ignored;
5. Requesting data queries from users is assumed to happen at any time, and events are queried randomly.

TABLE I. SIMULATION PARAMETERS

Parameters	Default Values
Area covered (*Area*)	1000×1000 m^2
Number of sensors	2000
Wireless range	50 m
Simulation Time	1000 round
Event types (*E*)	10
Number of queries	1500
Event region size of one event type	$0 \sim Area/E$ m^2
Event region mobility rate	10 ~ 20 m/round
Data generation rate	1 packets/round
Event reporting period	1 round
Query temporal window	1 round
Data<*Time-stamp*, *Value*>	64 bits
Query<*Sub_queryID*, *EventID*, *Time-stamp*>	128 bits
Event reporting packet<*NodeID*, *EventID*, *Time-stamp*>	128 bits
EN control messages	64 bits

A summary of query and sensor network parameters and their default values used in our experimental evaluation is presented in Table I.

In the data disseminating phase, all the methods route the queried data to the BS without consideration of data fusion or compression, since these issues are not the interests of the paper.

We compare the algorithms in terms of the average energy consumption per network node while processing the same queries. In this simulation the same energy model is employed as that in [11]: $E_{Tx}=\alpha+\gamma\cdot d^n$, $E_{Re}=\beta$, where, E_{Tx} denotes the energy cost to transmit a bit, and E_{Re} is the energy cost to receive a bit. Based on the parameters adopted in [11], we set their values as follows: n=2, γ=10$pJ/bit/m^2$, $\alpha=\beta$=50nJ/bit. The presented results are yielded with the averaged over 10 simulations.

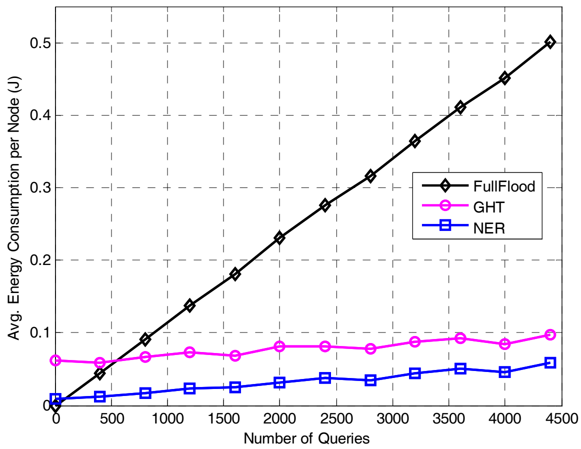

Figure 3. Performance of average energy consumption per node with the number of queries

In this experiment, we increased the number of queries while other parameters are fixed. The average energy consumption is shown in Figure 3, which illustrates how the number of queries affects on the energy consumption for the different algorithms.

From the three curves shown in Fig. 3, we noticed that the average energy consumption for all the three methods increases with the number of queries, which is reasonable since every query consumes energy. However, the FullFlood has the highest growth rate of energy consumption because all queries are flooded to the entire network. As for GHT, the lowest energy consumption growth rate is obtained because the queries are only sent to the specific mapped nodes. However, as for the proposed NER, all queries are guaranteed to be distributed to all the nodes within the event region, so the energy consumption growth is far slower than FullFlood while it is approximately the same as that of the GHT. Moreover, the energy consumption in GHT is more than that in NER because the former transmits all the sensed data to the mapped nodes no matter whether the data are queried or not.

IV. CONCLUSIONS

In this paper, we defined a multi-users application scenario of WSN, and proposed the NER, a query processing method for this scenario. The NER mainly resolved the problem of vast queries that do not indicate the query region in multi-users scenario but the event names. The event reporting mechanism employed in NER reports the on-going events to the BS, indicating what and where the event are in the network. And the embedded network constructed with the FMCC algorithm restricts the distribution of queries within the event region. These make the NER treat the queries in the multi-users scenario effectively. The performance of the proposed NER approach is evaluated by simulation. Compared to other well-known algorithms such as the GHT and FullFlood, the proposed NER query processing method can save at least 50% energy in multi-users scenario with the same network parameters.

REFERENCES

[1] M. Bal, W. Shen, H. Ghenniwa, "Collaborative signal and information processing in wireless sensor networks: A review", In *Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics*, San Antonio, TX, USA, Oct. 2009: 3151- 3156.

[2] A. Coman, M. A. Nascimento , J. Sander, "A framework for spatio-temporal query processing over wireless sensor networks", In *Proceeedings of the 1st International Workshop on Data Management for Sensor Networks: in conjunction with VLDB 2004*, Toronto, Canada, 2004: 104−110.

[3] A. Coman, J. Sander, M. A. Nascimento, "An analysis of spatio-temporal query processing in sensor networks", In *Proceedings of the 21st International Conference on Data Engineering Workshops*, Tokyo, Japan, Apr. 2005: 1190.

[4] N. Jain, R. Biswas, N. Nandiraju, D.P. Agrawal, "Energy aware routing for spatio-temporal queries in sensor networks", In *IEEE Wireless Communications and Networking Conference*, New Orleans, LA, USA, Mar. 2005, 3: 1860–1866.

[5] B. Karp, H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks", In *Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking*, Boston, Massachusetts, U. S., 2000: 243-254.

[6] X. Bian, X. Liu, H. Cho, "Study on a Cluster-Chain Routing Protocol in Wireless Sensor Networks," In *the third international conference on communications and networking (ChinaCom)*, Hangzhou, China, 25-27 Aug. 2008: 964 - 968.

[7] R. Misra, C. Mandal, "Minimum Connected Dominating Set Using a Collaborative Cover Heuristic for Ad Hoc Sensor Networks", *IEEE Transactions on Parallel and Distributed Systems*, Accepted for future publication, 2010.

[8] D. Kim, Y. Wu, Y. Li, F. Zou, D. Du, "Constructing Minimum Connected Dominating Sets with Bounded Diameters in Wireless Networks", *IEEE Transactions on Parallel and Distributed Systems*, Feb. 2009, 20(2): 147-157.

[9] N. Jain, R. Biswas, N. Nandiraju, D.P. Agrawal, "Energy aware routing for spatio-temporal queries in sensor networks", *In IEEE Wireless Communications and Networking Conference*, New Orleans, LA, USA, Mar. 2005, 3: 1860–1866.

[10] A. Coman, M. A. Nascimento , J. Sander, "A framework for spatio-temporal query processing over wireless sensor networks", In *Proceeedings of the 1st International Workshop on Data Management for Sensor Networks: in conjunction with VLDB 2004*, Toronto, Canada, 2004: 104−110.

[11] A. Coman, J. Sander, M. A. Nascimento, "Adaptive Processing of Historical Spatial Range Queries in Peer-to-Peer Sensor Networks", *Distributed and Parallel Databases*, Dec. 2007, 22: 133-163.