Congestion Avoidance and Energy Efficient Routing Protocol for Wireless Sensor Networks with a Mobile Sink

Majid I. Khan[1]
majid@ani.univie.ac.at

Wilfried N. Gansterer[1, 2], Günter Haring[1]
{wilfried.gansterer, guenter.haring}@univie.ac.at
[1] University of Vienna, Department of Distributed and Multimedia Systems, Austria
[2] University of Vienna, Research Lab Computational Technologies and Applications, Austria

***Abstract*- Congestion severely affects the performance of a wireless sensor network in two aspects: increased data loss and reduced lifetime. This paper addresses these problems by introducing a mobile sink based routing scheme for congestion avoidance and energy efficient routing in wireless sensor networks. The proposed scheme utilizes the sink mobility and an in-network storage model that is used to set up mini-sinks along the mobility trajectory of the sink. Mini-sinks are responsible for collecting data from the sensor nodes located in their vicinity, thus avoiding data flow to a single data collection point, e.g., a static sink that is the major cause of congestion, data loss and reduced lifetime of the sensor network. Also, in the proposed scheme data only has to travel a limited number of hops to reach the nearest mini-sink which helps to improve the energy consumption of the sensor nodes. Through simulation we show the effectiveness of the given routing scheme in terms of congestion avoidance and increased lifetime of the wireless sensor network.**

***Index Terms*- congestion avoidance, energy efficient routing, mobile sink, wireless sensor networks**

I. INTRODUCTION

The phenomenon of congestion can be observed in different types of wired and wireless networks even in the presence of robust routing algorithms. Congestion in wireless sensor networks (WSN) mainly occurs because of two reasons -- when multiple nodes want to transmit data through the same channel at a time or when the routing node fails to forward the received data to the next routing nodes because of the *out-of-sight problem*.

Applications of WSNs in the areas of environment and habitat monitoring require the sensor nodes to periodically collect and route data towards a sink. Also, it is known that each sensor node can only be equipped with a limited amount of storage, so if at any given routing node the data collection rate dominates the data forwarding rate congestion starts to build up at this node. Such type of congestion and data loss normally occurs at the nodes located in the vicinity of a static sink. Data loss at these nodes occurs due to the fact that at any given point of time a sink can only communicate with one or a limited number of sensor nodes. So the sink implements a round robin like algorithm to grant equal opportunity to all the sensor nodes located in the vicinity of the sink for transferring data to the sink. Thus, during any given time interval only a subset of the sink's neighboring nodes can transfer data to the sink, while the remaining nodes wait for their turn. In the meanwhile waiting nodes keep on receiving data from their neighboring nodes. As a result, after some time memory buffers at the waiting nodes fill up and further inflow of data leads to data loss. This type of congestion phenomenon that occurs because of many-to-one transmission is known as *funneling effect*.

The lifetime of the sensor network is another important aspect in environmental and habitat monitoring based applications of the WSN. Lifetime of a WSN can be defined as the time interval between the deployment of the sensor field and the time when the first sensor node fails due to complete energy dissipation.

In this paper we present an in-network storage model [1] based data routing protocol. The idea is to create a routing scheme for avoiding congestion, which is the major cause of data loss and increased energy consumption of the sensor field. We call the new protocol *congestion avoidance and energy efficient* (CAEE) routing protocol. In contrast to available congestion avoidance/control techniques [2, 3, 8, 10] the CAEE routing protocol is based on utilizing the sink mobility along a fixed trajectory in a WSN that leads to congestion avoidance and increased lifetime of the sensor network.

The rest of the paper is organized as follows: Section II summarizes related work, Section III states the problem and the presents the network model, Section IV discusses the in-network storage model, Section V discusses the CAEE routing protocol, Section VI summarizes the simulation based evaluation of the CAEE routing protocol, and Section VII concludes the paper.

II. RELATED WORK

This section summarizes the currently available techniques for congestion avoidance and removal in a sensor network. It also elaborates the recent work done on investigating the use of a mobile sink in WSNs.

A. *Congestion avoidance and control techniques*

Chen et al. [2] divided the techniques developed to address the problem of data congestion in WSN into two groups: congestion avoidance and congestion control. The former focuses on strategies to avoid congestion from happening and the latter works on removing congestion when it has occurred. From an implementation perspective these techniques can be categorized into three groups: data aggregation techniques, multi hop/path routing techniques, and flow control techniques.

Data aggregation techniques focus on utilizing spatial or temporal correlation between sensed data to reduce its quantity and hence prevent congestion [3]. These techniques are especially useful in environment or remote area monitoring applications where the consecutive data readings by the sensor nodes do not vary much over the time. Aggregation schemes that exploit such type of correlation amongst the sensed data are called temporal aggregation schemes [4]. Other types of aggregation schemes are known as spatial aggregation schemes. Routing nodes (responsible for forwarding the data towards the sink) implementing spatial aggregation schemes try to find correlation amongst the data received from different sensor nodes in an effort to reduce data size and hence avoid congestion. Galluccio et al. [5] showed that the use of spatial aggregation can be very helpful for avoiding congestion in the vicinity of the sink.

Barton et al. [6] tried to improve the data aggregation rate by applying a cooperative communication technique, where multiple nodes in a network cooperate to send data to the sink. It was observed that best results were obtained if the distance between the source and the sink is large. However, their scheme requires cross layer design for routing, scheduling and communication protocols to achieve preeminent results. Gao et al. [7] presented a tree based data aggregation scheme that utilizes a probe and recall protocol to connect the sensor nodes that are sparsely located at hot regions in the sensor field.

Multi hop/path routing techniques utilize the dense deployment of the sensor nodes to remove congestion from WSNs. These techniques enable the routing nodes to find alternate routing paths to reach the desired destination in case of congestion at a routing link. The idea is that when a routing node senses increased data traffic and packets start to drop, it requests the neighboring nodes to become part of the routing scheme, thus creating a multi path routing topology to share the data traffic and eliminate congestion from the network [8]. Zhu et al. [9] studied the tradeoff between the network lifetime and the required data rate from the sensor field to the sink. Although they do not address the issue of congestion directly, their work can be used to estimate the overhead caused by multi-path routing.

Flow control techniques try to control the amount of data that is flowing on the routing path to avoid congestion using various strategies. For example, Wan et al. [10] implemented a back pressure mechanism to restrict data flow, Akan et al. [11] allow the receiver node to regulate the outflow of data from the sender node. All the techniques belonging to this class restrict the flow of data which in general also results in data loss.

In addition to above mentioned schemes Wang et al. [12] proposed a node priority based congestion control scheme for wireless sensor networks. Their scheme is based on the assumption that the nodes located in a WSN have different bandwidth and wireless media control requirement for data transmission. Therefore, a node priority index can be generated on the basis of packet inter arrival time and service time at each node. With the help of this index, sensor nodes having heavy data traffic can be assigned more access to the transmission media than the nodes with less traffic. However, extra overhead is involved in this scheme for maintaining the priority index of the sensor nodes.

Chen et al. [2] presented a congestion avoidance scheme that is based on the idea that at any given point of time client nodes have complete information about the buffer status of their parent node. Therefore, in case of congestion the client node either reduces the data that it is forwarding to the parent node or switches to some other parent node.

Analysis of the given schemes has shown that although they avoid or remove congestion, but they fail to avoid data and energy loss during this process. For example, aggregation based techniques entirely depend on finding correlation between the sensed data. If the collected data is of diverse nature because of extraordinary activities of different types in the WSN, these techniques fail. Multi hop/path routing techniques tend to work well for controlling congestion and avoiding data losses, but they fail to avoid congestion in the vicinity of the sink. In case of flow control techniques data loss occurs when the back pressure reaches the leaf nodes. The leaf nodes become restricted and cannot forward the gathered data because of the back pressure mechanisms. On the other hand, sensor nodes require some space in the buffer to accommodate the newly collected data. Thus, in order to create room for storing the newly collected data, leaf nodes have to drop previously stored data which leads to data loss.

In summary, existing schemes tend to lead to data loss. In order to overcome this problem, an in-network storage model has been presented in [1]. Now will we discuss different types of sink mobility patterns and schemes for data routing to a mobile sink.

B. *Data routing towards a mobile sink*

Over the past few years the use of a mobile sink has increased in WSNs to achieve better performance, in particular for balanced utilization of the sensor field energy and to prolong the lifetime. The sink can follow three basic types of mobility patterns in a WSN: random mobility, predictable/fixed path mobility, and controlled mobility.

Random mobility: In this case the sink follows a random path in the sensor field and implements a pull strategy for data collection from the sensor nodes. Data can be requested from either one hop or *k* (where, $k>1$)

hop neighbors of the sink. Chatzigiannakis et al. [13] used random sink mobility in their schemes. Their results showed that if increased data latency is permissible then random sink mobility can be used for increasing the lifetime of the sensor network. Moreover, if the sink carries out data collection from k hop neighbors instead of one hop neighbors, data latency can be sufficiently reduced. However, with random sink mobility it is not possible to guarantee data collection from all the sensor nodes positioned in a WSN.

Predictable/fixed path mobility: Luo et al. [14] have tried to find a mobility strategy for the sink that can lead to the most energy efficient utilization of the sensor field. Their results show that the longest lifetime for the sensor network can only be achieved if the mobility trajectory of the sink is along the *periphery* of the sensor field. Increased data latency and packet loss are major problems that arise due to the sink mobility in wireless sensor networks. Luo et al. [15] have addressed these problems by presenting a routing protocol that not only balances the energy dissipation of the nodes but also tries to reduce data latency and loss. Their scheme is based on discrete mobility of the sink where the sink pause time is greater than its mobility time in the sensor field. One potential drawback of their scheme is that whenever the sink moves, routing paths need to be updated. Moreover, when the sink pauses at any point along the boundary, then the scenario becomes equivalent to that of a static sink case that leads to increased data loss in the vicinity of the sink.

Controlled mobility: Use of controlled sink mobility is also analyzed in WSNs for increasing the lifetime. Jayaraman et al. [16] outlined a framework which utilizes context aware mobile pervasive devices for data collection from the sensor field. These context aware devices are supposed to be intelligent enough to retrieve their possible future location on the basis of data generated by the field and direction of mobility. This information is utilized for planning the data collection process. Controlled sink mobility based schemes are a good option if reduced data latency is required, but they are less cost effective than random/fixed path mobility.

The discussion showed that if data latency is permissible, then the best routing strategy that incurs minimum data loss due to sink mobility and also provides maximum lifetime of the sensor network with minimum cost is obtained if the sink follows a discrete mobility pattern along the boundary of the sensor field [14, 15].

III. PROBLEM STATEMENT AND THE NETWORK MODEL

This section presents the problem addressed in this paper and also outlines the underlying network model.

A. Problem statement

As discussed in Section I, the major reason for congestion in a static sink based WSN is many-to-one transmission that is depicted in Figure 1. These routing patterns degrade the performance of a WSN in two dimensions: (*i*) increased data loss due to congestion in the vicinity of the sink, and (*ii*) reduced lifetime of the sensor network because the nodes located in the vicinity of the sink run out of their energy much quicker than the rest of the sensor field. So it can be inferred that if the single data collection point is replaced with multiple data collection points then significant reductions in the congestion and the data loss can be obtained. However any solution based on the use of multiple static sinks is not cost effective; moreover, nodes located in the vicinity of the static sinks also consume their energy much earlier than the rest of the sensor field. Therefore in this paper we address the above mentioned issues, by presenting a routing protocol that is based on an in-network storage model and a mobile sink.

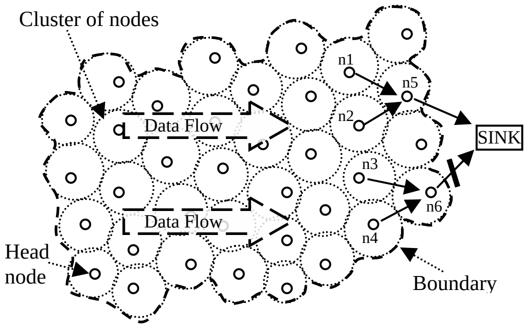


Figure 1. Sensor field with a static sink

B. Network model

We assume that the wireless sensor nodes are uniformly but randomly deployed to a remote sensor field in dense numbers. Nodes are responsible for sensing and reporting their readings in constant time intervals t to the sink. Moreover, we state the following assumptions about the sensor field shown in Figure 1;

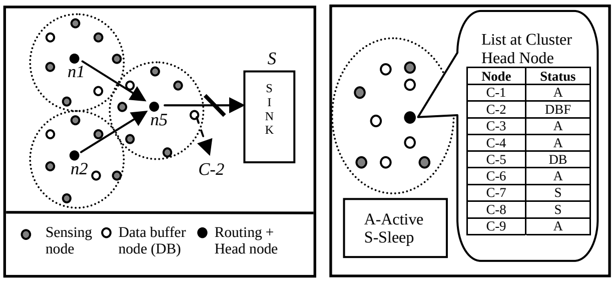
(i) Sensor nodes are grouped into clusters,
(ii) Each cluster has a head node that is responsible for forwarding the received data from neighboring head nodes and the client nodes towards the sink,
(iii) The head node also manages client nodes in the clusters by assigning them “awake” or “sleep” status depending on the node density and sensor field coverage requirements, and
(iv) Each node has a list of all its neighboring nodes (byproduct of many clustering algorithms [17]).

IV. IN-NETWORK STORAGE MODEL

This section reviews the in-network storage model that was developed to achieve data persistency in WSNs (for a complete discussion see [1]). The in-network storage model neither removes nor avoids congestion from happening, but it ensures data persistence under congestion and localizes its effects.

The in-network storage model is based on a clustered sensor field, where the cluster head node is responsible for energy efficient resource utilization and data routing towards the sink. It has been observed that under dense deployment of the sensor nodes only a subset of the nodes is needed to achieve complete coverage of the sensor field. Therefore, redundant sensor nodes are set to "sleep mode" by the cluster head nodes. This helps to increase the lifetime of the WSN and to avoid congestion by reducing the amount of data flowing along the routing paths towards the sink. The basic idea of the in-network storage is to utilize the redundant nodes (sleeping nodes) located in the vicinity of routing nodes (head node) as data buffers to avoid data loss from congestion in WSNs.

In order to better understand the idea, consider Figure 2(a) which shows a detailed view of a section of the sensor field shown in Figure 1, and Figure 2(b) which presents a detailed view of a cluster of nodes. It can be seen from Figure 2 that the nodes in a cluster are divided into two groups. One is the group of active sensor nodes which collect data from the field and the other is the group of sleeping sensor nodes. Since the head node is managing all the sensor nodes in a cluster, it maintains a list of all the cluster member nodes along with their current status as shown in Figure 2(b).



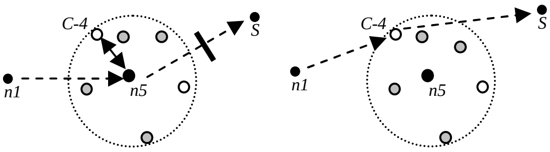
Node	Status
C-1	A
C-2	DBF
C-3	A
C-4	A
C-5	DB
C-6	A
C-7	S
C-8	S
C-9	A

(a) (b)

Figure 2. Clustered sensor field with data congestion at node N

Assume that the routing node *n5* in Figure 2(a) fails to forward the collected data to the sink because of a temporary out-of-sight problem or because the sink is busy collecting data from other neighboring nodes. As a result, congestion starts to build at node *n5*. In order to avoid data loss and to localize the effect of congestion in the vicinity of the node where congestion has occurred, node *n5* (which is also cluster head node) utilizes the in-network storage model and consults the list of client nodes shown in Figure 2(b). This list of neighbors provides information about the sensor nodes that can be used as data buffers. When the buffer at node *n5* reaches a certain threshold limit, then *n5* selects a buffer node (e.g., *C-5* shown Figure 2(b)) and starts to redirect the arriving data from *n1* and *n2* to the data buffer node. When a data buffer node becomes full to its capacity then it sends a BUFFER FULL message to the cluster head node. Then the head node marks this node as *DBF* and selects another sleeping node as data buffer. As a result, data loss can be avoided and the affect of congestion remains localized. Later on, when the forward link gets clear, the data can be retrieved from the buffer nodes and forwarded towards the sink by *n5*.

Two schemes can be used by the cluster head nodes (routing nodes) to redirect the arriving data to the buffer nodes. One is to always route the incoming data through the cluster head node towards the data buffer node. In this case no client node is allowed to communicate with any other node outside the cluster boundary. Therefore, all the incoming data is routed to appropriate buffer node via head node. Figure 3(a) shows one such scenario where the node *n1* is sending the data to node *n5*, which is then routed to buffer node *C-4* (by the node *n5*) to avoid data loss due to congestion at forward link *n5S*.



(a) (b)

Figure 3. Data storage and retrieval from the buffer node

The second option is to establish a direct connection between the data sender and the data buffer node, if they are within the communication ranges of each other. In this case, on detecting congestion the routing node *n5* first of all queries all the sleeping nodes about their list of neighbors and buffer capacities. If a buffer node (e.g., *C-4*) is located within the communication range of the data sender node *n1* then a direct link is established between *n1* and *C-4* by the routing node *n5* as shown in Figure 3(b). For this purpose the node *n5* transmits a message to *n1* informing about the new routing path *n1C-4* and the maximum amount of data that can be routed to the node *C-4* depending on *C-4's* buffer capacity.

The process of retrieving the stored data from the buffer nodes is similar to the process of storing the data. If the buffer node is located within the communication range of the next routing node *S*, then *C-4* is directed by the head node to send the stored data directly to *S*. Otherwise, the head node retrieves the data from *C-4* and forwards it to the node *S*, which potentially increases data latency and energy loss.

The next section shows how we utilized the in-network storage model to develop a congestion avoidance and energy efficient data routing scheme for WSN.

V. CAEE ROUTING PROTOCOL

To address the problems of congestion and energy efficiency in WSNs, we present the congestion avoidance energy efficient routing protocol (CAEE). CAEE is based on discrete sink mobility along a fixed trajectory in the WSN. Mini-sinks (*MS*) are created utilizing the in-network storage model along the mobility trajectory of the sink; each *MS* is managed by the data collector node (*Dc*). The term *mini-sink* in this paper represents a cluster of sensor nodes and a *data collector node* is the corresponding cluster's head node. The main responsibility of a data collector node is to receive and store the collected data from the sensor field to the mini-sink. The mobile sink periodically visits each mini-sink in the sensor field for data retrieval. Thus mini-sinks in our scheme act as temporary storage devices that are filled by the data from the sensor node and flushed by the sink.

Now we will answer the following questions: What should be the mobility trajectory of the sink? How are mini-sinks created? How can a routing node decide to which mini-sink it should forward the data?

The CAEE protocol does not impose any restriction on the shape of the mobility trajectory of the sink. However, it is known from [14, 15] that the most energy efficient routing is only possible if the mobility path of the sink is along the periphery of the sensor field. Therefore, without loss of generality, we select the periphery of the sensor field as the mobility trajectory of the sink.

How are mini-sinks created? During its first trip along the periphery of the sensor field the sink marks a subset of the nodes that it encounters as data collector nodes. These data collector nodes will perform two tasks: (*i*) set up a mini-sink utilizing in-network storage model, and (*ii*) inform the sensor nodes about the newly created mini-sink by broadcasting a message. The criterion for the selection of data collector nodes is based on distance measurement explained in the following,

The sink starts its mobility along the periphery of the sensor field from an arbitrary node located at the periphery of the sensor field called *start node*. If the start node is also a cluster head node then the sink assigns it the status of data collector node (*Dc-1)*. Otherwise, the sink queries the start node about its cluster head node. On retrieval of the required information, the sink assigns the status of data collector node (*Dc-1)* to the obtained cluster head node. Now the sink starts its mobility along the periphery of the WSN The sink selects the second data collector node *Dc-2* that is located at least *h* hops away from *Dc-1* and is also the cluster head node. Similarly, the third data collector node *Dc-3* is located at least *h* hops away from *Dc-2*, and so on. By the time the sink completes its first trip along the periphery of the sensor field a subset of the sensor nodes positioned along its mobility trajectory will have been converted into data collector nodes as shown in Figure 4.

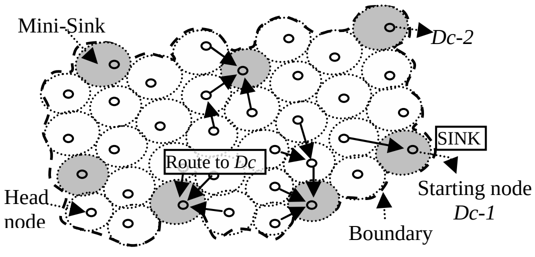


Figure 4. Sink following a fixed path around the sensor field

How are mini-sinks created? How can a routing node decide to which mini-sink it should forward the data? As mentioned before, when a node is assigned the status of data collector node by the sink then it broadcasts a message to inform the sensor nodes about the newly created mini-sink. The message contains two fields: the ID of the data collector node and the hop count that is initialized with 1. Each sensor node receiving this message performs the following check. If the available routing path to a data collector node is shorter than the newly reported route then discard the message; else update the previously stored route with the newly reported route. Increment the hop count by 1 in the received message and forward it. Thus, by the time the sink completes its first trip along the periphery of the WSN each node knows a shortest possible route to one of the data collector nodes as shown in Figure 4.

Each sensor node starts to broadcast the collected data to the nearest data collector node that receives and stores the data in one of the buffer nodes. The mobile sink stops at each mini-sink and requests data transfer from the data collector node. The data collector node first of all reports the total number of bytes that it wants to transfer to the sink and then starts the data transfer. The halting time of the mobile sink at a mini-sink is determined by the amount of data that is to be transferred from the mini-sink to the mobile sink. This saves us from keeping track of the position of the sink that is required in many existing mobile sink based routing protocols to avoid data loss.

Application of the CAEE protocol to a sensor field results in an increased lifetime of the WSN because data from each node has to travel only a minimal number of hops to reach the closest mini-sink from where it is collected by the mobile sink. Also, congestion and data losses can successfully be avoided because of the creation of multiple collection points instead of one static sink.

VI. SIMULATION BASED EVALUATION

This section presents a simulation based evaluation of the CAEE routing protocol. We used the OMNet++ simulation tool to analyze the performance of the CAEE protocol for congestion avoidance and energy efficient utilization of the sensor field. Since the CAEE routing protocol has introduced the idea of using a mobile sink and an in-network storage model based mini-sinks instead of a static sink for congestion avoidance and energy preservation in a WSN. Therefore in this section we analyze and compare the benefit of using the CAEE routing protocol over a static sink based scenario. However, to perform a fair comparison, also in the static sink based scenario routing nodes are equipped with the in-network storage model.

As mentioned before, sink mobility along the periphery of the WSN and a static sink positioned at the center of the sensor field leads to the most energy efficient data routing. Therefore, we used the same schemes for sink during our simulations. The simulation setup is comprised of a circular sensor field where the sensor nodes are deployed uniformly but randomly. Sensor nodes grouped themselves to form 200 clusters. Each cluster has a unique ID such that the cluster located at the center of the sensor field has the lowest ID (=1) and the clusters along the periphery have the highest ID (between 154 and 200). The radius of the sensor field is 300 meters and the communication range of each sensor node is set to 50 meters. All the sensor nodes are equipped with a limited battery of 0.5 joules. It is assumed that each message exchange (send/receive) costs a sensor node 50 nano joules of energy per bit [18].

By simulating the above mentioned setup we will answer the following questions: How many mini-sinks should be created in order to achieve congestion free and energy efficient routing in a WSN? How much benefit can be obtained with the implementation of the CAEE routing protocol instead of static sink based setup?

How many mini-sinks should be created in order to achieve congestion free and energy efficient routing in a WSN? We know from the definition of the CAEE routing protocol that each node has to route its data to the closest data collector node. Consequently, in order to reduce the energy dissipation of data routing nodes, a large number of data collector nodes should be selected uniformly along the mobility trajectory of the sink. This consideration leads to the following hypothesis.

Hypothesis: In order to achieve the best performance from the CAEE routing protocol in terms of energy efficiency and loss free routing *all* cluster head nodes located at the periphery should be assigned the status of data collector node.

To test this hypothesis we simulated the sensor field with different numbers of mini-sinks that are positioned equidistant from each other along the periphery of the sensor field. Each sensor node is equipped with an unlimited buffer capacity and is programmed to transmit a total of 100 data packet to its nearest data collector node. Figures 6 and 7 shows the results of our simulations in terms of the maximum number of data packets held by any routing node and the maximum energy consumption of the nodes in the WSN, respectively.

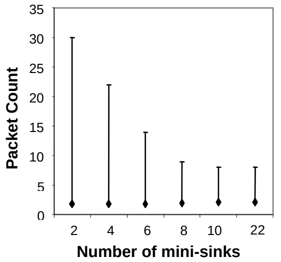


Figure 6. Avg./maximum no. of data packets held by routing nodes

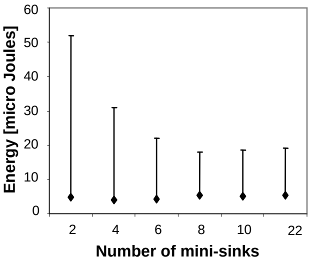


Figure 7. Avg./maximum energy consumption of routing nodes

When there were only two data collector nodes the maximum energy utilization and packet count at the routing nodes is very high compared to the average values over all the routing nodes. This indicates unfair resource utilization in the WSN leading to reduced lifetime and high data loss. In order to avoid this unfair resource utilization the difference between maximum and average values has to be reduced. Figures 6 and 7 illustrate that with an increase in the number of mini-sinks the difference between the maximum and average values of both the packet count and the energy consumption of the nodes starts decreasing and becomes almost constant when the data collector node count is 10.

These results support the previously formulated hypothesis to some extent, as increase in the number of mini-sinks helps to avoid congestion and balances out energy consumption in the WSN. But when the count of mini-sinks in the sensor field reaches a certain number (10 in our case) then addition of more mini-sinks has little or no effect on the sensor field in terms of reduced operation cost. So it can be inferred that in order to obtain best results from CAEE routing protocol the sink is required to set up only a limited number of mini-sinks which is substantially less than the maximum possible.

How much benefit can be obtained with the implementation of the CAEE routing protocol instead of a static sink based setup? The main purpose of developing the CAEE routing protocol was to eliminate congestion and prolonging the lifetime of the WSN by using a mobile sink and mini-sinks instead of a static sink. In order to evaluate the benefits in this regard we also carried out simulations to analyze the energy consumption and the state of the memory buffer at each routing node in the sensor network. For these simulations each node is programmed to transmit only one data packet. This restriction is applied for analyzing the data builds up at the mini-sinks, in order to determine the required speed of the mobile sink that can avoid data congestion at the mini-sinks. Also, as illustrated in Figures 6 and 7 the data packet count and energy consumption at the routing nodes become constant once the mini-sink count reaches to 10. Therefore we restrict the mini-sink count to 10 during our simulations. In the case of a static sink, it is placed at the center of the field.

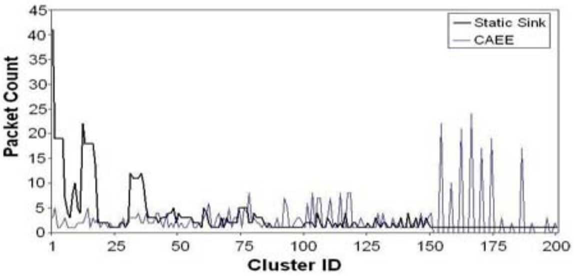


Figure 8. Maximum number of data packets held by each routing node

Figure 8 show the maximum number of data packets held by the routing nodes when routing is performed using the CAEE routing protocol and a static sink based routing scheme, respectively. It can be seen from Figure 8 that incase of CAEE routing protocol the maximum number of data packets held by any routing node except the periphery nodes (cluster IDs from 155 to 200) is 9. Also, the data routing load is almost evenly balanced over all the routing nodes. On the other hand, in the case of a static sink nodes located in the vicinity of the sink (Cluster IDs from 1 to 25) have the highest routing load and the nodes located at the periphery have constant minimal load (because they only have to forward their own data towards the sink). On the basis of these observations it can be concluded that in a static sink based scenario, the distance between the data routing node and the sink is inversely proportional to the routing load on the given node [14].

Figure 8 also shows the data packet count at the mini-sinks (cluster IDs 155-200) when one message per node has eventually reached to one of the mini-sinks. In the given scenario, if we impose a condition that each mini-sink can only store a maximum of 25 packets then the speed of the mobile sink should be such that it reaches all the mini-sinks before the arrival of new data. Analysis of the CAEE routing protocol shows that the speed of the sink should be directly proportional to the data generation rate of the sensor nodes and the number of mini-sinks in the sensor field, while the sensor node density should be inversely proportional to the speed of the sink. For example, if the data generation rate of the sensor nodes is high or if the number of mini-sinks in the WSN is low then the data buffers at the mini-sinks overflow more quickly because a large amount of data is being routed to each mini-sink. However, increase in the sensor node density does not affect the data generation rate of the sensor field in our setup because the cluster head nodes only activate those client nodes in a cluster that are required to achieve coverage of the sensor field. On the other hand, a higher number of redundant nodes increase the capacity of the mini-sink that helps to avoid data loss.

We repeated the above simulations with a condition that each routing node can only store a maximum of 10 data packets at any given time. Also, each sensor node was programmed to transmit a total of 100 data packets and it is supposed that the speed of the sink is fast enough to avoid data loss at the mini-sinks. Simulation results then indicated a packet loss of 32% in case of a static sink compared to 0% when the CAEE routing protocol was used with a mobile sink. The achieved gain is due the fact that the CAEE protocol removed the single data collection point which causes congestion and data loss in WSN.

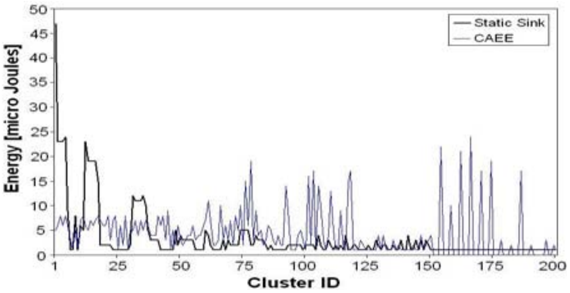


Figure 9. Energy consumption of routing nodes

Figure 9 present the energy utilization of each routing node when the CAEE routing protocol and a static sink based routing schemes were applied to the WSN, respectively. Analysis of Figure 9 shows that with the use of the CAEE routing protocol the lifetime of the WSN becomes almost twice as compared to that of static sink.

Another drawback of a static sink based routing scheme is that nodes located in the vicinity of the sink dissipate their energy much earlier than the rest of the sensor field. As a result, the link between the sink and the sensor field gets broken and the deployed setup becomes useless even if the majority of the sensor nodes are still active. While the CAEE routing protocol successfully avoids such scenarios by utilizing sink mobility.

Further improvement of the CAEE routing protocol

In order to further improve the performance of the CAEE routing protocol in terms of increased lifetime of the WSN the following strategy can be applied. Figures 8 and 9 show that only 10 out 45 potential data collector nodes (cluster head nodes located at the periphery of the WSN, cluster IDs 155 to 200) were heavily utilized. In order to utilize the remaining potential data collector nodes and to balance out the workload amongst them, the sink is programmed to revoke the status of a data collector node from a node *x* if the energy level of this node falls below a certain threshold. Then the sink searches for other potential data collector nodes in the vicinity of the node *x* with higher energy level and assigns the status of data collector node to the newly selected node.

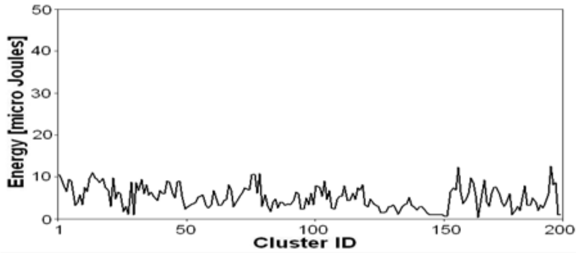


Figure 10. Balanced energy consumption of the routing nodes with improved scheme for CAEE routing protocol

Figure 10 shows that after applying the mentioned modifications energy consumption of the WSN become more balanced compared to Figure 9. This leads to nearly four times gain in the lifetime of the WSN compared to static sink based routing strategy. Moreover, since the new data collector nodes are selected in the vicinity of the old ones, only a slight change in the routing paths to the mini-sinks takes place that is within the few hops of the new and old data collector nodes. Thus the application of above mentioned strategy has negligible overhead compared to the huge gain in the lifetime of the WSN.

VII. Conclusion

This paper presented an in-network storage model based routing scheme that exploits dense sensor node deployment and the mobility of the sink in a WSN to set up congestion free and energy efficient routing paths, leading to increased lifetime of the WSN. Since in the given model the sink performs the data collection by moving from one mini-sink to another, the presented model is best suited for delay tolerant sensor networks.

References

[1] Majid I. Khan, Wilfried N. Gansterer, Guenter Haring, “In-network storage model for data persistence under congestion in wireless sensor network”. *In Proceedings of the First International Conference on Complex, Intelligent and Software Intensive Systems (CISIS'07)*, 2007, pp. 221-228.

[2] Shigang Chen, Na Yang, “Congestion avoidance based on lightweight buffer management in sensor networks,” *IEEE Transactions on Parallel and Distributed Systems*, Sept 2006. vol. 17, no. 9, pp. 934-946.

[3] Fan, K., Liu, S., and Sinha, P. “Scalable data aggregation for dynamic events in sensor networks”. *In Proceedings of the 4th international Conference on Embedded Networked Sensor Systems* (Boulder, Colorado, USA, October 31 -

November 03, 2006). SenSys '06. ACM Press, New York, NY, pp. 181-194.

[4] Sharaf, M. A., Beaver, J., Labrinidis, A., and Chrysanthis, P. K. “TiNA: a scheme for temporal coherency-aware in-network aggregation”. *In Proceedings of the 3rd ACM international Workshop on Data Engineering For Wireless and Mobile Access*. San Diego, CA, USA, September 19, 2003. MobiDe '03. ACM Press, New York, NY, pp. 69-76.

[5] Laura Galluccio, Andrew T. Campbell, Sergio Palazzo, “Concert: aggregation-based congestion control for sEnsoR networks”. *In Proceedings of the 3rd international conference on Embedded networked sensor systems 2005, Conference On Embedded Networked Sensor Systems*, San Diego, California, USA, pp. 274-275.

[6] Barton, R.J. Zheng, R. “Order-Optimal Data Aggregation in Wireless Sensor Networks Using Cooperative Time-Reversal Communication”. *40th Annual Conference on Information Sciences and Systems*, 22-24 March 2006, Princeton, NJ, pp. 1050-1055.

[7] Gao, J., Guibas, L., Milosavljevic, N., and Hershberger, J. “Sparse data aggregation in sensor networks”. *In Proceedings of the 6th international Conference on information Processing in Sensor Networks* (Cambridge, Massachusetts, USA, April 25 - 27, 2007). IPSN '07. ACM Press, New York, NY, pp. 430-439.

[8] Alberto Cerpa and Deborah Estrin. “ASCENT: Adaptive Self-Configuring sEnsor Networks Topologies”. *In the IEEE Transactions on Mobile Computing Special Issue on Mission-Oriented Sensor Networks*, July- September 2004, vol. 3, no. 3, pp. 272-285.

[9] Zhu, J., Hung, K., and Bensaou, B. “Tradeoff between network lifetime and fair rate allocation in wireless sensor networks with multi-path routing”. *In Proceedings of the 9th ACM international Symposium on Modeling Analysis and Simulation of Wireless and Mobile Systems* (Terromolinos, Spain, October 02 - 06, 2006). MSWiM '06. ACM Press, New York, NY, pp. 301-308.

[10] Wan, C., Eisenman, S. B., and Campbell, A. T. “CODA: congestion detection and avoidance in sensor networks”. *In Proceedings of the 1st international Conference on Embedded Networked Sensor Systems* (Los Angeles, California, USA, November 05 - 07, 2003). SenSys '03. ACM Press, New York, NY, pp. 266-279.

[11] Akan, Ö. B. and Akyildiz, I. F. “Event-to-sink reliable transport in wireless sensor networks.” *IEEE/ACM Trans. Network* 13, 5(Oct. 2005), pp. 1003-1016.

[12] Wang, C. Li, B. Sohraby, K. Daneshmand, M. Hu, Y., "Upstream congestion control in wireless sensor networks through cross-layer optimization," *IEEE journal on selected areas in communications*, vol. 25, no.4, 2007.

[13] Chatzigiannakis, I., Kinalis, A., and Nikoletseas, S. “Sink mobility protocols for data collection in wireless sensor networks.” *In Proceedings of the international Workshop on Mobility Management and Wireless Access* (Terromolinos, Spain, October 02 - 02, 2006). MobiWac '06. ACM Press, New York, NY, 52-59.

[14] J. Luo and J.-P. Hubaux. “Joint mobility and routing for lifetime elongation in wireless sensor networks.” *In Proceedings of the 24th Annual Conference of the IEEE Communications Societies (INFOCOM’05)*, FL, USA.

[15] J. Luo, J. Panchard, M. Piorkowski, M. Grossglauser, and J.-P. Hubaux. MobiRoute: Routing towards a Mobile Sink for Improving Lifetime in Sensor Networks. *In International Conference on Distributed Computing in Sensor Systems* (DCOSS’06), San Francisco, CA, USA.

[16] Prem Prakash Jayaraman, Arkady Zaslavsky, Jerker Delsing. “Sensor data collection using heterogeneous mobile devices.” *ICPS'07: IEEE International Conference on Pervasive Services* July 15-20, 2007, Istanbul, Turkey.

[17] Sun, K., Peng, P., Ning, P., and Wang, C. “Secure Distributed Cluster Formation in Wireless Sensor Networks”. *In Proceedings of the 22nd Annual Computer Security Applications Conference* (December 11-15, 2006). ACSAC. IEEE Comp Society, WashingtonDC,pp.131-140.

[18] He, Y., Zhang, Y., Ji, Y., and Shen, X. “A new energy efficient approach by separating data collection and data report in wireless sensor networks”. *In Proceeding of the 2006 international Conference on Communications and Mobile Computing* (Vancouver, British Columbia, Canada, July 03 - 06, 2006). IWCMC '06. ACM Press, New York, NY, 1165-117

Majid Iqbal Khan is currently a Ph.D. student at the Department of Distributed and Multimedia Systems at the University of Vienna, Austria. He received a masters degree in Software Engineering (2004) from COMSATS Institute of Information Technology, Islamabad Pakistan. His main research interests are in mobile wireless sensor networks.

Wilfried N. Gansterer is currently an associate professor of computer science at the University of Vienna, Austria, where he is head of the Research Lab for Computational Technologies and Applications. He received a PhD in computational mathematics from Vienna University of Technology, an MS in scientific computing and computational mathematics from Stanford University, and an MS in applied mathematics from Vienna University of Technology. His main research interests are in parallel and distributed computing as well as in scientific computing and computational science.

Günter Haring is currently a full professor in Applied Computer Science in the Department of Distributed and Multimedia Systems at the University of Vienna, Austria, since fall 1985. He received his doctor degree in Technical Physics from Graz University of Technology in 1970. His research expertise is in the areas of workload characterisation of computer and communication systems, QoS in mobile systems, and performance evaluation of parallel and distributed systems. From January 2000 until September 2004 he was dean of the School of Business Administration, Economics and Computer Science at the University of Vienna. Since October 2004 he is dean of the Faculty of Computer Science at the University of Vienna. From 1989 until 1993 he was in function as president of the Austrian Computer Society. He was one of the three founding members of the Austrian Center for Parallel Computation (ACPC). He was also founding member of the Computer-Measurement-Group-Central-Europe (CMG-CE). He is a member of the IEEE, IEEE Computer and Communication Societies, the Association for Computing Machinery (ACM), the Austrian Computer Society (OCG) and the German Gesellschaft für Informatik (GI). He has served as a chair/member of technical programm committees of many international conferences. He was and is partner in many european research projects. He has published over 150 scientific papers and is coeditor of four books.