International Review on Computers and Software (I.RE.CO.S.), Vol. 4, N. 2
March 2009

A New Middleware for Multipurpose Applications in Wireless Sensor Networks

S. M. Mazinani[1,2], M. H. Yaghmaee[3], M. T. Honary[1,2], F. Tashtarian[4], J. Chitizadeh[2]

Abstract – *Dramatic development of the wireless sensor networks (WSNs) in several applications is seeking for multi-application implementation of such networks. In this way, the middlewares contain a runtime environment that supports and coordinates multiple applications. In this paper, a middleware is proposed to support multi-application network that simultaneously satisfies the quality of service (QoS) requirements of users, that is the response time and data accuracy, and that of network that is optimizing the network resources (i.e. energy) utilization. Organizing nodes into clusters and preparing them to efficiently report data of any detected event, the middleware features an inheritance pattern among nodes that trains them to support multiple events. More over, employing the Data Integration among nodes not only reduces the energy consumptions for data reporting but also increases the data accuracy of every detected event.*

***Keywords**: Middleware, Inheritance, Wireless Sensor Network, Clustering, Quality of Service, Data Accuracy*

Nomenclature

CH	Cluster head
CM	Cluster member
NoDE	Number of detected events
E	Energy
DA	Data accuracy
DF	Data freshness
EiCH	ith event cluster head
BS	Base station

I. Introduction

The development of the wireless sensor networks (WSNs) in several industrial, home and military applications is dramatically increasing and multi-application implementation of such networks seems to be necessary. In this way, the middlewares are the solutions that contain a runtime environment that supports and coordinates multiple applications, and standardized system services such as data aggregation, control and node management policies adapting to target application, and mechanism to prolong the WSN's lifetime [1]. The QoS issue has been investigated by several researchers. Different applications require different quality of services (QoS) requirements and these requirements are satisfied in the different protocol layers. The author in [2] has outlined WSNs QoS requirements for several layers. In this paper we refer to a model proposed in [3] that is a simplified QoS model that includes two QoS perspectives: Application/User and Networks (see Fig. 1).

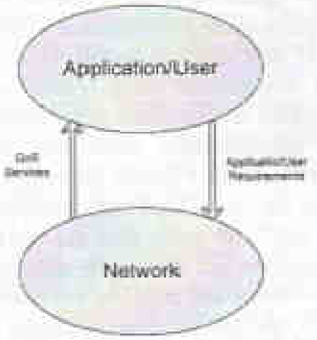


Fig. 1. A simple QoS model

Similar to [3], we assume that applications and users are in the same group because of their common way they perceive quality. The applications/users expect a specific QoS in terms of response time and data accuracy. These QoS requirements are passed to the network implicitly or explicitly and the underlying networks are responsible in part for meeting these requirements. Users are not concerned about how the network manages its resources or what mechanisms are involved in QoS provision. However, the users are concerned about the services that networks provide which directly impact the perceived quality of the application.

From the network perspective, the networks' goal is to provide the QoS services that adequately meet the users' needs while optimizing the network resources (i.e., energy) utilization. To achieve this goal, the networks analyze the application requirements, manage the network

Manuscript received and revised February 2009, accepted March 2009

Obviously, unless there is no Data Integration, the complex query requesting all the three events E_1, E_2 and E_3 will be missed. Even, the queries that request both E_2 and E_3 will be missed either. Whereas, the complex query is going to be responded by a Grandparent if there is Data Integration. A comparison is made between the number of missed complex queries in two different cases, i.e. with and without Data Integration and the results are given in Table II.

TABLE II
TOTAL NUMBER OF QUERY RESPONSES AND MISSED COMPLEX QUERIES

With Data Integration	336	322	14
Without Data Integration		298	38

The evaluation of energy

The energy of the network is drained gradually during its operation. In Fig. 15 the simulated network is run for two scenarios: Inheritable CHs and non-Inheritable CHs. Besides, Data Integration feature is once considered in the former scenario. The sharpest slope belongs to the Non-inheritable scenario in which all CHs detected the specific event respond to the requesting queries. Thus, network total energy dissipates so fast. However, in two other cases with the inheritance feature the network total energy is decreasing more slowly since only one CH is supposed to send.

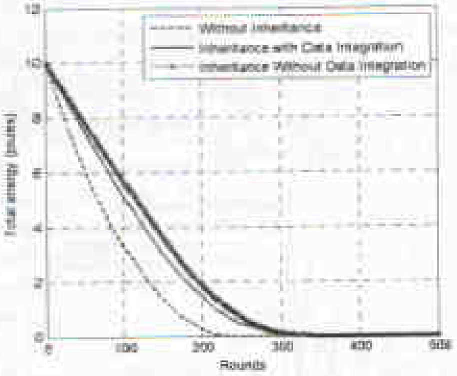


Fig. 15. Total energy of the network per round

In spite of that, the Data Integration feature requires in-network data processing that results in more (but not much) energy consumption in the network compared with inheritance without Data Integration feature. Importantly, a little bit more energy consumption of having Data Integration feature is worth to provide data that are more accurate for the user.

Number of alive nodes

The number of alive nodes are measured for the above three cases for a simulated network operating in equal situations for all. According to Fig. 16, in the non-Inheritance scenario the nodes' death happens much sooner than in the Inheritance scenario. That is due to the large number of data transmissions to the BS by the CHs. On the other hand, nodes live longer in the inheritance scenario since only one CH is supposed to send each round. The figure shows quite low difference between the nodes lifetime in the network both with Integration and without Integration in an inheritance-featured network. Going back to Fig. 15, that is due to that the energy consumption of the network that performs Data Integration is more than a network not doing so.

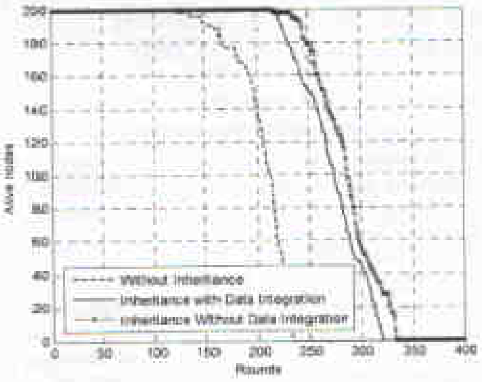


Fig. 16. Number of alive nodes per round

VI. Conclusion

In this paper, a middleware is proposed that uses a simplified QoS model that includes two QoS perspectives: Application/User and Network. The application/user expects a specific QoS in terms of response time and data accuracy. Whereas, from the network perspective, the networks' goal is to provide the QoSs that adequately meet the users' needs while optimizing the network resources (i.e., energy) utilization.

Being a multi-purpose middleware, it features in supporting various types of events (applications) by reconfiguring the network with respect to any new occurring event. To do this, the nodes are organized into clusters according to the type of any specific event and the nodes of each cluster become responsible to gather the data of the event they are specified to.

The inheritance feature is introduced in this middleware in which the new cluster head inherits the attributes of old cluster heads in supporting previous types of events besides its own specific one. The concepts of "Parent" and "Child" are introduced where the Child cluster head is the one that inherits the previous events from its Parent and becomes responsible of gathering and reporting them from now on. Through this way, the domain of detection of parental event is increased and more nodes in the Child cluster detect and report it. The inheritance feature reduces the probability of having missed events. Moreover, the Data Integration between cluster heads that are from an inheritance set (inheritance domain) gives the ability of providing a more accurate data for user. One cluster head -that we call it the

V. Evaluation and Simulation Results

The proposed middleware is simulated via MATLAB twice considering two different scenarios: Inheritable and Non-Inheritable clusters. The comparison of two scenarios clarifies the effectiveness of the innovative features of the proposed middleware such as inheritance and Data Integration. The measurements are done in number of missed events, data accuracy, number of missed complex queries, total energy of the network and total alive nodes in the whole network lifetime. The simulation parameters are listed in Table I.

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Network size	10000 m^2
BS location	(50,150) m
Number of Sensors	200
Initial Energy	0.05 J
E_{elec}	50 nJ/bit
ε_{fs}	10 pJ/bit/m^2
ε_{mp}	0.0013 pJ/bit/m^4
E_{DA}	5 nJ/bit/singnal
Data Packet Size	4000 bits
Control Packet Size	32 bits
d_c	87 m
$w_1, w_2, \ldots, w_i$	equal

Three types of events are supposed to occur regularly in random places and times. Events are considered as:

- E_1: Temperature more than 70°c.
- E_2: Humidity less than 20%.
- E_3: Smoke more than 10%

Scenario 1: Evaluation of network with Inheritable CHs

Events occur randomly in the network's region. The CHs specified to an event are responsible to report that queried event to the observer.

In other words, they are not going to gather the data of other types of events unless they are specified to them.

The inheritance feature brings the ability to the CHs to gather the data of other events too.

Without this feature, other types of events that have occurred in the respective cluster radius will be missed since the CH is not specified to them.For simulation, a simple query is broadcasted to the network requesting the event E_1.

Fig. 13 depicts the number of missed events in the whole network lifetime.

It compares the number of missed events considering the Inheritable and non-Inheritable CHs.

As we expected previously, the number of missed events in the network with Inheritable CHs are much lower than that in the non-Inheritable case.

A gradual rise is seen in the figure that is due to the death of the nodes and their failure to detect the events in the network.

Two figures almost meet each other at round 330 when all nodes in the network are dead and the events are totally missed.

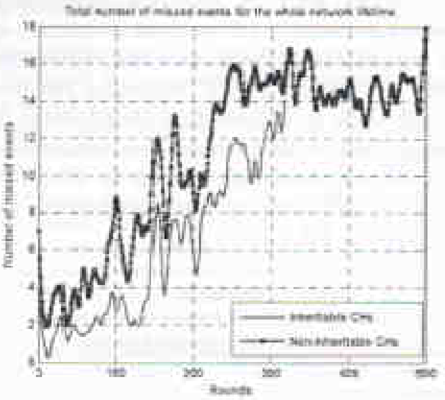


Fig. 13. Number of missed events per round in both scenarios

Scenario 2: Evaluation of network with Data Integration

The Data Integration is applicable in inheritance-featured networks in which the CHs in an inheritance domain integrate their gathered data. In our opinion, this feature results in more data accuracy and less missed complex queries.

Data accuracy measurement

During the Data Integration phase, the CHs in an inheritance domain send their measured data accuracy besides the real data to their Grandparent. The Grandparent has options to choose among the received data with more accuracy and freshness. Obviously, Data Integration results in more accurate data that is finally received at the BS. Fig. 14 illustrates the comparison of data accuracies with or without considering the data integration. Note that the inheritance feature is considered in both cases above.

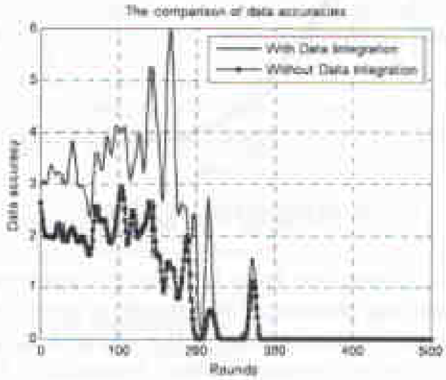


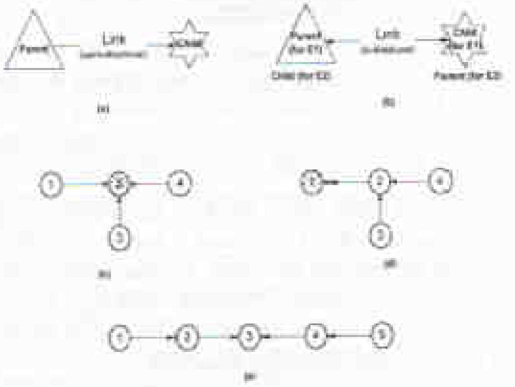
Fig. 14. Data accuracy of query responses per round in both scenarios

Number of missed complex queries

As mentioned previously, complex queries are answered by the inherited CHs. For example, a CH specified to E_1 and E_2 is able to respond to the complex queries requesting E_1 and E_2. Now, consider another CH that can respond to queries requesting E_2 and E_3.

required for all nodes to send their data to the maximum-degree node which can be more energy saving for them. In Fig. 11(c), the CH_2 can be reached by all other nodes for only one hop. Therefore, totally three number of transmissions to CH_2. However, as shown in Fig. 11(d), if CH_1 is selected, there would be totally five number of transmissions. The Grandparent node is likely to be the one with the maximum degree in an inheritance domain.

In Fig. 11(e), all three cluster heads, CH_2, CH_3 and CH_4 have in the same degree because of having equal numbers of I/O links. In this case, it is better to select Grandparent among those with more residual energy. The reason is that the Grandparent handles much more loads of functionalities such as internal node data processing and final data transmission to BS.



Figs. 11. The Links among CHs in an inheritance domain

Query response providing

The Grandparent is most qualified CH that is responsible to gather the data of other CHs in the inheritance domain. It alerts its qualification to others using the timing pattern (see: Apx1). After the Data Integration phase, the Grandparents choose the most suitable received data for the query response in their respective inheritance region. There may be several Grandparents in a large scale network that every one of them has provided a query response with the best data accuracy and freshness in its respective domain. The responses may be different in the sense of data accuracy and freshness for each individual event. The final query response must be provided so that it satisfies the user's requirement that had been mentioned in the query.

In other words, if the user is interested in a specific event, he will query that event with a larger weight value. The weight values in a complex query show the user's interests in the queried events. Thus, the final query response will be the one that is more qualified by the following equation:

$$Qualification\ Degree = \sum_{i=1}^{l} w_i . X_i \qquad (11)$$

where w_i is the weight value brought in the query that shows the user's interest in the event E_i and X_i is the degree of that event and it has been calculated once in the node using (10).

Based on the qualification degree in (11), Grandparents become candidate to send the query response to the BS and only one would send the final response using the timing pattern.

Routing and Reporting Sub-layer

According to Fig. 5, the "*routing and reporting*" sub-layer is across other layers in the distributed part. The most important function of routing and reporting sub-layer is to find the most suitable routs to the final destination BS. Since CHs especially Grandparents are exhausted due to high functionality they may die after the distant transmission to BS. The final Grandparent CH that has provided the final query response tries to find another node within its cluster in order to send the final data to BS. Obviously, the most suitable node would be the one with the most residual energy and most proximity to BS. The final grandparent CH finds this node via negotiation and sends its data to the selected node and it forwards them to BS.

Recall that there is a deadline time for arrival of every query responses. The period of time from sending the query towards the network to time of finding final Grandparent is called T' in time units. Also, suppose that it takes a constant T time units for final Grandparent to find the most suitable node in the cluster for data forwarding.

According to Fig. 12, as long as the sum of these durations is less than deadline that means final Grandparent CH_1 has enough time to find the suitable node inside its cluster using negotiation.

However, this is not achievable for final Grandparent CH_2 since the total time $T'+T$ passes the deadline.

In this case, CH_2 is forced to send the final response itself to the BS.

It is even worse for CH_3 since the final Grandparent is selected after the deadline time. Thus, its data is no more useful for the user and it is just send to BS (by a member) to be saved in the *storage component* for future use in query responses.

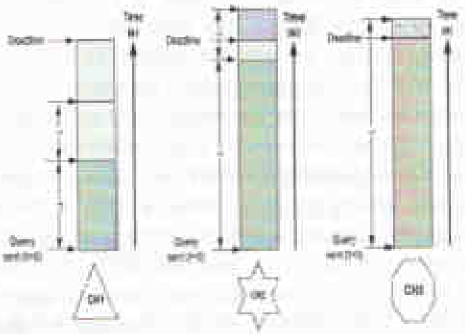


Fig. 12. Providing query response and final node selection based on query deadline

Suppose the winner candidate node, E_2CH, is among the nodes that are already located inside the cluster radius of E_1CH (See Fig. 7(b)). In that case, it is forced to gather the data of E_1 besides E_2. In other words, the E_2CH inherits E_1 from E_1CH and we call it "*in-cluster inheritance*". This is because it was potentially a member of E_1CH and used to report data of E_1 to E_1CH as well.

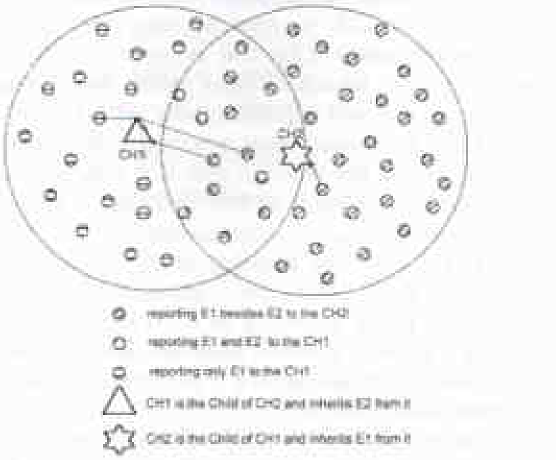


Fig. 7(b). In-cluster inheritance

Due to the probability of occurrences of two events E_1 and E_2 in the common area, the nodes in this area are responsible to report both of these events. Therefore, they detect both events E_1 and E_2 and send their information to their respective cluster head E_1CH. Thus, we say, "E_1CH must inherit E_2 from E_2CH so that it could gather this information".

The other ordinary nodes outside of the common area that have heard the advertisement of E_2CH and have not been members of any cluster must join the E_2CH and report the event E_2 to it. However, since E_2CH has inherited event E_1, the member nodes must also report the data of E_1. Thus, we say, "E_2CH must inherit E_1 from E_1CH so that it could gather this data". Other nodes outside of the common area that have been members of E_1CH will keep on reporting just data of E_1 to E_1CH. They do not report event E_2 since they have not heard the advertisement of E_2CH. There are two keywords in the concept of inheritance: "Parent" and "Child". The new CHs, known as "Childs", take over (or inherit) attributes and behavior of the pre-existing CHs, which are referred to as Parents (or ancestor CHs). According to these definitions, as illustrated in Fig. 7(a) for the event E_2, E_2CH would be the parent and E_1CH would be the child since it has inherited the E_2 from E_2CH. In Fig. 7(b), E_1CH is the parent and E_2CH is the child for the event E_1 and vise versa for the event E_2. The dynamic nature of the network makes it modify the configuration to support any newly occurred event namely E_3. Again, for the event E_3, some nodes become CHs and the inheritance is performed once more. Some cases of inheritance between CHs of E_1, E_2 and E_3 are depicted in Fig. 8.

Inheritance Domains

As mentioned above, any new event that occurs in the network results in forming some clusters with new CHs for supporting that event. These new clusters may be in commons with some other previously formed clusters. Thus, they inherit the events from each other based on the inheritance circumstances. "*An inheritance domain is the region that contains some inheritably related CHs that support different events*". These CHs have inherited events from each other. There may be several inheritance domains in different sizes in the network and there may be several different events being detected in each domain. A domain with two CHs has the least size. Some inheritance domains are also depicted in Fig. 8.

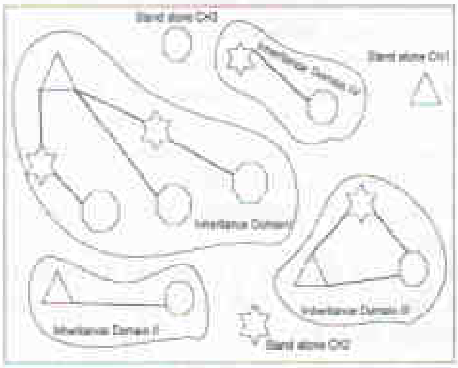


Fig. 8. The inheritance domains

The domains expand in size when any new event occurs. This is due to the fact that the new CHs enter the inheritance domains and that consequently the regions under coverage of parents enlarge in sizes.

For example, in Fig. 9, E_1CH is the parent and the coverage region of E_1 event (the parental event) is expanded after occurrence of the new event E_4 and formation of the cluster of E_4CH.

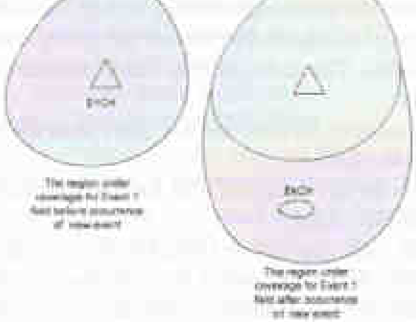


Fig. 9. The expansion of inheritance domain

Data Gathering/Event Detection sub-layer

This sub-layer is mainly responsible for gathering data based on received queries from users and provide appropriate response for them. The cluster members (CMs) take action to gather and report data to their own

Suppose the winner candidate node, E_2CH, is among the nodes that are already located inside the cluster radius of E_1CH (See Fig. 7(b)). In that case, it is forced to gather the data of E_1 besides E_2. In other words, the E_2CH inherits E_1 from E_1CH and we call it *"in-cluster inheritance"*. This is because it was potentially a member of E_1CH and used to report data of E_1 to E_1CH as well.

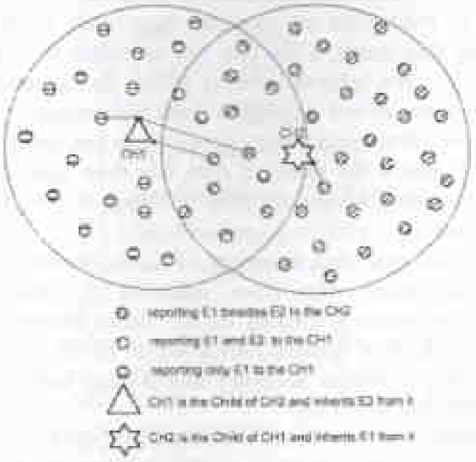


Fig. 7(b). In-cluster inheritance

Due to the probability of occurrences of two events E_1 and E_2 in the common area, the nodes in this area are responsible to report both of these events. Therefore, they detect both events E_1 and E_2 and send their information to their respective cluster head E_1CH. Thus, we say, "E_1CH must inherit E_2 from E_2CH so that it could gather this information".

The other ordinary nodes outside of the common area that have heard the advertisement of E_2CH and have not been members of any cluster must join the E_2CH and report the event E_2 to it. However, since E_2CH has inherited event E_1, the member nodes must also report the data of E_1. Thus, we say, "E_2CH must inherit E_1 from E_1CH so that it could gather this data". Other nodes outside of the common area that have been members of E_1CH will keep on reporting just data of E_1 to E_1CH. They do not report event E_2 since they have not heard the advertisement of E_2CH. There are two keywords in the concept of inheritance: "Parent" and "Child". The new CHs, known as "Childs", take over (or inherit) attributes and behavior of the pre-existing CHs, which are referred to as Parents (or ancestor CHs). According to these definitions, as illustrated in Fig. 7(a) for the event E_2, E_2CH would be the parent and E_1CH would be the child since it has inherited the E_2 from E_2CH. In Fig. 7(b), E_1CH is the parent and E_2CH is the child for the event E_1 and vise versa for the event E_2. The dynamic nature of the network makes it modify the configuration to support any newly occurred event namely E_3. Again, for the event E_3, some nodes become CHs and the inheritance is performed once more. Some cases of inheritance between CHs of E_1, E_2 and E_3 are depicted in Fig. 8.

Inheritance Domains

As mentioned above, any new event that occurs in the network results in forming some clusters with new CHs for supporting that event. These new clusters may be in commons with some other previously formed clusters. Thus, they inherit the events from each other based on the inheritance circumstances. *"An inheritance domain is the region that contains some inheritably related CHs that support different events"*. These CHs have inherited events from each other. There may be several inheritance domains in different sizes in the network and there may be several different events being detected in each domain. A domain with two CHs has the least size. Some inheritance domains are also depicted in Fig. 8.

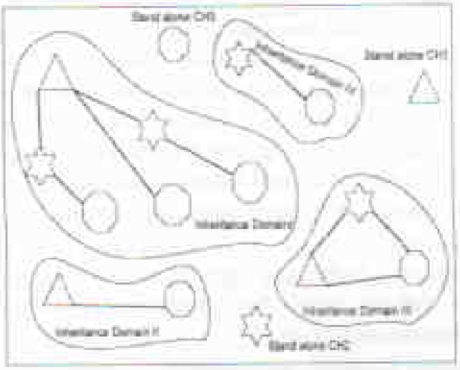


Fig. 8. The inheritance domains

The domains expand in size when any new event occurs. This is due to the fact that the new CHs enter the inheritance domains and that consequently the regions under coverage of parents enlarge in sizes.

For example, in Fig. 9, E_1CH is the parent and the coverage region of E_1 event (the parental event) is expanded after occurrence of the new event E_4 and formation of the cluster of E_4CH.

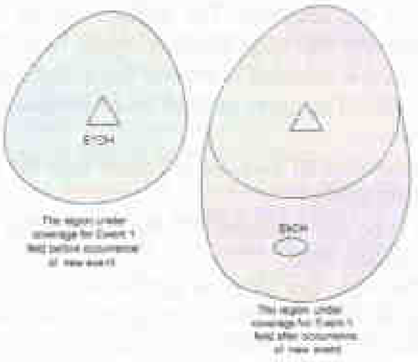


Fig. 9. The expansion of inheritance domain

Data Gathering/Event Detection sub-layer

This sub-layer is mainly responsible for gathering data based on received queries from users and provide appropriate response for them. The cluster members (CMs) take action to gather and report data to their own are broadcasted towards the network through this component.

Besides, in order to form the hierarchical structure in a wireless sensor networks, some commands need to be forwarded towards the network. These commands contain the characteristics of candidate nodes that are to be cluster heads later. Sending and receiving these commands are also performed by this component using broadcast transmission.

IV.2. Distributed Fart Inheritance Sub-Layer

This sub-layer includes two components that are "cluster formation" and "cluster inheritance".

Cluster formation component

The "cluster formation component" is responsible for organizing nodes into clusters in every round. In a specific region of the network, the clusters are formed according to the probability of occurrence of an event in that region. In other words, if the occurrence of a specific event is not probable in a region there would not be a cluster specified to that event.

A significant qualification of the proposed middleware is that it functions properly when apposing to the number of various events taking place and it can respond to the queries requesting data from these events as well. This is the due to the high scalability of this middleware in supporting of various events. Suppose the three events E_1, E_2 and E_3 occur in different times and places in the network. The dynamic nature of the network makes it to configure itself based on the frequency and location of the event's occurrence. Some important parameters are considered in the "cluster formation" sub-layer to select CHs and form clusters:

- Residual energy of nodes

The most energy draining activities in the cluster is subjected to CHs. The CHs are responsible for gathering, aggregation and reporting data. Performing such activities requires the CH node to be in an outstanding residual energy level.

- Data accuracy

As discussed previously, in the designing middleware one of the QoS requirements from the applications/users perspective is the "data accuracy". Several definitions have been offered in literature for this concept. For example, the authors in [16] believe that "data accuracy" of a node depends on its physical and topological features, given by its nominal precision, the environmental noise, its measurements and its proximity to the target area. In a network that contains sensor nodes with equal sensitivity deployed in a free space environment, the nodes closer to the event are likely to have a high relevance for sensing tasks.

- Data freshness

In some applications, the query sent to the network contains requests for the most recently events that have taken place. Obviously, the more qualified node is the one succeeded in detecting and recording the latest event in the duration of occurrence period. Data Freshness of a node is important in such applications. In order to gather the most recent data, the CH candidate nodes must be among those that have sensed the latest events.

- Number of detected events

In a period of time, an event may occur several times. In such a case, the node which has detected more events could be more qualified to be a candidate of being a CH.

According to the above mentioned characteristics of a qualified candidate node for being a CH, the qualification function, Q_i, is defined as below:

$$Q_i = \alpha_i \times E + \beta_i \times DA_i + \gamma_i \times DF_i + \lambda_i \times NoDE_i \qquad (5)$$
$$i = 1,2,\dots,I \qquad I : \textit{Total types of events}$$

where *E*, *DA*, *DF* and *NoDE* stand for "*Energy*", "*Data Accuracy*", "*Data Freshness*" and "*Number of Detected Events*" respectively and α, β, γ and λ are the "*QoS Coefficients*" and they are constants. Based on the type of application, the real values of these coefficients can be set and stored in the memory of each node. Consider that the first term of the above equation is to satisfy the QoS requirement from the network perspective that is "*energy*", while the other terms are considering the user QoS requirements.

The "time of detection" and the "Data Accuracy" of each event are stored in memory of each node shown in the Fig. 6. Data Freshness is earned by the detection time of each event.

Event Data Type	Detection Time	Data Accuracy
E1	DT1	DA1
E2	DT2	DA2
E3	DT3	DA3
E1	DTS-1	DAS-1
E2	DTS-2	DAS-2
E3	DTS-3	DAS-3

Fig. 6. A part of the node's storage

In order to elect candidate nodes to be CHs for a specific event, a command is broadcasted from BS toward the network in the following form:

$$Command_i = \{Q_i, MinQ_i\} \qquad (6)$$
$$i = 1,2,\dots,I \qquad I : \textit{Total types of events}$$

By receiving the command, all nodes calculate the qualification function (6). The result is compared with a threshold, *Min Q_i* that is the "minimum qualification requirement". Based on the network's circumstances and supply and a powerful radio so that is able to cover the whole network.

It organizes the sensors around itself into a layered infrastructure.

Other assumptions about the sensor nodes in the underlying network are as follows:

- Sensors nodes and the sink are all stationary after deployment. Nodes are uniformly distributed.
- All nodes are homogeneous and have equal nominal precisions.
- Each node is assigned a unique identifier (ID).
- Nodes are not location-aware, i.e. they are not equipped with GPS unit.
- All sensor nodes in the network are supposed to be synchronized via a synchronization beacon broadcasted from the BS at the beginning of every round.
- Nodes can use power control to vary the amount of transmission power that depends on the distance to the receiver.
- Nodes are able to sense every type of phenomenon and report any event in the network.

Every data transmission operation is performed in round. Each round is composed of three stages (Fig. 2).

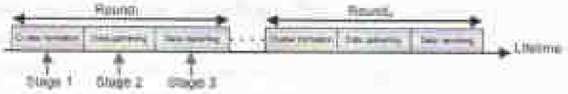


Fig. 2. The stages in every round

The first stage is cluster formation phase in which the clusters are formed and the cluster heads (CHs) are determined based on some metrics discussed later.

In the second stage, the members of each cluster send their gathered data to their respective CHs.

The last stage is specified to CHs to provide query responses that will be discussed later.

III.2. Energy Model

A simplified model shown in [4] for the radio hardware energy dissipation is used as follows.

To transmit an l-bit data to a distance d, the radio expends:

$$E_{Tx}(l,d) = E_{Tx-elec}(l) + E_{Tx-amp}(l,d) = \begin{cases} lE_{elec} + l\varepsilon_{fs}d^2 & d < d_0 \\ lE_{elec} + l\varepsilon_{mp}d^4 & d \geq d_0 \end{cases} \quad (1)$$

where first term presents the energy consumption of radio dissipation, while the second presents the energy consumption for amplifying radio. Depending on the transmission distance both the free space, ε_{fs}, and the multi-path fading, ε_{mp}, channel models are used.

It can also be written in a more general from as:

$$E_{Tx}(d) = p + qd^{\alpha} \quad (2)$$

In which p and q are constants related to node energy dissipation to run the radio electronics and power amplifier in transmitter and α is the path loss factor.

Each CH consumes energy for receiving data from adjacent nodes or cluster members and fuses it into a single packet.

This consumed energy for receiving and data fusion can be outlined by:

$$E_{rf}(m) = mE_{Rx} + E_{BF} \quad (3)$$

$$E_{Rx}(l) = lE_{elect} \quad (4)$$

where m is the number of members in the cluster that has detected the specific event and E_{BF} is the consumed energy for beam forming data aggregation [4].

IV. The Proposed Middleware

The proposed middleware considers the QoS from user and network's perspectives simultaneously in each application. Designing such a middleware requires consideration of a weighted combination of two different approaches: Bottom-up and Top-Down. The Bottom-Up designing approaches are typically focused more on constraints of sensor network hardware than on application requirements. These approaches have been dominant yet since the hardware constraints like the battery, memory and computational capabilities have been the most important parameters in the designing considerations [15]. While, in Top-down designing approaches a deep understanding of application requirements is needed for the design of middleware.

A weighted combination of these two approaches is used in the proposed middleware, in which both network and users' requirements are to be considered in designing such a middleware. The proposed middleware is separated into centralized and distributed parts.

The centralized part has got the most coupling with user interface and it forwards the user's requirements to the network. The distributed part mainly focuses to the network's requirements. The OSI model of the stack protocol is illustrated in Fig. 3. The Middleware layer has overlapping with the Network and Application layers.

Each part consists of some layers and sub-layers (Fig. 4) that are described in the proceeding. The "Application/User" and "base station" sub-layers are related to the centralized part and "Inheritance", "Data gathering/Event detection" and "Reporting and Routing" sub-layers are related to the distributed part. These sub-layers and their components are shown in Figs. 4 and 5.

Grandparent- among all cluster heads in an inheritance domain becomes volunteer of gathering the fused data of other cluster heads in that domain. The Grandparent selects the data having the best accuracy and sends it to the observer. In addition, the data integration feature results in more energy savings since only one node is responsible to send the final data. An evaluation was performed for the proposed middleware in a simulated network where some queries were broadcasted to the network requesting multiple data events. The Inheritance and Data Integration features were evaluated separately in two different scenarios in the sense of number of missed queries and accuracies of received data. The satisfactory results of simulation indicate that these features perform important roles in providing the QoS requirements of ser and network. There is an important limit of our proposed middleware. It is not hard real time because it works in query based manner.

Appendix: Timing pattern

Suppose that there is a set of nodes that are equipped with an internal timer. Every node in this set has the degree X_i such that:

$$X_{i\min} \le X_i \le X_{i\max} \qquad \text{(A1)}$$

Suppose that the node with the highest degree value is required. One way to select this node is that they negotiate with each other and select the node having maximum degree. Obviously, this is energy draining due to several message exchanges between nodes especially in dense regions. The timing pattern applies a timing competition among nodes. All nodes that are supposed to be initially synchronized set their internal timer to a back off time as below:

$$Backoff\ time = \frac{t_0}{X_i} \qquad \text{(A2)}$$

where t_0 is the maximum preset back off time. The node with the maximum degree X_{max} will time out sooner than others and it will send an advertisement packet around itself to announce its winning. Other nodes that hear the winning message will quit the competition by stopping their timer.

References

[1] S. Hadim, N. Mohamed, Middleware, Middleware Challenges and Approaches for Wireless Sensor Networks, *IEEE Distributed Systems Online*, Vol. 7, March 2006.

[2] Y. Wang, X. Liu, J. Yin, Requirements of quality of service in wireless sensor , *Proceedings of the International Conference on Systems*, IEEE 2006.

[3] A. Ganz, Z. Ganz and K. Wongthavarawat, *Multimedia Wireless Networks: Technologies, Standards and QoS*, (Prentice Hall, Upper Saddle River, NJ, 2004).

[4] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, An Application-Specific Protocol Architecture for Wireless Microsensor Networks, *IEEE Transaction on Wireless Communication*, Vol. 1, pp. 660-670, Oct. 2002.

[5] F. Tashtarian, M. Tolou Honary, M. Mazinani, A. Haghighat and J. Chitizadeh, A New Level Based Clustering Scheme for Wireless Sensor Networks, *Proc. of The sixth ACS/IEEE International Conference on Computer Systems and Applications (AICCSA-08)*, Doha, Qatar, March 31 - April 4, 2008.

[6] F. Tashtarian, M. Tolou Honary, M. Mazinani, A. Haghighat and J. Chitizadeh, An energy efficient data reporting scheme for wireless sensor networks, *Proc. of The sixth ACS/IEEE International Conference on Computer Systems and Applications (AICCSA-08)*, Doha, Qatar, March 31 - April 4, 2008.

[7] Bartolome Rubio, Manuel Diaz, Jose M. Troya. Programming approaches and challenges for wireless sensor networks. In *Proc. the 2nd International Conf. Systems and Networks Communications (ICSNC07)*, Cap Esterel, French Riviera, France, August 25-31, pp.36. 2007.

[8] Wassim Masri, Zoubir Mammeri. Middleware for wireless sensor networks: A comparative analysis. In *Proc. the 2007 IFIP International Conference on Network and Parallel Computing*, Dalian, China, Sept. 18|21, pp. 349-356, 2007.

[9] Nuno Costa, Antonio Pereira, Carlos Serodio, Virtual machines applied to WSN's: The state-of-the-art and classification. *In Proc. the 2nd International Conference on Systems and Networks Communications (ICSNC 07)*, Cap Esterel, French, Riviera, France, August 25|31, 2007.

[10] Wang MM, Cao JN, Li J et al. Middleware for wireless sensor networks: A survey. *Journal of computer science and technology. Vol. 23, no. 3*, pp. 305-326, May 2008.

[11] J. Gehrke, S. Madden, Query Processing in Sensor Networks, *IEEE Pervasive Computing, Vol. 3, no. 1*, pp. 46-55, January-March, 2004.

[12] S.R. Madden, M.J. Franklin, and J.M. Hellerstein, TinyDB: An Acquisitional Query Processing System for Sensor Networks, *ACM Trans. Database Systems, Vol. 30, no. 1*, pp. 122–173, 2005.

[13] S. Li, S. Son, and J. Stankovic, Event Detection Services Using Data Service Middleware in Distributed Sensor Networks, *Proc. 2nd Int'l Workshop Information Processing in Sensor Networks (IPSN 03)*, LNCS 2634, Springer, pp. 502–517, 2003.

[14] C. Srisathapornphat, C. Jaikaeo, and C. Shen, Sensor Information Networking Architecture, *Proc. Int'l Workshop Parallel Processing*, IEEE CS Press, pp. 23–30, 2000.

[15] K. Henricksen and R. Robinson, A survey of middleware for sensor networks: state-of-the-art and future directions, In *Proceedings of the international Workshop on Middleware For Sensor Networks*, pp. 60-65, Melbourne, Australia, November 28 - 28, 2006.

[16] F. Delicato, F. Protti, L. Pirmez and J.F. de Rezende, An efficient heuristic for selecting active nodes in wireless sensor networks, *Computer Networks. Vol. 50, Issue 18*, pp. 3701-3720, Dec. 2006.

Authors' information

[1]Computer and Communication Research Center,
Mashhad, Iran

[2]Dept. of Electrical Engineering,
Ferdowsi University of Mashhad, Iran

[3]Dept. of Computer Engineering,
Ferdowsi University of Mashhad, Iran

[4]Dept. of Computer and Information Technology,
Islamic Azad University of Qazvin, Iran

E-mails: smajidmazinani@gmail.com
m.tolou@ieee.org
h.yaghmaee@um.ac.ir
chitizad@um.ac.ir
tashtarian@yahoo.com