Energy-Efficient Data Aggregation for Wireless Sensor Networks

Rabindra Bista and Jae-Woo Chang
Chonbuk National University
South Korea

1. Introduction

A Wireless Sensor network (WSN) (Heinzelman et al., 2000; Yick et al., 2008) consists of a large number of spatially distributed autonomous resource-constrained tiny sensor devices which are also known as sensor nodes (Horton et al., 2002). WSNs have some unique features, for instance, limited power, ability to withstand harsh environmental conditions, ability to cope with node failures, mobility of nodes, dynamic network topology, communication failures, heterogeneity of nodes, large scale of deployment and unattended operation. Although sensor nodes forming WSNs are resource-constrained, i.e., limited power supply, slow processor and less memory, they are widely used in many civilian application areas, including environment and habitat monitoring, healthcare applications, home automation, traffic control and in military applications such as battlefield surveillance (Pottie & Kaiser, 2000).

Because data from sensor nodes are correlated in terms of time and space, transmitting only the required and partially processed data is more meaningful than sending a large amount of raw data. In general, sending raw data wastes energy because duplicated messages are sent to the same node (implosion) and neighboring nodes receive duplicate messages if two nodes share the same observing region (overlapping). Thus, data aggregation, which combines data from multiple sensor nodes, has been actively researched in recent years. An extension of this approach is in-network aggregation (Considine et al., 2004; Madden et al., 2002; Bista et al., 2009) which aggregates data progressively as it is passed through a network. In-network data aggregation can reduce the data packet size, the number of data transmissions and the number of nodes involved in gathering data from a WSN.

The most dominating factor for consuming precious energy of WSNs is communication, i.e., transmitting and receiving messages. Therefore, reducing generation of unnecessary traffics in WSNs enhances their lifetime. In addition, involving as many sensor nodes as possible during data collections by the sink node can utilize maximum resources of every sensor node. As a result, an adverse scenario will not happen in a WSN in which the sensor nodes closer to the sink run out of energy sooner than other nodes and the network loses its service ability, regardless of a large amount of residual energy of the other sensor nodes.

Since communication is responsible for the bulk of the power consumption, many routing schemes in WSN are carefully designed to provide highly efficient communications among the sensor nodes (Heizelman et al., 1999). Among them, data-centric schemes are very popular where data transmissions are based on their knowledge about the neighboring nodes. Directed Diffusion (DD) (Intanagonwiwat et al., 2002a) and Hierarchical Data Aggregation (HDA) (Zhou et al., 2006) schemes are two representative data-centric schemes. A usual concept of conventional data gathering schemes is that they collect data by a sink node from sensor nodes and transfer data towards the sink node through multi-hop. However, it gives rise to two problems. The first one is the hotspot problem, in which the sensor nodes closer to the sink run out of energy sooner than other nodes. As a result, network loses its service ability regardless of a large amount of residual energy of the other nodes. The second one is that network generates unnecessary traffics during data transmission for choosing a proper path to send data.

Aggregated result of sensor data at the sink node is used for making important decisions. Because WSNs are not always reliable, it cannot be expected that all nodes reply to all request. Therefore, the final aggregated result must be properly derived. For this, the information of the sensor nodes (Node Identifications, IDs) contributing to the final aggregated result must be known by the sink node. And, the communication cost of transmitting IDs of all contributed sensor nodes along with the aggregated data must be minimized. Following are some promising reasons for transmitting IDs of sensor nodes along with their sensed data.

- To know the exact picture of sensors data by identifying which sensor nodes are sending their data for data aggregation.
- Data loss due to collision is inevitable in WSNs. Therefore, IDs of sensor nodes are needed to deal with data loss resiliency and accuracy of the final aggregated result of sensors data at the sink node.
- To know either a sensor node is providing service or not (survivability of a sensor node).
- In end-to-end encryption techniques such as (Girao et al., 2005; Castelluccia et al., 2005) sensor nodes share a common symmetric key with the sink node. Therefore, without knowing the sensor nodes that are contributing data in the aggregated result decryption of the encrypted aggregated result is impossible at the sink node.
- Many privacy preserving data aggregation techniques (Bista et al., 2010; He et al., 2007; Conti et al., 2009; Zhang et al., 2008) use seeds to hide sensor data. The sink node must know the IDs of sensor nodes that are contributing data to the aggregation result so that it can deduce the real aggregated result by subtracting seed values of the sensor nodes which were previously used for data hiding.
- In health care application, to support a common type of query like *"Select the sensor nodes which measure temperature > 98"* for knowing the patients with abnormal temperature.

Hence, a sink node must be aware of node IDs of those sensor nodes which contribute in aggregated value of sensors data in order to derive exact result of the collected data in WSNs. This is possible only when if there exists such a scheme which can transmit IDs of all the participating sensor nodes to the sink node. But, currently existing TinyOS (Hill et al., 2000) – an operating system running on the Berkeley motes (i.e., Mica Motes) (Horton et al., 2002) which has been envisioned as application development platform for WSNs– based privacy preserving data aggregation protocols for WSNs, like (Castelluccia et al., 2005), can not transmit the IDs of those all sensor nodes which contribute to the aggregated value of sensor data to the sink node due to following two reasons. The first is that TinyOS offers limited payload size of 29-byte. The second is that each sensor node ID is transmitted as a plaintext (2-byte) to the sink node. As a result, it restricts sending IDs of all contributed sensor nodes. Handling power is of utmost important. A small size packet is always preferable to WSNs because the communication of even a single bit consumes a significant amount of energy.

For Mica Motes, TinyOS predefined a packet of maximum 36 bytes size. As shown in Fig. 1, out of the 36-byte of the packet, 29-byte are allocated to sensor data (payload) and rest bytes to destination address, Active Message (AM) type, length, group and Cyclic Redundancy Check (CRC) to detect transmission errors. The payload may consist of sampled data, an encryption key/s for security reason and source ID. Since the size of the payload is limited to 29-byte there must be an optimal method in order to adjust IDs of a large number of sensor nodes in a single packet for huge WSNs.

Dest (2)	AM (1)	Len (1)	Grp (1)	Data (0 - 29)	CRC (2)

Fig. 1. TinyOS packet format for Mica Motes. The byte size of each field is indicated below the label. The shaded grey color is data field which can be encrypted.

For these reasons, we, in this chapter, propose a Designated Path (DP) scheme for energy-efficient data aggregation for WSNs. The propose scheme pre-determines a set of paths and runs them in round-robin fashion so that all sensor nodes can participate in the workload of gathering data from WSNs and transmitting the data to the sink node without generating unnecessary traffics during data transmissions. The main idea of our scheme is that each sensor node knows when the sensed/received data has to be sent through which one of its parent nodes for data aggregation before reaching to the sink node by avoiding the communication cost for knowing an appropriate parent node selection in order to aggregate data. In addition, we propose a novel mechanism in which a special set of real numbers are assigned as the IDs to sensor nodes so that a single bit is sufficient to hold an ID of a sensor node while transmitting aggregated data to the sink node. For this, we, first, generate signatures of fixed size for all IDs of respective sensor nodes and then superimpose the signatures of IDs of contributed sensor nodes during data aggregation phase. The analytical and simulation results show that our scheme is more efficient than existing methods in terms of energy dissipation while collecting data from WSNs.

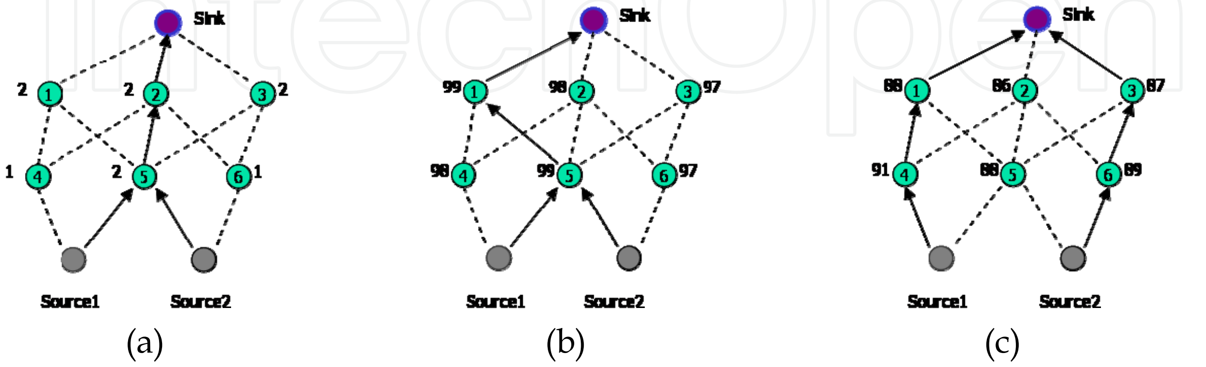
The rest of the chapter is organized as follows. In Section 2, we present related work. In Section 3, we describe how DP scheme works to aggregate data in WSNs and present our signature method to transmit IDs of many sensor nodes to the sink node. In Section 4, we show analytical models for our schemes and the existing schemes. Analytical performance evaluations are shown in Section 5. Section 6 presents simulation results. In Section 7, we conclude this chapter with some future directions.

2. Related Work

In this section, we, first, present a short review of the most related previous work on energy efficient data aggregation for WSNs and then briefly describe the work dealing with sending IDs of sensor nodes to the sink node.

Some researchers have explored in-network aggregation to achieve energy efficiency when propagating data from sensor nodes to the sink node (Madden et al., 2002; Madden et al., 2005; Intanagonwiwat et al., 2002b); Yao & Gehrke, 2003). In-network aggregation approaches are mainly differentiated by their network protocols for routing data. Among them, data-centric routing schemes are very popular where data transmissions are based on their knowledge about the neighboring nodes. Although there are many data-centric approaches (Akkaya & Younis, 2005), DD (Intanagonwiwat et al., 2002a) and HDA (Zhou et al., 2006) are two most related works to our research. In DD scheme, four phases are piggyback with four steps: *interest, exploratory data, reinforcement,* and *data*. A sink node broadcasts an interest describing the desired data to its neighbors. As interests are passed throughout the network, gradients are formed to indicate the direction in which the collected data will flow back. However, DD has two main problems to achieve an energy efficient data aggregation in WSNs. First, even though source nodes are near to the sink node, many other unnecessary nodes in the network are involved to propagate interests and setup gradients to the whole network. Due to this, DD generates unnecessary traffics during data transmissions. Second, DD achieves energy inefficient data aggregation because sources do not know where to forward data for aggregation. In DD, data are aggregated only by chance if the gradients are established as a common path for all sources nodes. As a result, many unnecessary nodes involved to gather data is energy inefficient. On the other hand, HDA overcomes the aforementioned two limitations of DD scheme. For this, HDA proposes a hierarchical structure to constrain exploratory data in a small scope between sink and source nodes. It also proposes parent-select aggregation principle to provide stronger aggregation capability than DD. However, the parent-select aggregation still suffers to achieve energy balanced data aggregation for WSNs. In HDA, there are two types of parent-select aggregation methods to perform data-level aggregation. In the first method, sources choose the parents which have the best attribute, in terms of number of child nodes, to save energy as shown in Fig. 2. Best attributes means the strongest data gathering capacity from as maximum number of sources as possible. This method suffers from hotspot problem and cannot balance energy for WSNs because some core nodes near to the sink, i.e., nodes 2 and 5 in the Fig. 2(a), are frequently used to gather data and run out of energy sooner than other nodes in the network. In the second method, sources choose the parents which have much energy than their siblings. It can balance energy for WSN but cannot guarantee data aggregation frequently as shown in Fig. 2(b & c). Due to this, the number of sensor nodes involved to gather data from the network increases leading to energy inefficiency. Moreover, in HDA, parent-select aggregation is achieved by periodically exchanging exploratory data and reinforcement between sources and the sink node. As a result, it generates unnecessary traffic during data transmissions. In addition, a common problem of both DD and HDA approaches is that they cannot be used for continuous data delivery for event-driven applications (Akyildiz et al., 2002).

On the other hand, CMT (Castelluccia et al., 2005) proposes additively homomorphic scheme to achieve secure data aggregation for WSNs. In the CMT scheme, each sensor node shares a key with the base station (BS) and uses the key to protect data privacy during their aggregation on the way to the BS. Therefore, the BS has to know which sensor has sent the data in order to decrypt the received aggregated data. This process requires transmission of all participated sensor nodes' IDs to the BS. For this, the CMT scheme first divides sensor nodes of a WSN into two groups (a group of data contributing sensor nodes and another group of data not contributing sensor nodes) and then sends IDs of sensor nodes from the group with lower number of sensor nodes as plaintexts (2 bytes of each ID) to the BS. Finally, the BS filters out real aggregated value from the collected data by subtracting proper key stream from the received encrypted aggregated data. However, considering TinyOS based Mica Motes for WSNs, the CMT scheme is not scalable because by using this scheme IDs of just twelve (12) sensor nodes are possible to send along with encrypted aggregated data. For larger size WSNs, it is impossible to decrypt the received data at the BS because of lack of knowledge of participated sensor nodes. In Reference (Zhang et al., 2008), each sensor node adds a seed to hide its data from other sensor nodes for achieving data privacy. Therefore, the knowledge of all source nodes is mandatory for the sink node to compute real aggregated value from the received aggregated data. For this, the work in (Zhang et al., 2008) transmits the IDs of data contributing sensor nodes as plaintexts to the sink node. A WSN is always prone to message-loss due to inevitable data collision property existed in wireless communications. Twin-key approach (Conti et al., 2009) deals with data-loss resiliency while achieving privacy preserving data aggregation by assuring a pair of common key alive for node to node communication. The IDs of those sensor nodes from which data is not getting are sent as plaintexts to the sink node. Like in the work (Castelluccia et al., 2005), both schemes (Conti et al., 2009; Zhang et al., 2008) are not scalable and they need much energy to transmit IDs of sensor nodes.



(a) (b) (c)

Fig. 2. Parent selection two data aggregation methods in HDA. Best attribute approach (a). Best energy approach with data aggregation (b). Best energy approach without data aggregation (c).

3. Propose Schemes

In this section, we first present our data aggregation scheme and then a scheme for transmitting IDs of a large number of sensor nodes to the sink node which we named signature scheme.

3.1 Our Data Aggregation Scheme

To overcome the shortcomings of DD and HDA schemes, we propose a new energy balanced and efficient approach for data aggregation in wireless sensor networks, called Designated Path (DP) scheme. In DP scheme, a set of paths is pre-determined and run them in round-robin fashion so that all the nodes can participate in the workload of gathering data form the network and transferring the data to the sink node. We use Semantic Routing Tree (SRT) (Madden et al., 2005) for disseminating any kind of aggregation query to get aggregated value such as *MIN, MAX, AVG, SUM* and *COUNT* (Madden et al., 2002).

3.1.1 Network Model

We assume a wireless sensor network model which is appropriate for data gathering applications such as target tracking. The network model has the following properties. First, a sink node without energy constraint is the root of the network topology and located on the top of it. Second, a large number of energy-constrained sensor nodes (e.g., MICA Motes) are deployed uniformly in the network area and they are equipped with power control capabilities to vary their output power. They are arranged in different levels based on the hop-count from the sink node. Third, each sensor node has the capabilities of sensing, aggregating and forwarding data and it can send fixed-length data packets to the sink node periodically. Finally, the sensor nodes can switch into sleep mode or a low power mode to preserve their energy when they do not need to receive or send data (Madden et al., 2005).

Our wireless sensor network model is similar to the structure of HDA scheme which is a multi-parent-multi-child hierarchical structure as shown in Fig. 3. In the multi-parent-multi-child tree structure, one sensor node can have many parent and child nodes and so the sensor node maintains them in two different lists, one for parent nodes and another for child nodes. But, packets are only transmitted between two nodes in neighboring levels. In this structure, all sensor nodes (M×N) are arranged in M levels starting from a sink node. The sink node is the root of the topology and is at level 0; nodes being one hop far from the sink are at level 1; nodes being two hops far from the sink are at level 2 and so on. As a result, lower the level a node is in, the nearer to the sink. Nodes at level i-1 are called 'parents' of nodes at level i, and nodes at level i+1 are called 'children' of nodes at the level i. To have a parent-child relationship between two sensor nodes, they must be within the communication range of each other.

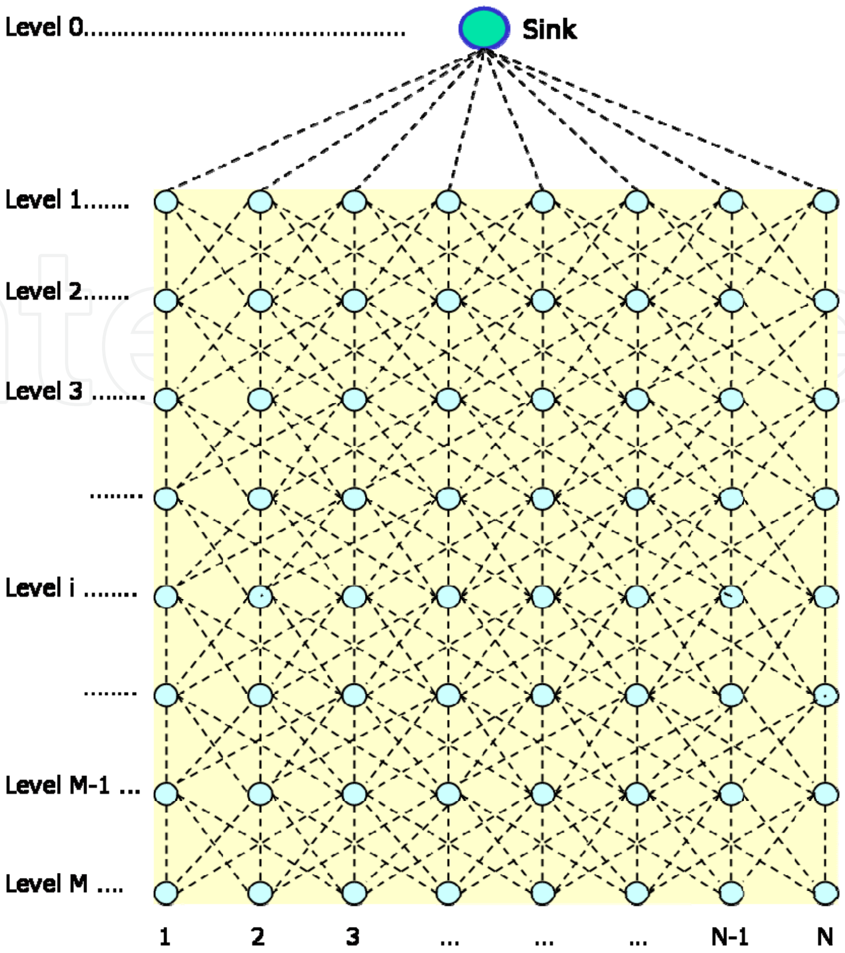


Fig. 3. A general view of network model for our data aggregation scheme.

3.1.2 Designated Path (DP) Scheme

Designated paths are a set of in-built paths, especially, designed for energy balance and efficient data aggregation for WSNs. In the DP scheme, a set of paths is pre-determined and run them in round-robin fashion so that all the nodes can participate in the workload of gathering data form the network and transferring the data to the sink node. In DP scheme, the forwarding behavior of all the nodes is scheduled to balance their burden of aggregation and transmitting network data. By using data aggregation knowledge, each sensor node knows when sensed or received or aggregated data has to send to which one of its parent nodes during data transmissions. In this way, unlike the existing schemes, DP does not generates unnecessary communication traffics to find an appropriate parent node and hence it works in energy efficient way. There are four main phases of DP scheme which are *path construction phase, best node selection phase, knowledge injection phase,* and *paths running phase*.

(a) Path construction phase: After deploying sensor nodes in a field, a multi-parent-multi-child hierarchical tree structure is constructed to provide communication paths for a WSN. In addition, N number of paths (for simplicity, N is equals to the number of columns of the WSN) are constructed for achieving energy-balanced data aggregation in the WSN. Each path is the shortest path from a sensor of level 1 to that of level M. So the first path, P1, consists of the sink and a sequence of the 1st sensor nodes of level 1 to level M, the second

path, P2, consists of the sink and a sequence of the 2nd sensor nodes of level 1 to level M and so on. In this way, we can create N paths for any M×N WSN and store them into a list of paths, PList. Because the paths of the PList will be allocated mainly for data aggregation in WSNs, we termed them as designated paths (DP).

(b) Best node selection phase: Based on the network connectivity, the best node from each path is determined for all of the sensor nodes of the WSN. A sensor node is said to be the best node among other sensor nodes of a path when the sensor node can be reached by any other sensor node of the network in the cost of minimum hop-count. By using Dijkstra's shortest path algorithm (Dijkstra, 1959), we can compute the best nodes for every sensor node of the network. If a sensor node can not reach to a path, then it inserts 'NULL' value and PathID of the path into its routing table. Otherwise, it inserts 'NodeID' of the best node and 'PathID' of the path. In this way, every node maintains the information of the best N nodes from the N number of designated paths, one node from each path in its routing table. The main goal of this phase is to create the routing table in order to use it as data aggregation knowledge for the WSN. Based on the routing table of the best nodes of a sensor node, the sensor node maps the best nodes to its parent sensor nodes so that it doesn't need to store a full path to reach the best node of any path.

(c) Knowledge injection phase: The application knowledge about designated paths and the best nodes is now loaded to each sensor node to achieve an efficient data aggregation in the WSN. By using this knowledge, in DP scheme, each sensor node of the WSN knows where to forward network data during their transmissions without generating unnecessary traffics. On the other hand, most of the existing routing protocols for sensor networks have to decide this task during data transmissions. For this, sensor nodes have to exchange unnecessary messages frequently among each others. It hurts a system in terms of energy efficiency because communication is the bulk of the power consumption and it decreases lifetime of a WSN. It also introduces a delay to the system.

(d) Paths running phase: The N paths from the PList are globally scheduled to all sensor nodes of the WSN so that the sensor nodes can run the paths in round-robin fashion. So, in one round, only one path, for instance P1, of the PList becomes active during data gathering and all the sensor nodes of the network are aware of P1 is active in this round. They send sensed/received/aggregated data to their best nodes from the path (P1) by using the data aggregation knowledge and data is automatically aggregated during their course to the sink node because all the sensor nodes use the same path which is active for the round. In the next round, the next path will be active, for example P2, and all of the sensor nodes send their data through P2 to the sink node. Data is aggregated progressively on their way to the sink node through P2. In the same way, the rests of the paths of PList are active one at a time to collect data from the WSN. The process is repeated after finishing one turn of all paths of the PList. Using designated paths in a round-robin mechanism provides an opportunity to all sensor nodes of the WSN to participate in the workload of gathering data from the network and transferring the data to the sink node. The forwarding behavior of all the nodes is scheduled to balance their burden of aggregating and transmitting the network data to the sink node. In this way, we overcome hotspot problem of the conventional approaches and believe that our DP scheme can achieve energy-efficient data aggregation in WSNs.

Furthermore, as DP scheme does not need to generate unnecessary traffics to select a path during data transmissions, it makes the networks energy efficient. In addition, our DP scheme can support continuous data delivery for event-driven applications.

3.1.3 Data Aggregation Algorithm

To avoid unnecessary communications overheads and achieve energy efficient data aggregation for WSNs, we present an algorithm for data aggregation in WSNs as given below in Fig. 4. The main goal of the propose algorithm is to generate data aggregation application knowledge for sensor nodes and they use it during data transmissions to the sink node.

For example, an 8×6 sensor nodes with a powerful sink are organized in a multi-parent-multi-child hierarchical structure, as shown in Fig. 5, where the total number of levels, M = 8, and the total number of columns, N = 6. In the first step, our algorithm creates six designated paths, P1, P2, P3, P4, P5 and P6 by selecting a sequence of appropriate sensor nodes for each path. The sequence of the nodes for P1, P2, P3, P4, P5 and P6 are < 1, 7, 13, 19, 25, 31, 37, 43 >, < 2, 8, 14, 20, 26, 32, 38, 44 >, < 3, 9, 15, 21, 27, 33, 39, 45 >, < 4, 10, 16, 22, 28, 34, 40, 46 >, < 5, 11, 17, 23, 29, 35, 41, 47 >, and < 6, 12, 18, 24, 30, 36, 42, 48 > respectively, starting from the sink node. All of the six paths are stored into a list of paths, PList. In the second step, the algorithm chooses the nearest nodes (in terms of minimum hop-count, MIN_hopc), called Best_nodes, one for each path for all of the sensor nodes of the network by using Dijkstra's shortest path algorithm (Dijkstra, 1959). If the algorithm can not find the best node from a path for any sensor node, it simply assigns value 'NULL' to the path. The meaning of 'NULL' is that when the path becomes active, the sensor node sends data through its default path (i.e., the path in which a node is situated in the network) because it is not located at the sub-tree of the path. This information is stored into the routing table (RTable) of the network. A sample of RTable to store the information of the best nodes is presented in Table 1. In this table, the first column represents the node identity of a sensor node for which we want to find the best nodes from the designated paths. The second column has entry type <Pi, Nj> where Nj represents the best node from path Pi to the sensor node of the first column. In the third step, the sink node uploads the routing table to all of the sensor nodes and each sensor node updates its original routing table which has already stored such information as a list of parent nodes, a list of child nodes, and its level in the network. The final step of this algorithm is to initialize the WSN. For this, the sink node either receives a SQL like aggregate query from a user or generates itself such type of query. Before propagating the query to the WSN, a query scheduler fetches the time duration of the query and assigns six time slots to the respective paths since the number of designated paths is 6 in this example. Then, it attaches the time schedule to the query and issues it to the WSN by instructing sensor nodes to run them in round-robin mechanism accordingly. When the sensor nodes receive the query, they send the data to the sink node according to the schedule. In this way, all the sensor nodes are synchronized to send the data through the particular active path and data are automatically aggregated during their course to the sink node through the active path. In the example, P3 is active at the moment, so all the source nodes, shown as dark nodes, send their data to their respective best nodes from P3 (for instance, node 15 is the best node for nodes 19 and 20) and data are aggregated before reaching to the sink node.

Input: Hierarchical (multi-parent-multi-child) M×N WSN, and
SQL type aggregation query
Output: Aggregated data from the network

Step1. Create a set of N number of designated paths through each column of the WSN

```
for sensor nodes  Nj =1 to N, Pj=1 to N; Nj++, Pj++;
   for level  Li =1 to M; Li++
      select LiNj
         insert into NList[ LiNj]    // list of nodes of a path
   Pj = NList
   insert into PList[Pj]
```

Step2. Select N number of best nodes, one from each path, for every sensor node

```
for sensor nodes LiNj =[1,1] to [M,N], Li++, Nj++;
  for Pj=1 to N, Pj++
    MIN_hopc = infinite value
    Best_node = NULL
    for Li =1 to M; Li++ make shortest hopc Array
 // using Dijkstra's algorithm, it finds hopc for LiNj and Pj
      Arry_hopc = DDistance(LiNj, Pj) ;
         if ( MIN_hopc > Array_hopc[Pj [Li]] )
           MIN_hopc = Array_hopc[Pj [Li]]
           Best_node = Li
  insert Pj and Best_node into RTable    // routing table
```

Step3. Load routing information to the sensor nodes

```
for sensor nodes LiNj =[1,1] to [M,N], Li++, Nj++;
   load (RTable);
```

Step4. Schedule and run the designated paths to collect data

```
Initialize ( );   // issuing an aggregation query
Time_to_run  =T    // life time of a query
Schedule( T);
   Pj = T/N  // Slotting T into N number of designated paths
for Pj =1 to N; Pj++
   Round_robin(PList [Pj] )  // running a path for a time slot
      Send_data(value)   //  sending  data through the path
      Aggregate(value);   /*data is aggregated during the
                                    course through the path*/
return value;
```

Fig. 4. Data aggregation algorithm for our DP scheme.

NodeID	Best Nodes For the Designed Paths
N1	{ <P1, NULL>, <P2, NULL>, <P3, NULL>, <P4, NULL>, <P5, NULL>, <P6, NULL> }
…	………………………………………………….
N8	{ <P1, N1>, <P2, N2>, <P3, N3>, <P4, N4>, <P5, NULL>, <P6, NULL> }
…	…………………………………………………..
N18	{ <P1, N1>, <P2, N2>, <P3, N3>, <P4, N4>, <P5, NULL>, <P6, NULL> }
…	………………………………………………….
N29	{ <P1, N13>, <P2, N14>, <P3, N21>, <P4, N22>, <P5, N23>, <P6, N24> }
…	………………………………………………….
N48	{ <P1, N25>, <P2, N32>, <P3, N33>, <P4, N34>, <P5, N41>, <P6, N42> }

Table 1. Routing information of sensor nodes.

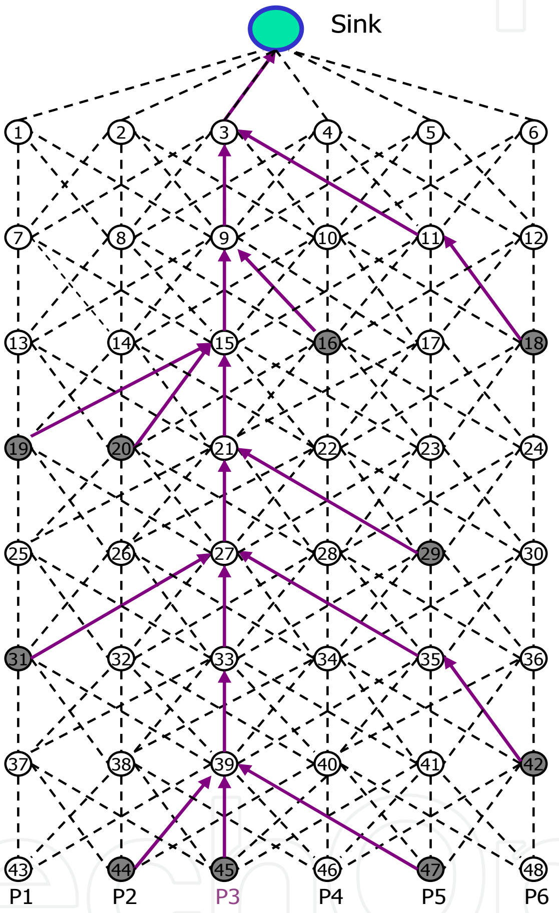


Fig. 5. Data aggregation in our DP scheme where path P3 is being active.

3.1.4 Scheduling

There are two levels of time scheduling in DP scheme. They are *path scheduling* and *communication scheduling*. For path scheduling, DP scheme applies a simple TDMA (Time Division Multiple Access) transmission scheduling mechanism which can be done either using the life time value of WSN or that of a user query (T), depending on the requirement of an application. Its basic idea is to subdivide T into as many number of fixed-length time intervals (slots) as the number of designated paths in a WSN. If the value of the T is very large, like in the case of continuous aggregate query, the path scheduler first divides T into M time slots and each time slot is further divided into the same number of slices as the number of designated paths, N. Fig. 6 shows the path scheduling for DP scheme. The designated paths are run in round-robin mechanism to collect data from the network. For each slice, only the scheduled path becomes active and path synchronization is maintained by all the sensor nodes of the WSN. The communication scheduling is related to how to synchronize the working behavior of all sensor nodes when the sink node collects data from the WSN. During processing of aggregation queries, it is required to coordinate the awaking times of children and parents in such a way that parent nodes can receive data from their child nodes before aggregating. To manage it, we adopt slotted approach (Madden et al., 2005) where the epoch is subdivided into a number of intervals, and assigned the intervals to the sensor nodes based on their position in the routing tree level of the hierarchical structure. It has been shown that the slotted approach can save a significant amount of energy in a hierarchical network structure.

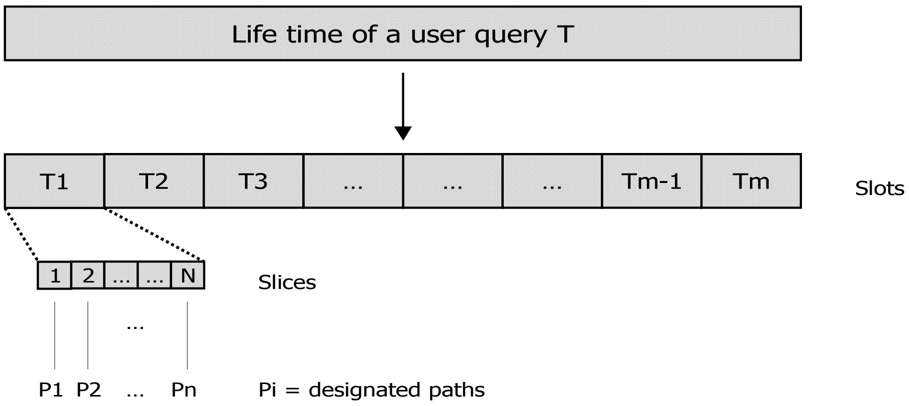


Fig. 6. Time division for designated paths in DP scheme.

3.2 Signature Scheme

To transmit IDs of a large number of sensor nodes in resource-constraint WSNs, we propose a novel approach based on signature of node ID so called signature scheme. There are five (5) steps in our signature scheme which we briefly describe each of them as follows.

(a) Assigning node ID to each sensor node: In this step, we assign a special type of positive integer 2^n (where, $n = 0$ to $Bn \times 8 - 1$, such that Bn is the number of free bytes available in the payload) to every sensor node as node ID. This is because the binary value of every integer of 2^n type has only one high bit (1). In addition, the position of the high bit for all integers of this type is unique. We termed this node ID as Real ID of a sensor node. The sink node knows a data contributing sensor node through its Real ID.

(b) Generating signatures of each sensor node ID: The Real ID of a sensor node assigned in the previous step is used to generate a signature of a fixed length. A signature is a fixed size bit stream of binary numbers for a given integer. Signature of a senor node ID can be generated by using the technique presented in the work (Zobel et al., 1998). We can determine the length of the signature based on the size of a given WSN. When the size of the WSN increases we can increase the length of the signature up to the *Bn* bytes. In other words, different size WSNs can have signatures of different lengths.

(c) Transmitting sensor data with signature of sensor ID: In this step, every source sensor node appends its signature as a sensor node ID rather than a plaintext used in the case of the

existing work. After including signature of its nodes ID in the payload, the sensor node forwards its packet to the upper layer sensor node. The sink node is the final destination of all sensor data where they ultimately aggregated.

(d) Data aggregation and superimposing signatures of IDs of sensor nodes: In this step, data aggregators collect data and signatures of the associated sensor nodes to perform following tasks. First of all, they aggregate received data according to the provided aggregation function such as *Average* of sensor data. Next, they superimpose signatures of the sensor nodes by performing bitwise *OR* operation on the bit streams of their Real IDs. Finally, the data aggregators rout aggregated result with the superimposed signatures of Real IDs of contributed sensor nodes to the sink node. Sine this approach needs just one bit to carry an ID of a sensor node it is 16 times scalable than the existing work where plaintexts (2-byte each) are used for carrying IDs of sensor nodes by simply concatenating them.

(e) Computing the final aggregated result and fetching IDs of contributed sensor nodes: When the sink node received partially aggregated data and the superimposed signatures from every sub-tree, it deduces the final aggregated result from the received aggregated data. Since the payload of the partially aggregated data contains signatures of IDs of sensor nodes the sink node can know all the contributed sensor nodes. To know the knowledge of contributed sensor nodes, the sink node separates the high bits (1s) of the superimposed signature of the each sub-tree by performing bitwise *AND* operation with the pre-stored signature files of Real IDs of sensor nodes.

SN ID	Real ID	2-byte Signature
1	$2^{0} = 1$	0000000000000001
2	$2^{1} = 2$	0000000000000010
3	$2^{2} = 4$	0000000000000100
4	$2^{3} = 8$	0000000000001000
5	$2^{4} = 16$	0000000000010000
6	$2^{5} = 32$	0000000000100000
7	$2^{6} = 64$	0000000001000000
8	$2^{7} = 128$	0000000010000000
9	$2^{8} = 256$	0000000100000000
10	$2^{9} = 512$	0000001000000000
11	$2^{10} = 1024$	0000010000000000
12	$2^{11} = 2048$	0000100000000000
13	$2^{12} = 4096$	0001000000000000
14	$2^{13} = 8192$	0010000000000000
15	$2^{14} = 16384$	0100000000000000
16	$2^{15} = 32768$	1000000000000000
Signature Superimposing by using bitwise OR operator (|)		1111111111111111
Example: The sink node fetches SN 8 using the signature of Real ID 128 and AND operator (&)		1111111111111111 & 0000000010000000
		= 0000000010000000

Table 2. Real ID of sensor nodes with signature.

Table 2 illustrates Real ID of 16 sensor nodes (SNs) with 2-byte size signature of each Real ID, signature superimposing process by using bitwise OR operator and an example of fetching a sensor node (SN 8) from the superimposed signature by using the Real ID 128 of SN 8 at the sink node.

3.2.1 Extension to Real ID Assignment and Signature Structure

In the previous section, we described about assigning Real ID to each sensor node using a set of positive integers of type 2^n. Now, we present variants of the integer type 2^n are also applicable to use as Read IDs for sensor nodes. For simple exposition of our idea, we consider three types of integer set: $2^n – 1$, 2^n and $2^n + 1$. For a Real ID of each set, we allocate memory of 2 bytes. Therefore, the total space required to include three Real IDs one for each integer set in the payload is 6 bytes. They can be organized in ascending order, i.e., first an ID of type $2^n – 1$, then ID of type 2^n and finally ID of type $2^n + 1$ occupying continuous 6 bytes space. Fig. 7 shows an algorithm for providing 6-byte signature containing all the three types of Real ID of sensor nodes. The main notion of this algorithm is to make use of the signatures of 2^n type Real IDs for both $2^n - 1$ and $2^n + 1$ types Real IDs and they are distinguished by allocating a particular slot to each type of Real IDs in the memory space of the payload. Every source node transmits its data along with 6-byte bit stream of its Real ID to the immediate parent node. The parent node aggregates sensor data of its child nodes, superimpose their 6-byte size signatures and forwards the packet towards the sink node. When the sink node receives a packet of aggregated data from each sub-tree it executes the algorithm shown in Fig. 8 to identify the contributed source nodes. The sink node first separates the superimposed 6-byte signature into three chunks each of continuous 2-byte size. Next, it generates a list of Real IDs from each chunk as shown in Table 2 and assembles them. By mapping Real IDs to SN IDs, the sink node finally knows all the contributed sensor nodes of the received aggregated data.

```
Input: Real IDs of sensor nodes
Output: Signatures of Real IDs
// Check the types of Real IDs
  if  Real ID type = $2^n$
    GenSig (Real ID);               // 2 bytes
     Padding zeros left and right;  // 2 bytes in each sides
  else if  Real ID type =$2^n – 1$
    GenSig(closest $2^n$);
     Padding zeros right;           // 4 bytes
  else                              // type = $2^n + 1$
    GenSig(closest $2^n$);
     Padding zeros left;            // 4 bytes
```

Fig. 7. An algorithm to fix spaces for the signatures of Real IDs of types 2^n -1, 2^n and $2^n + 1$ by padding zeros.

```
Input: Superimposed fixed size bit stream (6-bytes)
Output: List of contributed sensor nodes
// Separates the superimposed bit stream from the payload
  split(superimposed bit stream);
    A = 2-byte; B=2-byte; C=2-byte;
     select A;                          // the first 2 bytes
       { fetch_Real_IDs(A);             // as shown in Table 1
  for all Real IDs
    Real ID = Real ID – 1;         // $2^n$ – 1 type
   List1 = Real ID;}
   select B;                            // middle 2-byte
     { fetch_Real_IDs(B);
         for all Real IDs
                  List2= Real ID;}               // $2^n$ type
   select C;                            // the last 2-byte
    { fetch_Real_IDs(C);
               for all Real IDs
                Real ID = Real ID + 1;        // $2^n$ + 1 type
                  List3 = Real ID;}
   List =List1 + List2+ List3;  // list of all Real IDs
   List_SN_ID = List;               // using mapping file
    Retrieve List_ SN_ID;
```

Fig. 8. An algorithm to show the process of generating IDs of contributed sensor nodes from the superimposed bit stream of a packet by the sink node.

Table 3 illustrates ID of sensor nodes (SN ID), their respective Real ID with signatures of 6-byte for 32 sensor nodes. First, out of 32 sensor nodes, SNs <3, 6, 9, 12, 15, 18, 21, 24, 27, and 30> have Real IDs of type 2^n – 1 and they have signatures of the closest 2^n type integers. For instance, SN 6 has Real ID 7 and the Real ID 7 takes the signature of Real ID 8 because latter is the closest 2^n type integer to former. Since every 2^n – 1 type integer is smaller than respective 2^n type integer former occupies earlier position in the 6-byte space than latter. So, in the signature of every 2^n -1 integer a high bit (1) appears within the first 2-byte of the 6-byte signature and the remaining 4-byte space is padded with zeros. Next, SNs <1, 2, 4, 7, 10, 13, 16, 19, 22, 25, 28 and 31> have Real IDs of 2^n type integers. For instance SN 10 has Real ID 16, and the signature of this type takes the middle position of the 6-byte space having 2-byte zero padding in both left and right sides. Finally, the remaining SNs <5, 8, 11, 14, 17, 20, 23, 26, 29 and 32> have Real ID of type 2^n + 1 and they have signature of the closest 2^n type integers. For instance, SN 14 has Real ID 33 and it takes the signature of Real ID 32 which is the closest integer of type 2^n. Since every 2^n + 1 type integer is larger than respective 2^n type integer it occupies the last 2-byte of the 6-byte signature. For instance, SN 17 has Real ID 65 and the Real ID 65 takes the signature of Real ID 64 with 4-byte zero padding in the beginning.

SN ID	Real ID	2-byte Signature	6-byte Signature (Padding 4-byte Zeros)
1	$2^0 = 1$	0000000000000001	000000000000000000000000000000010000000000000000
2	$2^1 = 2$	0000000000000010	000000000000000000000000000000100000000000000000
3	$2^2 -1 = 3$	0000000000000100	000000000000010000000000000000000000000000000000
4	$2^2 = 4$	0000000000000100	000000000000000000000000000001000000000000000000
5	$2^2 +1 = 5$	0000000000000100	000000000000000000000000000000000000000000000100
6	$2^3 -1 = 7$	0000000000001000	000000000000100000000000000000000000000000000000
7	$2^3 = 8$	0000000000001000	000000000000000000000000000010000000000000000000
8	$2^3 +1 = 9$	0000000000001000	000000000000000000000000000000000000000000001000
9	$2^4 -1 = 15$	0000000000010000	000000000001000000000000000000000000000000000000
10	$2^4 = 16$	0000000000010000	000000000000000000000000000100000000000000000000
11	$2^4 +1 = 17$	0000000000010000	000000000000000000000000000000000000000000010000
12	$2^5 -1 = 31$	0000000000100000	000000000010000000000000000000000000000000000000
13	$2^5 = 32$	0000000000100000	000000000000000000000000001000000000000000000000
14	$2^5 +1 = 33$	0000000000100000	000000000000000000000000000000000000000000100000
15	$2^6 -1 = 63$	0000000001000000	000000000100000000000000000000000000000000000000
16	$2^6 = 64$	0000000001000000	000000000000000000000000010000000000000000000000
17	$2^6 +1 = 65$	0000000001000000	000000000000000000000000000000000000000001000000
18	$2^7 -1 = 127$	0000000010000000	000000001000000000000000000000000000000000000000
19	$2^7 = 128$	0000000010000000	000000000000000000000000100000000000000000000000
20	$2^7 +1 = 129$	0000000010000000	000000000000000000000000000000000000000010000000
21	$2^8 -1 = 255$	0000000100000000	000000010000000000000000000000000000000000000000
22	$2^8 = 256$	0000000100000000	000000000000000000000001000000000000000000000000
23	$2^8 +1 = 257$	0000000100000000	000000000000000000000000000000000000000100000000
24	$2^9 -1 = 511$	0000001000000000	000000100000000000000000000000000000000000000000
25	$2^9 = 512$	0000001000000000	000000000000000000000010000000000000000000000000
26	$2^9 +1 = 513$	0000001000000000	000000000000000000000000000000000000001000000000
27	$2^{10} -1 = 1023$	0000010000000000	000001000000000000000000000000000000000000000000
28	$2^{10} = 1024$	0000010000000000	000000000000000000000100000000000000000000000000
29	$2^{10} +1 = 1025$	0000010000000000	000000000000000000000000000000000000010000000000
30	$2^{11} -1 = 2047$	0000100000000000	000010000000000000000000000000000000000000000000
31	$2^{11} = 2048$	0000100000000000	000000000000000000001000000000000000000000000000
32	$2^{11} +1 = 2049$	0000100000000000	000000000000000000000000000000000000100000000000

Table 3. Real ID of thirty-two (32) sensor nodes with 6-byte signature.

In this way, we can assign Real ID to sensor nodes by using small size integers which is convenient to use rather than using big size integers. If necessary, we can easily create further Real ID of types like 2^n -2, 2^n + 2 and so on. For this, we have to add just 2 more bytes for every new type in the signature and pad zeros accordingly. Hence, we can assure that our approach is technically feasible for transmitting IDs of very large number of sensor nodes in data aggregation for WSNs.

4. Analytical Models

In this section, first we present analytical model for the data aggregation schemes and then for carrying maximum number of node ID by pre-defined payload of resource-constraint sensor node.

4.1 Analytical Model for Data Aggregation Schemes

Parameters	Descriptions	Parameters	Descriptions
P_{DP} or E_{DP}	Energy consumed by DP Scheme	N_{msg}	Number of message generated by DP per round
P_{HDA} or E_{HDA}	Energy consumed by HDA Scheme	E_{Rx}	Energy consumed by a node to *receive* data
P_{DD} or E_{DD}	Energy consumed by DD	E_{Tx}	Energy consumed by a node to *transmit* data
C	Number of source groups within a WSN	E_{Idle}	Energy consumed to be in idle state for a node
M, M'	Number of rows of WSN, the highest level of a source node	a	Energy dissipation to be in idle state
N	Number of columns of the WSN	β	Energy dissipation to transmit data
A_n	ID of an *Active path*	γ	Energy dissipation to receive data
level	WSN hierarchy level	X	Number of sources
m_j	Number of associated nodes to collect data per level	Y	Number of aggregation nodes
G_i	Source group	Z	Number of routing nodes
n_i	Number of source nodes in a group	r	One side coverage range of a parent
P_a	Communication overhead due to missing data aggregation	$n_{c,}$	Average no. of children per parent (network cardinality)
P_β	Communication overhead due to frequent transmission of parent nodes' energy information	n_p	Average no. of parents per child (network cardinality)
P_γ	Communication overhead for sending gradients from children to their parents	T_{NP}	Total number of parent nodes
$f1$	A ratio of sampling rate to frequency of attributes/parents' energy status sending	T_{NC}	Total number of children nodes
$f2$	A ratio of sampling rate to frequency of gradients set-up	W	Weight that represents excess number of messages than DP generates

Table 4. Parameters used in power consumption cost model.

The energy consumption issue for WSNs is the most important because the lifetime of a sensor node is extremely depends on the available energy of its battery. There are three domains to be considered regarding energy consumption: (i) sensing activity (data collection from the environment), (ii) communication (sending and receiving packets) and (iii) data processing/in-network data aggregation. Although all these activities waste energy, communication is responsible for the bulk of the power consumption which is the main point of attention in many algorithms designed for sensors networks. That is to say, energy saving by reducing the communication activity consequently increases WSN lifetime (Madden et al., 2005). Inspired by this notion, we design a mathematical cost model to compute how much power dissipates by our DP scheme in order to gather data with aggregation in WSN. In addition, we present the cost model in terms of the same metric for DD and HDA schemes. Table 4 lists the parameters used to design the power dissipation cost models.

4.1.1 Power Consumption by DP Scheme

We first divide the source nodes into different groups based on their positions in a WSN. This is done by determining how far they are located, in terms of hop count, from an active designated path. By using following equation we can know the number of groups of source nodes (C) for the given WSN.

$$C = \{\max(N - A_n, A_n - 1) + 1\} \times 1/r \tag{1}$$

Here, r is the one-side coverage range of a parent node and its value is determined during hierarchical multi-parent multi-child tree construction. For instance, in the Fig. 9, there are 48 sensor nodes (M=8 and N=6), a designated path P3 is active and the value of r equals 2. By substituting the values to parameters, we get

$$C = \{\max(6-3, 3-1) + 1\} \times 1/2 = \{\max(3, 2) + 1\} \times 1/2 = 2.$$

Therefore, source nodes can be divided into two groups, say group one is G1 (shown in dotted rectangle) and another is G2 (rest part of the network), as shown in Fig. 9. It means that the source nodes of G1 and G2 are located one hop and two hops away from P3, respectively.

The next step is to calculate the number of messages generated during data transmission from all of the source nodes to the sink node. The number of messages *Nmsg* can be calculated by using following expression.

$$N_{msg} = \sum_{i=1}^{C} G_i * n_i + (M' - 1) \tag{2}$$

As we can see in the Fig. 9, G1 and G2 consists of eight and two source nodes out of total ten source nodes (shown as dark colored nodes), respectively. Moreover, data from sources nodes of G1 and G2 need one hop and two hops to reach P3, respectively. If we substitute the values for the parameters, we can get *Nmsg* = (8×1 + 2×2) + (8-1) = 12+7 = 19. It is exactly the same number of messages generated (i.e., 19 solid arrows as shown in the Fig. 9) in the network.

Alternatively, there is another way to compute *Nmsg*. In this method, we simply use the number of all levels of WSN and associated number of sensor nodes in each level involved during data transmissions. Since each of the involved sensor nodes generates one message, the number of messages generated is equivalent to the number of the sensor nodes involved for data transmission. For this, we use following expression.

$$N_{msg} = \sum_{i=1}^{M} level_i \sum_{j=1}^{N} m_j \tag{3}$$

To prove the correctness of this expression, we can substitute the values for its parameters in the Fig.9. In this calculation, we put the value of involved sensor nodes in the decreasing order of level, i.e., starting from level M (in this case M=8) to 1. Then, we can get *Nmsg* = (3+2+3+2+3+3+2+1) = 19. Out of the 19 nodes, 10 nodes are source nodes (X) and 5 nodes are aggregation nodes (Y) which receive more than one message and partially aggregate data. The rest 4 nodes are routing nodes (Z) which just forward the incoming message to their parents. Hence, the number of messages generated in WSN is the sum of the source nodes, aggregation nodes and routing nodes involved during data transmissions.

Mathematically, we can express it as *Nmsg* = X+Y+Z. Since both of the methods result the same number of messages one method verifies the correctness of another and vice-versa.

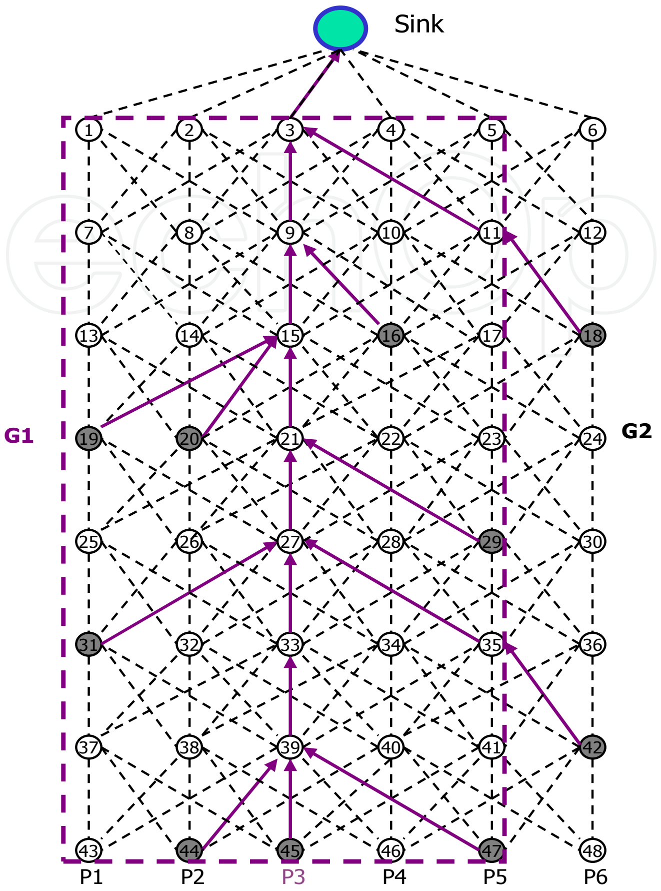


Fig. 9. Two groups of source nodes (G1 and G2).

For a given M×N WSN, the energy dissipation can be defined as the sum of the energy consumed by four types of nodes involved during data transmission to the sink node which are: sensor nodes being in the idle state, source nodes, aggregation nodes and routing nodes, and this can be calculated as below.

$$E_{DP} = (M \times N) \times E_{Idle} + \sum_{s=1}^{numSources} (E_{Tx}) + \left(\sum_{m=1}^{numAgrNodes} \left(\sum_{l=1}^{numSampleRcv} E_{Rx} + E_{Tx} \right) \right) + \sum_{n=1}^{numRouteNodes} (E_{Rx} + E_{Tx}) \tag{4}$$

The first part of the right hand side of the expression is the energy required for all of the sensor nodes of the M×N WSN which are in the idle state. The second part gives the energy consumed by the sources nodes. The third part measures summation of the energy dissipated by each aggregation node. The second summation notation of the third part counts the number of received messages by an aggregation node. The fourth and the final part gives the energy required to receive and transmit a message for routing nodes. By using the notations of the Table 4, we can deduce the above expression as follow.

$$\begin{aligned} E_{DP} &= (M \times N) \times \alpha + X \times \beta + Y(\gamma + \beta) + Z(\gamma + \beta) \\ &= (M \times N)\alpha + (X + Y + Z)\beta + (Y + Z)\gamma \\ &= (M \times N)\alpha + N_{msg} \times \beta + (N_{msg} - X)\gamma = P_{DP} \end{aligned} \tag{5}$$

This is the cost model which can compute the power dissipation by our DP scheme while collecting data in WSNs.

4.1.2 Power Consumption by HDA Scheme

HDA requires more power than our DP due two factors. The first one is that HDA frequently misses data aggregation and thus more number of messages is generated, due to the involvement of the many sensor nodes to forward data to the sink node. When we denote this extra communication overhead by weight factor W, in terms of number of messages, the power dissipated by HDA can be given as follow.

$$P_\alpha = W \times (E_{Rx} + E_{Tx}) \tag{6}$$

The second factor is that, in HDA, parent nodes have to frequently notify their energy-status/best-attributes/interests to their child nodes so that the child nodes can determine appropriate parent nodes for forwarding data to the sink node. Therefore, each parent node transmits a message to its child nodes and each of the child nodes has to receive the same number of messages as the number of its parent nodes, due to the multi-parent multi-child hierarchy tree structure. But our DP can avoid such type of unnecessary traffic during data transmission because every node has data gathering application knowledge. We can compute this messages overhead of HDA mathematically, as shown below.

Total number of parent nodes: T_{NP} = (M-1)×N
Total number of child nodes: T_{NC} = N + (M-1) × N × n_c

Hence, the power dissipation to transmit a message by many parents ($P_{\beta1}$) and that to receive a message by many child nodes ($P_{\beta2}$) are given below. Here, f_1 is the ratio of sample rate to the frequency of notifying/receiving energy-status/best-attributes.

$P_{\beta1}$ = ((M-1) ×N × E_{TX}) ×1/f_1
$P_{\beta2}$ = (N + (M-1) × N × n_c × E_{RX}) × 1/f_1

By combining above two expressions, we get,

P_β = $P_{\beta1}$ + $P_{\beta2}$ = (((M-1) ×N× E_{TX}) ×1/f1) + (N + ((M-1) × N × n_c × E_{RX}) × 1/f_1)
= ((M-1) ×N × E_{TX} + N + (M-1) × N × n_c × E_{RX}) × 1/f_1
= N ((M-1) (E_{TX} + n_c × E_{RX}) +1) × 1/f_1.

As a result, the total power dissipation by HDA for data transmission to the sink node can be computed as below.

$$P_{HDA} = P_{DP} + P_\alpha + P_\beta \tag{7}$$

4.1.3 Power Consumption by DD Scheme

In the DD scheme, there are three more factors responsible for power consumption than that of DP scheme. Because the first two factors are the same as those of HDA, we just use them here. The third factor is that, in DD, each child node sends gradients to its all parent nodes in the response of frequently received interests from parent nodes. We derive the cost of gradients as below.

Total number of parent nodes: TNP = $(M\text{-}1) \times N \times n_p$
Total number of child nodes: $T_{NC} = M \times N$

Hence, the power dissipation to receive a gradient by many parents ($P_{\gamma 1}$) and that to transmit a gradient by many child nodes ($P_{\gamma 2}$) are as follows. Here f_2 is the ratio of sample rate to the frequency of receiving/sending gradients.

$P_{\gamma 1} = ((M\text{-}1) \times N \times n_p \times E_{RX}) \times 1/f_2$
$P_{\gamma 2} = (M \times N) \times E_{TX} \times 1/f_2$

By combining above two expressions, we get,

$$
\begin{aligned}
P_{\gamma} &= P_{\gamma 1} + P_{\gamma 2} \\
&= (M\text{-}1) \times N \times n_p \times E_{RX} \times 1/f_2 + (M \times N) \times E_{TX} \times 1/f_2 \\
&= N\,((M\text{-}1) \times n_p \times E_{RX} + M \times E_{TX}) \times 1/f_2
\end{aligned}
$$

As a result, the total power dissipation by DD for data transmission to the sink node can be by using following expression.

$$P_{DD} = P_{DP} + P_{\alpha} + P_{\beta} + P_{\gamma} \tag{8}$$

In summary, above analytical model shows that our DP scheme is an energy efficient scheme to aggregate data in WSN because it can aggregate data efficiently by avoiding unnecessary traffics during data transmissions.

4.2 Analytical Model for Sending ID of Sensor Nodes

As we mentioned earlier, communication is responsible for the bulk of the power consumption in WSNs. The limited power of sensor nodes can be saved by reducing communication overhead so that the lifetime of WSNs can be prolonged. There are many ways to reduce the communication overhead in WSNs. Some of them are: minimizing generation of messages in the network, shortening duty cycling and determining small size packet. Former two processes are applications dependent in WSNs whereas determining small size packet, in the case of low powered sensor nodes (Mica Motes), is controlled by TinyOS, an operating system that runs motes hardware. For Mica Motes, TinyOS predefined a 36-byte packet out of which 29-byte is allocated to the payload. With the commence of in-network data processing for WSNs, aggregation of sensor data became popular because data aggregation can reduce the number of data transmissions to the sink node by combining correlated sensor data . But, in many applications, data aggregation in WSNs needs the sink node to acquire knowledge of the contributed sensor nodes so that the sink node can compute actual result of aggregated data. This requirement creates a problem of sending IDs of participated sensor nodes to the sink node for larger size WSNs because the payload is of limited size. In this section, we present an analytical model for sending IDs of the contributed sensor nodes to the sink node for the existing CMT and our schemes. We assume that N is the total number of sensor nodes of a sub-tree rooted at the sink node in a WSN. We also assume that N_{cl} and N_{ncl} are the lists of contributing nodes and the list of non-contributing nodes of the WSN respectively. Hence, $N=N_{cl}+N_{ncl}$, where $Ncl < N_{ncl}$.

4.2.1 CMT Scheme

In this method, each node ID is considered as a plaintext (2-byte) and all the IDs are concatenated while sending to the sink node. Out of the fixed 29 bytes payload, an encrypted sensor data uses 4 bytes leaving 25 bytes as free space for carrying IDs. Therefore, the number of sensor node IDs can be included in the list of N_{cl} is 12 while sending the aggregated data to the sink node. For the CMT scheme, the value for scalability in terms of carrying IDs is $O(N_{cl})$=12.

4.2.2 Signature Scheme

On the other hand, since we superimpose signatures of sensor node IDs, a single bit is enough to hold ID of a sensor node. Therefore, for the available 25 bytes free space of the payload, our scheme can include 25×8 = 200 sensor node IDs in the list of N_{cl} while sending the aggregated data to the sink node. Hence, for our scheme, the value for scalability in terms of carrying IDs is $O(N_{cl})$ = 200.

This analytical model shows that, if necessary, our scheme can transmit around 16 times more number of sensor node IDs than does the CMT scheme. Therefore, our scheme is obviously a scalable one to apply in such data aggregation applications for WSNs that need the information of contributed sensor nodes at the sink node e.g., privacy preserving data aggregation for WSNs.

5. Analytic Performance Evaluation

Based on the previous mathematical models, first we compare the performance of DP scheme with HDA and DD schemes in terms of energy dissipation required to collect data from WSNs and then compare the performance of our signature scheme with CMT scheme in terms of energy efficiency and scalability in order to transmit IDs of sensor nodes to the sink node.

5.1 Analytic Performance Evaluation of DP, HDA and DD Schemes

We consider the scenario where the frequency of attributes/parents-energy-status/gradients sending is once per 50 seconds as in HDA. We use such parameters as idle-time power dissipation of 35 mW, receiving power dissipation of 395 mW, and transmitting power dissipation of 660 mW, as presented in DD. The sampling rate is one sample per second. For this evaluation, we study on the impacts of *network size*, *the number of source nodes* and *network cardinality* over the energy consumption.

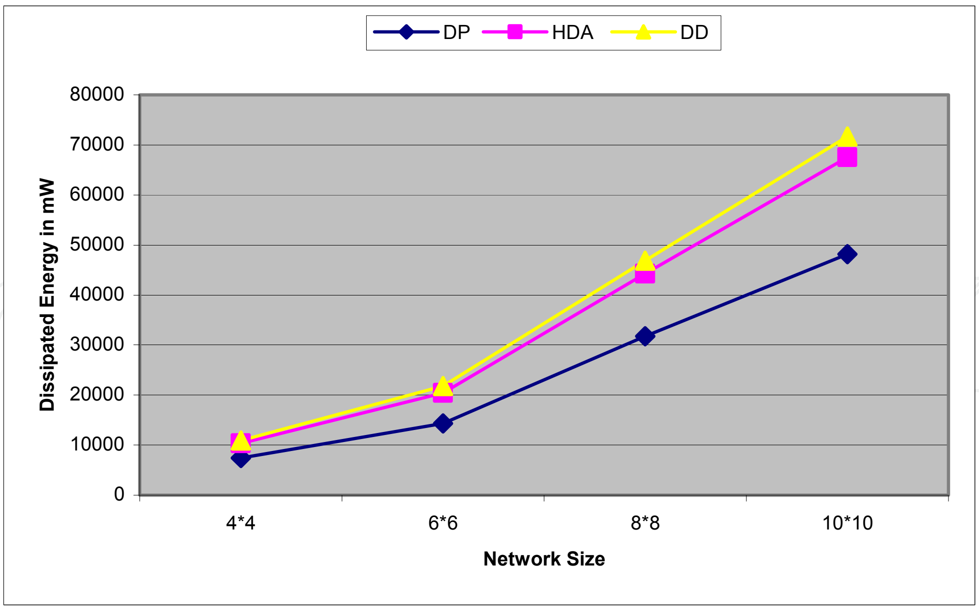


Fig. 10. Energy consumption for varying network size.

(a) Network size: For this, the density of source nodes is fixed to 25% of sensor nodes from different sizes of WSNs. In Fig. 10, it is shown that the performances of all the three schemes DP, HDA and DD are decreased as the size of the network increases from 4×4 to 10×10. This is because as the size of a network increases, the number of source nodes also increases. As a result, the number of generated messages increases during data transmissions in the networks. Consequently, a larger WSN consumes much amount of energy than a smaller one. However, the performance of our DP scheme is always better than both of HDA and DD schemes. It is because DP scheme generates less number of messages in the networks by avoiding unnecessary traffics generation during data transmissions to the sink node. Moreover, as the size of network increases, the performance gap between DP and HDA as well as that between DP and DD get wider. It indicates that data aggregation scalability of our scheme is better than both HDA and DD schemes.

(b) Source nodes: We change the density of the sources nodes from 10 to 50 for a fixed size 10×10 WSN. In Fig. 11, it is shown that as the number of source nodes increases from 10 to 50, the amount of dissipated energy for transmitting data to the sink node also increases for all DP, HDA and DD schemes. The reason is that a larger number of source nodes means that the network generates more number of messages and it needs larger amount of energy to transmit them. However, as the number of source nodes increases, the rate of increase in the amount of the dissipated energy is lower for DP scheme than both HDA and DD schemes. In this way, the performance of the DP scheme improves further for higher number of source nodes in a WSN. It justifies the efficiency of DP scheme to aggregate data in WSNs.

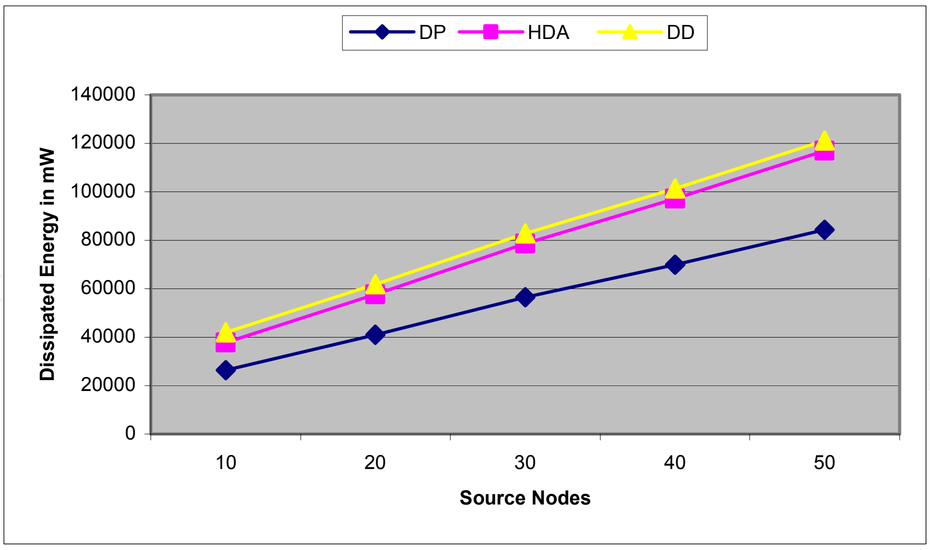


Fig. 11. Energy consumption for varying source nodes.

(c) Network cardinality: The network size and the number of source nodes are fixed to a 10×10 WSN and 15% of sensor nodes respectively. We change network cardinality from 3 to 5 as shown in Fig. 12. The cardinality of a network means an average number of child nodes and parent nodes per sensor node in the WSN and it is determined during the construction of the multi-parent-multi-child hierarchical network structure. The Fig. 12 depicts that our DP scheme has better performance than HDA and DD schemes although the amount of dissipated energy for all the three schemes decreases when the network cardinality increases. This is because the coverage of sensor nodes increases with the increase in the network cardinality. As a result, the number of messages generated in the network is reduced while transmitting data to the sink node.

Above analytical performances show that proposed DP scheme is a more energy efficient scheme to aggregate data in WSNs than HDA and DD schemes.

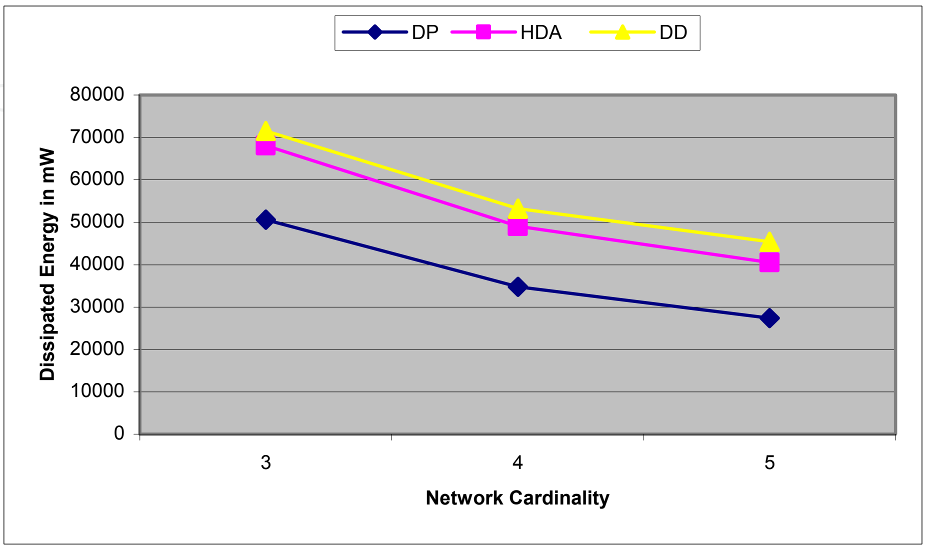


Fig. 12. Energy consumption for varying network cardinality.

5.2 Analytical Performance Evaluation of CMT and Signature Schemes

In this section, we show the efficiency of our scheme by comparing it with the CMT scheme considering transmissions of IDs of contributed sensor nodes along with aggregated data to the sink node. The CMT scheme is the standard work that deals with sending IDs of sensor nodes to the sink node for WSNs. We present the performance results of both schemes in terms of four metrics: *scalability, energy consumption, payload size* and *computation overhead*.

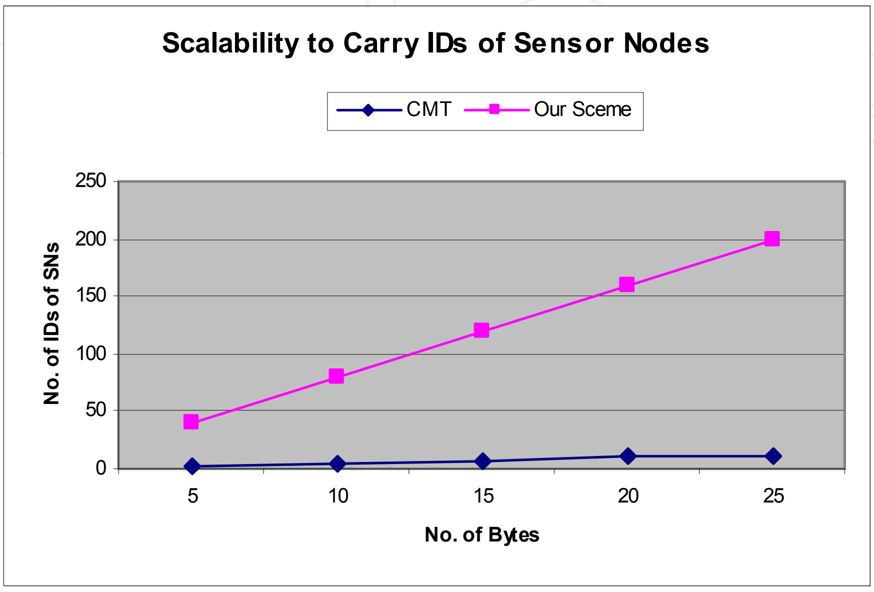


Fig. 13. Carrying IDs of sensor nodes by Our and CMT schemes.

(a) Scalability: For TinyOS based Mica Motes, the maximum payload size is of 29-byte. We assume each of sensor data and a key is of 2-byte. Therefore, the remaining maximum free space of the payload is 25-byte. The scalability measure is given in terms of IDs of how many sensor nodes can be sent by using the available limited free space (25-byte) by both schemes. As shown in Fig. 13, for the given limited 25-byte free space, our scheme can send IDs of up to 200 sensor nodes while transmitting aggregated sensor data to the sink node. On the other hand, the CMT scheme is unable to send IDs of more than 12 sensor nodes. The reason is that our scheme can hold ID of a sensor node just by a single bit whereas the CMT scheme needs 2-byte for the same task. Therefore, it is obvious that our scheme is much more (about 16- time) scalable than the CMT scheme in terms of carrying the number of IDs of sensor nodes in the course of transmitting aggregated value to the sink node in WSNs.

(b) Energy consumption: In this measure, we consider the amount of energy required to transmit and receive a packet by a sensor node. This is calculated as given in (Bi et al., 2007). The total energy (E_{Total}) to communicate a packet is calculated by adding transmission energy (E_{Tx}) and receiving energy (E_{Rx}) as below.

$$E_{Tx} = L \times E_{elec} + L \times \varepsilon \times d^2 \quad (9)$$

$$E_{Rx} = L \times E_{elec} \quad (10)$$

$$E_{Total} = E_{Tx} + E_{Rx} \quad (11)$$

where, L is the length of the packet in bits, E_{elec} is electronic energy (= 1.16 μJ/bit), the parameter ε = 5.46 pJ/bit/m^2, and d is crossover distance (= 40.8 m).

Table 5 illustrates energy efficiency of our scheme over the CMT scheme to communicate a packet which consists of 2-byte sensor data, 2-byte key and IDs of 12 sensor nodes. To achieve this, our scheme dissipates just about 36% of that energy which is required by the CMT scheme. This is because our scheme needs less number of bytes than that of CMT scheme to transmit the packet with aforementioned features. By saving the precious energy of sensor nodes In this way, our signature scheme can enhance the lifetime of WSNs.

Method	Energy Dissipation in mJ	Energy Gain Ratio
CMT	0.670778	63.88%
Our Scheme	0.242225	

Table 5. Energy consumption by a packet to carry an encrypted data along with IDs of 12 sensor nodes.

(c) Payload size: We measure this in terms of bytes required to send different number of IDs of sensor nodes along with an encrypted aggregated sensor data (4-byte) to the sink node. In Fig. 14, it is shown that our scheme needs only 5-byte to send IDs of up to eight sensor nodes with the encrypted data and it adds one more byte for every additional ID of up to eight sensor nodes. On the other hand, the CMT scheme needs 2 more bytes for each additional sensor node ID. Therefore, the size of payload in the CMT scheme is directly proportional to the number of IDs of sensor nodes. For instance, to send IDs of 12 sensor nodes with their encrypted aggregated value, our signature scheme needs just 6-byte (4-byte for encrypted aggregated value and 2-byte for carrying IDs of 12 sensor nodes) payload whereas the CMT scheme needs 28- byte (4-byte for encrypted aggregated value and 24-byte for carrying IDs of 12 sensor nodes). In this way, our signature scheme reduces the size of payload greatly. As a result, the proposed signature scheme not only reduces the packet communication cost but also decreases the message loss rate because the probability of message interference is higher for larger size messages (Muller et al., 2007).

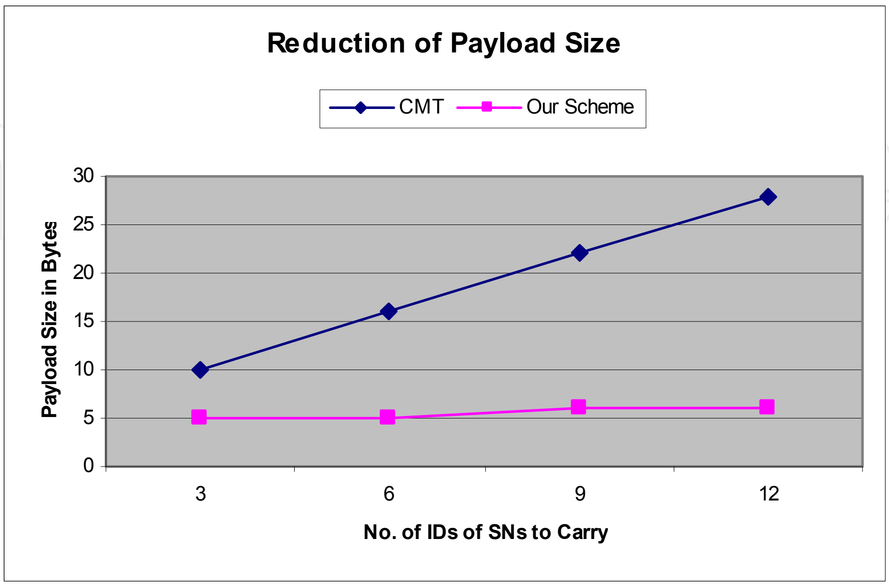


Fig. 14. Variation of payload size with increasing number of node ID.

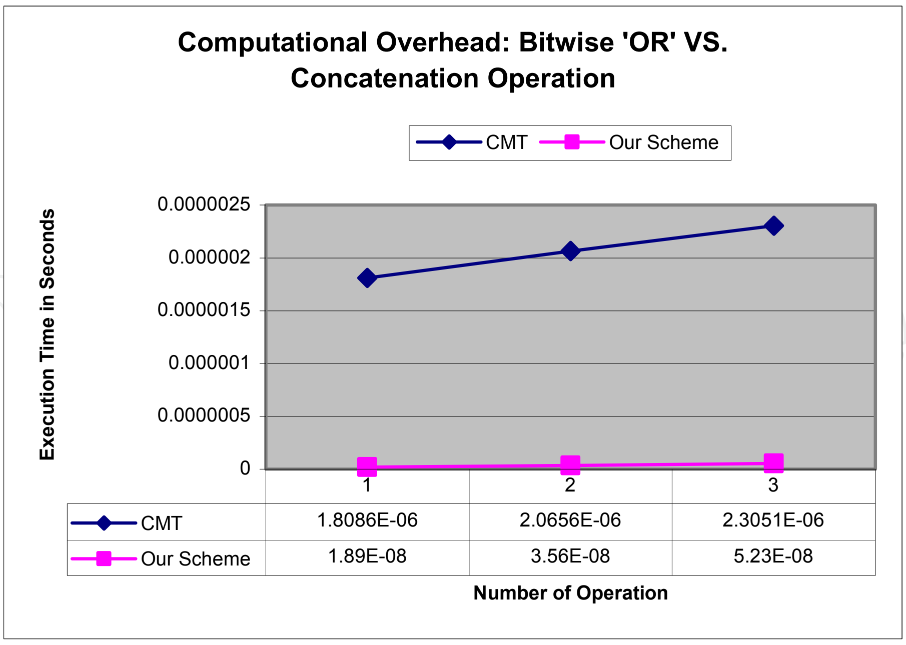


Fig. 15. Computational efficiency of Our scheme over CMT scheme.

(d) Computation overhead: We measure execution time required to: i) concatenate IDs of sensor nodes (plaintexts) in the case of the CMT scheme and ii) superimpose IDs of sensor nodes in our scheme. We use MATLAB® 7.6.0.324 (R14) to compute the execution time. In this experiment, we consider the execution time required for one, two and three concatenation and bitwise *OR* operations to combine IDs of two, three and four sensor nodes (each ID is of 2-byte size, a positive integer type) for the CMT and our scheme respectively. In Fig. 15, it is shown that the execution time of our approach to combine IDs of sensor nodes is always faster than that of the CMT scheme by an order of two-magnitude. The reason is that our scheme uses bitwise *OR* operation to combine signatures of node IDs. Needless to say that the bitwise operation is the fastest one among all available operations for a processor.

6. Performance Evaluation

In this section, by using TOSSIM (Levis et al., 2003) simulator, we evaluate the performances of our DP scheme comparing with HDA and DD schemes, in terms of dissipated energy. We consider the scenario where the frequency of attributes/parents-energy-status/gradients sending is once per 50 seconds as in HDA. We use such parameters as packet receiving, packet transmitting and data aggregation for power dissipation. The sampling rate is one sample per second. We study on the impacts of network size, the number of source nodes and network cardinality over the energy consumption. We consider the same network scenarios for simulations as we did in the previous section for all the three analytic evaluations.

(a) Network size: Similar to the analytic performance, Fig. 16 shows that our DP scheme requires less amount of energy than HDA and DD schemes to collect data from different size WSNs. It is because our DP scheme generates less number of messages in the networks

by avoiding unnecessary traffics generation during data transmissions to the sink node. Moreover, as the size of network increases, the performance gap between DP and HDA schemes as well as that between DP and DD schemes get wider. It indicates that, in of our DP scheme, data aggregation efficiency improves further with the increasing size of the networks.

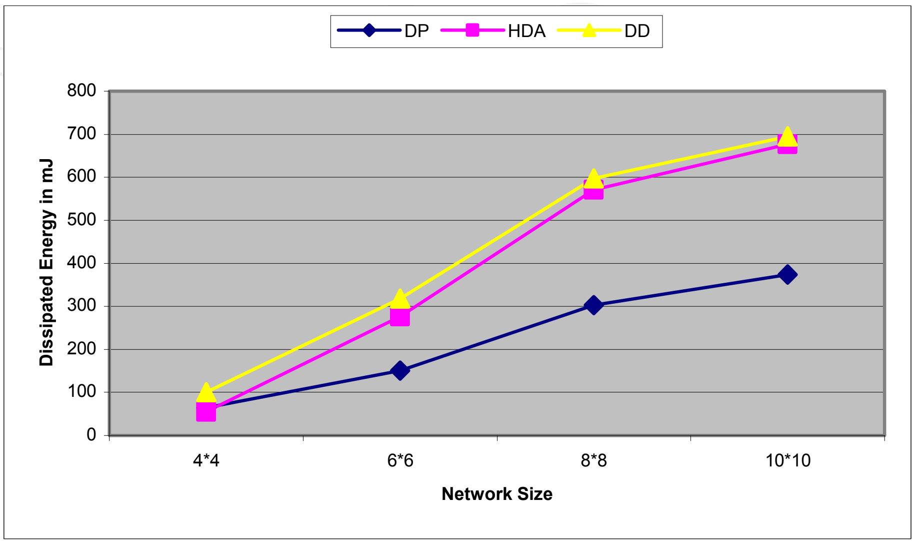


Fig. 16. Energy consumption for varying size of WSN when source nodes are fixed to 25% of the sensor nodes.

(b) Source nodes: Similar to the analytic performance, Fig. 17 shows that our DP scheme always require less amount of energy to aggregate data than HDA and DD schemes when the number of source nodes in a WSN varies. In addition, the rate of increase in the amount of the dissipated energy improves further in DP scheme with the increasing number of source nodes in a WSN. The reason is that, unlike HDA and DD schemes, DP scheme doesn't generate extra traffics and it guarantees data aggregation in WSNs.

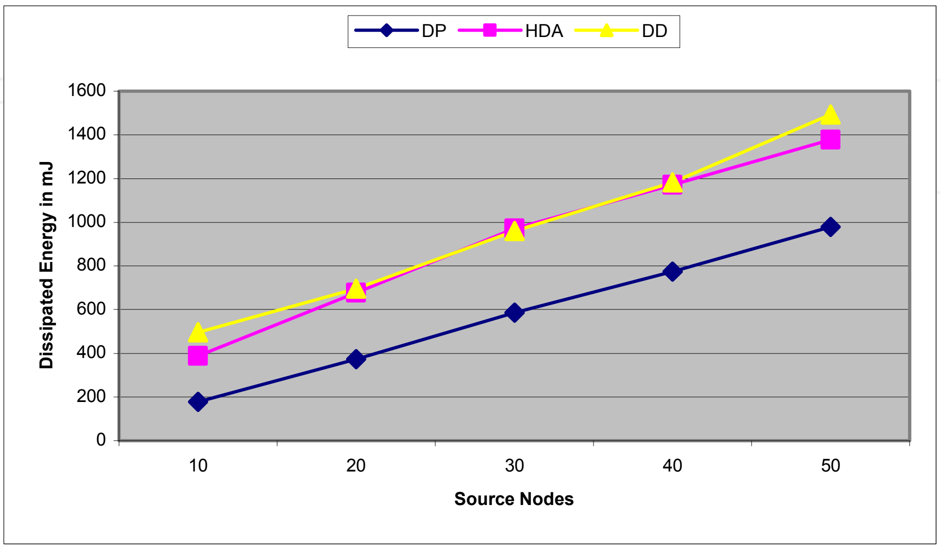


Fig. 17. Energy consumption for varying source nodes in a 10×10 WSN.

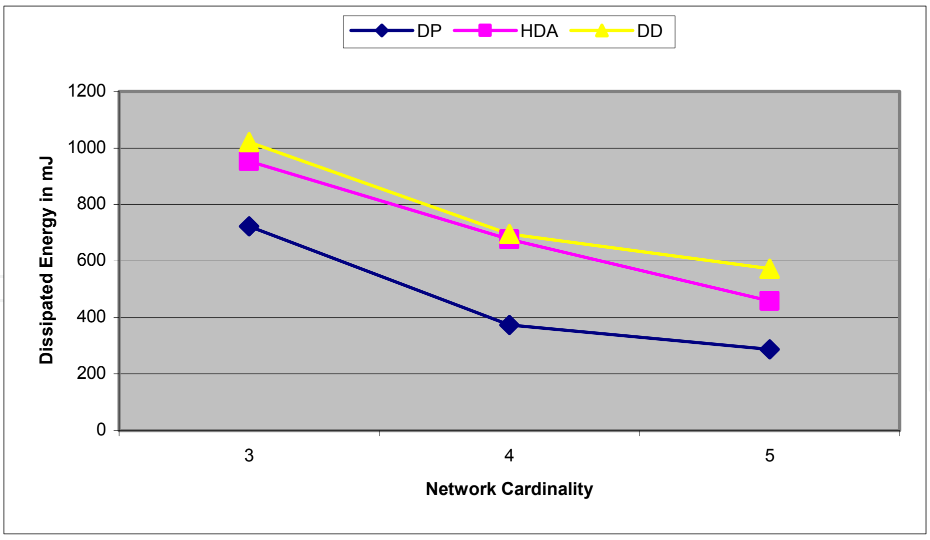


Fig. 18. Energy consumption for varying network cardinality when source nodes are fixed to 15% of sensor nodes in a 10×10 WSN.

(c) Network cardinality: Fig. 18 depicts that when the network cardinality increases the amount of dissipated energy for data transmissions to the sink node decreases for all DP, HDA and DD schemes. This is because with the increase in the network cardinality, the coverage range of each node also increases. As a result, it reduces the total number of messages in the network and so does the dissipated energy. As above analytical performance evaluation, the performance of our DP scheme is always better than those of HDA and DD schemes for varying network cardinality. The reason is that, in DP scheme, all sensor nodes utilize data aggregation application knowledge for when and where to send data during their transmissions to the sink node. However, on the one hand, a larger value for network cardinality gives more energy efficiency to a WSN; but on the other hand, increasing data transmission rage of sensor nodes costs much energy. Therefore, there must be a reasonable trade-off of the network cardinality over the data transmission range. For this time, we would like to keep this issue as our future work.

7. Conclusion and Future Work

In this chapter, we proposed two energy efficient schemes for resource-constraint WSNs. First, we proposed DP scheme as energy efficient data aggregation for WSNs in which a pre-determined set of paths is run in round-robin-fashion in order to tackle the unnecessary traffics and hotspot problem of the conventional data aggregation schemes which always drive data flow towards the sink node/s. In our DP scheme, all sensor nodes participate in gathering all the sensed data and transferring them to the sink node. Because all the nodes in the network are charged for the heavy workload, we believe that the sensor nodes consume their energy almost equally and the hotspot problem can be significantly relieved. In addition, DP scheme avoids unnecessary traffics during data transmissions to the sink node by utilizing data aggregation application knowledge. Moreover, unlike both DD and HDA schemes, DP scheme can be used for continuous data delivery for event-driven applications because unnecessary traffics do not intervene during data collection processes.

The presented analytical performance evaluations and simulation results have similar trends to achieve energy efficiency. Both of them show that DP scheme is more energy efficient for aggregating data in WSNs and hence it can prolong the lifetime of resources-constraints WSNs than HDA and DD schemes. Second, we propose a novel scheme called signature scheme in order to efficiently transmit IDs of a large number of sensor nodes along with aggregated sensor data to the sink node. In our signature scheme, first, the sink node generates a unique signature for the Real ID of every sensor node. Then, parent nodes (data aggregators) superimpose the signatures of their child nodes including their own signatures and transmit the superimposed signatures along with aggregated data to the sink node. For this, a single bit is enough to hold the information of a sensor node. Through analytical performance evaluations, we have shown the efficiencies of the signature scheme over the existing work in terms of scalability, energy consumption, payload size and computation overhead.

Transmitting IDs of contributed sensor nodes along with sensed data is mandatory for many applications designed for WSNs. Therefore, as our future work, first we would like to show simulation results of the signature scheme and then we will mingle DP scheme with signature scheme in order to provide further more energy efficient scheme to collect data in WSNs. In addition, we would like to apply our combined scheme to arbitrary types of WSN and networks with multiple sink nodes.

8. Acknowledgment

This research was financially supported by the Ministry of Education, Science Technology (MEST) and Korea Institute for Advancement of Technology (KIAT) through the Human Resource Training Project for Regional Innovation. This work was also supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MEST) (No. 2010-0000202).

9. References

Akkaya, K. & Younis, M. (2005). A survey on routing protocols for wireless sensor networks, Ad Hoc Networks 3 (2005) pp. 325-349.

Akyildiz, I.; Su, W.; Sankarasubramaniam, Y. & Cyirci, E. (2002). Wireless sensor networks: a survey, In Computer Networks 38 (4) (2002), 393–422.

Bi, Y.; Li, N. & Sun, L. (2007). DAR: An energy-balanced data-gathering scheme for wireless sensor networks, In Computer Communication 30 (2007) 2812-2825.

Bista, R.; Kim, Y-K. & Chang, J-W. (2009). A New Approach for Energy-Balanced Data Aggregation in Wireless Sensor Networks, In CIT09, cit, vol. 2, pp. 9-15.

Bista R., Chang J-W. Privacy-Preserving Data Aggregation Protocols for Wireless Sensor Networks: A Survey, Sensors 2010, 10(5) : 4577-4601.

Castelluccia, C.; Mykletun, E. & Tsudik, G. (2005). Efficient aggregation of encrypted data in wireless sensor networks, In MobiQuitous, pp. 109–117, 2005.

Considine, J.; Li, F.; Kollios, G. & Byers, J. (2004). Approximate aggregation techniques for sensor databases, In Proceedings of ICDE, pp. 449-460, April, 2004.

Conti, M.; Zhang, L.; Roy, S.; Pietro, R-D.; Jajodia, S. & Mancini, L-V. (2009). Privacy-preserving robust data aggregation in wireless sensor networks, Security and Communication Networks, 2009; 2:195–213.

Dijkstra, E-W. (1959). A Note on Two Problems in Connection with Graphs, Numeriche Mathematik, Vol. 1 (1959) pp. 269-271.

Girao, J.; Westhoff, D. & Schneider, M. (2005). CDA: Concealed Data Aggregation for Reverse Multicast Traffic in Wireless Sensor Networks, In ICC 2005, Vol.5, pp. 3044-3049.

He, W.; Liu, X.; Nguyen, H.; Nahrstedt, K. & Abdelzaher, T. (2007). Pda: Privacy-preserving data aggregation in wireless sensor networks, In Proceeding of INFOCOM, pp. 2045–2053, 2007.

Heinzelman, W-R.; Kulik, J. & Balakrishman, H. (1999). Adaptive protocols for information dissemination in wireless sensor networks, In Proceedings of MOBICOM, pp. 174–185, August, 1999.

Heinzelman, W.; Chandrakasan, A. & Balakrishnan, H. (2000). Energy-efficient communication protocols for wireless microsensor networks, In Proceedings of HICSS, January, 2000.

Hill, J.; Szewczyk, R.; Woo, A.; Hollar, S.; Culler, D-E. & J.Pister, K-S. (2000). System Architecture Directions for Networked Sensors, In ASPLOS, pp. 93–104, 2000. TinyOS is available at http://webs.cs.berkeley.edu.

Horton, M.; Culler, D.; Pister, K.; Hill, J.; Szewczyk, R. & Woo, A. (2002). MICA the commercialization of micro sensor motes, In IEEE Sensors J., April 2002, 19(4): 40-48.

Itanagonwiwat, C.; Govindan, R. & Estrin, D. (2002a). Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks, In Proceedings of MOBICOM, pp. 56-67, 2002.

Itanagonwiwat, C.; Estrin, D.; Govindan, R. & Heidemann, J. (2002b). Impact of Network Density on Data Aggregation in Wireless Sensor Networks, In Proceedings of the 22nd ICDCS, pp. 457-458, 2002.

Levis, P.; Lee, N.; Welsh, M. & Cullar, D. (2003). TOSSIM: Accurate and scalable simulation of entire TinyOS applications,
http://www.cs.berkely.edu/~pal/research/tossim.html.

Madden, S.-R.; Franklin, M.-J.; Hellerstein, J.-M. & Hong, W. (2002). TAG: a tiny aggregation service for ad hoc sensor networks, In Proceedings of the OSDI02, pp. 1-16, December, 2002.

Madden, S.-R.; Franklin, M.-J.& Hellerstein, J.-M. (2005). TinyDB: an acquisitional query processing system for sensor networks, ACM TDS 30 (1) (2005), pp.122–173.

Mueller, R.; Kossmann, D. & Alonso, G. (2007). A Virtual Machine for Sensor Networks, In EuroSys'07, pp. 145-158, March 2007.

Pottie, G-J. & Kaiser, W-J. (2000). Wireless integrated network sensors, Communications of ACM, May 2000.

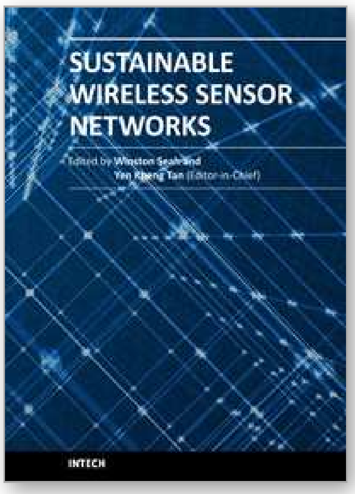
Yao, Y. & Gehrke, J. (2003). Query processing for sensor networks, In Proceedings of the CIDR 2003.

Yick, J.; Mukherjee, B. & Ghosal, D. (2008). Wireless sensor network survey, In Computer Networks, 2008, 52(12): 2292-2330.

Zhang, W-S.; Wang, C. & Feng, T-M. (2008). GP2S: generic privacy-preservation solutions for approximate aggregation of sensor data, concise contribution, In Proceedings of PerCom, pp.179–184, 2008.

Zhou, B.; Ngoh, L. H.; Lee, B. S. & Fu, C-P. (2006). HDA: A hierarchical Data Aggregation Scheme for Sensor Networks, Computer Communication 29 (2006) 1292-1299.

Zobel, J.; Moffat, A. & Ramamohanarao, K. (1998). Inverted Files versus Signature File for Text Indexing, In ACM TDS, Vol. 23, No. 4, 1998, pp. 453-490.


SUSTAINABLE WIRELESS SENSOR NETWORKS
Edited by Winston Seah and Yen Kheng Tan (Editor-in-Chief)
INTECH

Sustainable Wireless Sensor Networks

Edited by Yen Kheng Tan

ISBN 978-953-307-297-5
Hard cover, 574 pages
Publisher InTech
Published online 14, December, 2010
Published in print edition December, 2010

Wireless Sensor Networks came into prominence around the start of this millennium motivated by the omnipresent scenario of small-sized sensors with limited power deployed in large numbers over an area to monitor different phenomenon. The sole motivation of a large portion of research efforts has been to maximize the lifetime of the network, where network lifetime is typically measured from the instant of deployment to the point when one of the nodes has expended its limited power source and becomes in-operational â€“ commonly referred as first node failure. Over the years, research has increasingly adopted ideas from wireless communications as well as embedded systems development in order to move this technology closer to realistic deployment scenarios. In such a rich research area as wireless sensor networks, it is difficult if not impossible to provide a comprehensive coverage of all relevant aspects. In this book, we hope to give the reader with a snapshot of some aspects of wireless sensor networks research that provides both a high level overview as well as detailed discussion on specific areas.

Rabindra Bista and Jae-woo Chang (2010). Energy Efficient Data Aggregation for Wireless Sesor Networks, Sustainable Wireless Sensor Networks, Yen Kheng Tan (Ed.), ISBN: 978-953-307-297-5, InTech, Available from: http://www.intechopen.com/books/sustainable-wireless-sensor-networks/energy-efficient-data-aggregation-for-wireless-sesor-networks

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821