International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC) Vol.4, No.1, February 2013

DESIGNING LOGICAL TOPOLOGY FOR WIRELESS SENSOR NETWORKS: A MULTI-CHAIN ORIENTED APPROACH

Quazi Mamun

School of Computing and Mathematics, Charles Sturt University, NSW, Australia
qmamun@csu.edu.au

ABSTRACT

An optimal logical topology of a wireless sensor network (WSN) facilitates the deployed sensor nodes to communicate with each other with little overheads, lowers energy consumption, lengthens lifetime of the network, provides scalability, increases reliability, and reduces latency. Designing an optimal logical topology for a WSN thus needs to consider numerous factors. Chain oriented topologies have been found to offer a number of improvements in energy consumptions, lifetime, and load balancing than other topologies of WSNs. However, they usually suffer from latency, scalability, reliability and interference problems. In this paper, we present a chain oriented logical topology, which offers solutions to those problems. The proposed topology is designed such that it retains the advantages of the chain oriented topologies, and at the same time, overcomes the problems of the chain oriented topology such as latency, scalability, and data reliability. The proposed topology provides a communication abstraction, which can be easily used to devise a range of application protocols. Moreover, the logical topology offers node management, resource management, and other services. The performance of the proposed topology is compared with other topologies in respect to total energy consumption and lifetime of the network.

KEYWORDS

Wireless sensor network, chain oriented network, multi-chain, logical topology, topology management.

1. INTRODUCTION

Wireless sensor networks (WSNs) are formed by a large collection of power-conscious wireless capable sensors without the support of pre-existing infrastructure, possibly by unplanned deployment. With the sheer number of sensor nodes, their unattended deployment and hostile environment very often preclude reliance on physical configuration or physical topology. It is, therefore, often necessary to depend on the logical topology. The logical topology of a wireless sensor network is formed by the communication graph of the network. A communication graph of a WSN is an undirected graph $G = (V; E)$ where V denotes the sensors deployed, and E denotes the available communication links among the sensor nodes. As logical topology inherently defines the type of routing paths, indicates whether to use broadcast or unicast, and determines the sizes and types of packets and other overheads, choosing the right topology helps to reduce the amount of communication needed for a particular problem. Thus energy can be saved. An efficient topology, which ensures that neighbours are at a minimal distance, reduces the probability of message being lost between sensors. A topology can also reduce the radio interference, thus reducing the waiting time for sensors to transmit data [1–3]. Moreover, topology facilitates data aggregation, which greatly reduces the amount of processing cycles and energy, resulting in a longer lifetime for the network [4,5]. In addition, topology inherently defines the size of a group, how to manage new members in a group, and how to deal with members who have left the group. With the awareness of the underlying network topology,

DOI : 10.5121/ijasuc.2013.4101

more efficient routing or broadcasting schemes can be achieved. Furthermore, the network topology in WSNs can be changed by varying the nodes' transmitting ranges and also by adjusting the wake / sleep schedule of the nodes [6,7]. Therefore, more energy can be saved if the network topology is maintained in an optimal manner.

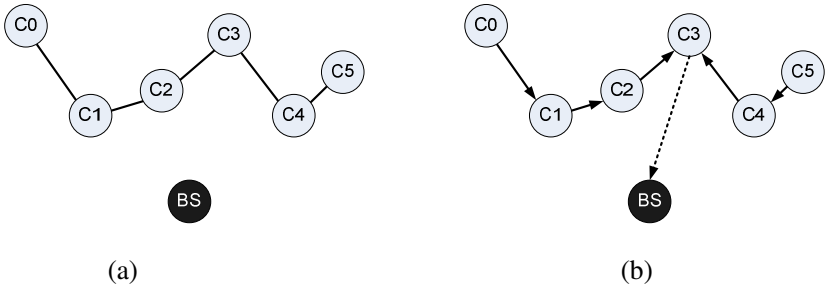
Additionally, much research has taken place to justify the performance of different logical topologies [8–11].Chain oriented topology has been identified as being more promising than other topologies of WSNs [12–17]. Chain oriented topologies minimize many of the constraints of WSNs. For example, energy consumptions by the sensor nodes can be greatly reduced by the chain oriented topology [18–21]. For data fusion/aggregation, chain oriented topology offers substantial advantages due to the logical structure of the sensor nodes [22,23]. It is also possible to obtain collision-free transmissions using a chain oriented topology [24]. Other WSNs requirements, such as connectivity, robustness, scalability, responsiveness, and reliability can also be enhanced by chain oriented topologies.

To achieve the above mentioned outcomes, careful designing of chain oriented topology is essential. Designing a logical topology for WSNs needs to be considered from different perspectives, namely i) resource oriented considerations, such as energy consumption and time requirement, ii) networking related considerations, such as connectivity, robustness, and reliability, iii) data centric considerations, such as data collection strategies and data aggregation facilities, iv) architecture oriented considerations, such as scalability, task orientation, and light weighting, and v) Network management considerations, such as fault detection and performance management. The drawbacks of chain oriented topologies, such as latency, also need to be considered. In this paper, we propose a variant of chain oriented logical topology. The main aim of this study is to design a logical topology, so that the proposed topology retains the advantages of the chain oriented topologies, and at the same time, overcomes the problems of the chain oriented topology. In designing the proposed logical topology, we considered all the aspects discussed above.

2. EXISTING CHAIN ORIENTED TOPOLOGIES

Chain oriented topologies have been used by researchers in designing various protocols, among which data broadcasting protocols, data collection/gathering protocols and routing protocols are the major instances. Chain topologies are mainly used in these protocols to reduce the total energy consumption, and thus to increase the lifetime of the network. This section discusses different protocols, which use chain oriented topologies.

Lindsey and Raghavendra present several chain oriented data broadcasting and data collection/gathering protocols for sensor networks [24,26]. They investigate broadcast problems in sensor networks and adopt a chain oriented approach for situation awareness systems, where networked sensors track critical events via coordination. They propose a linear-chain scheme for all-to-all broadcasting and data gathering. They also propose a binary-combining scheme for data gathering which divides each communication round into levels in order to balance the energy dissipation in sensor networks. For broadcasting, the linear-chain scheme starts data transmission with a packet at the beginning of a chain. Each node along the chain attaches its own data to this packet. Eventually, information from the entire network reaches the end of the chain. The same procedure runs in the reverse direction to complete all-to-all broadcasting. The linear-chain scheme can also be applied to gather data in sensor networks. To gather data, each node senses and transfers information along the chain to reach one particular node which will send data to a remote base station (BS). Such a scheme is named PEGASIS (Power-Efficient Gathering in Sensor Information Systems) [24].



(a) (b)

Figure 1. PEGASIS protocol chain. (a) chain formation using greedy method, (b) data fusion at the leader node, and transmitting it to BS.

PEGASIS is the first protocol which uses chain oriented topology for periodic data collection from the target field. PEGASIS forms a chain of the sensor nodes and uses this chain as the basis for data aggregation. In PEGASIS, the chain is formed using a greedy approach, starting from the node farthest to the sink. The nearest node to this is added as the next node in the chain. This procedure is continued until all the nodes are included in the chain. A node can be in the chain at only one position. Figure 1(a) shows the chain creation method. In this figure, the node *C0* lies furthest from the BS. Chain construction starts from the node *C0*, which connects to the node *C1* as *C1* is the closest node to *C0*. The node *C1* then connects to its closest node *C2*, and so on. In this fashion a chain *C0-C1-C2-C3-C4-C5* is created. Figure 1(b) shows the data collection strategy adopted by PEGASIS. In the constructed chain, a leader node for each round is selected randomly. The authors argue that randomly selecting a head node is beneficial as nodes are more likely to die at random locations thus providing robust network. All nodes send their data to the leader node, and then, the leader node sends the data to the BS. For example, in Figure 1(b), *C3* is selected as the leader node. The node *C5* passes its data to the leader node *C3* via the node *C4*.

PEGASIS suffers from several problems. First, in this protocol the role of the leader node changes in every round of data collection. This causes extra overhead. Moreover, when a node is selected as the leader, the protocol considers neither the distance of the node from the BS, nor its energy level.

Additionally, the chain in PEGASIS is constructed by a greedy algorithm. Using this chain causes some problems, such as an unexpectedly long transmission time, and non-directional transmission to the BS. These problems adversely affect the energy efficiency of WSNs. All nodes in sensor networks transmit their data in order. Therefore, the delay increases linearly as the number of nodes increases. Thus, PEGASIS is not scalable for large-scale WSNs. PEGASIS also causes redundant transmission of data as a result of having a single leader.

To resolve the delay problem of PEGASIS, a 3-level PEGASIS was proposed. In 3-level PEGASIS, the chain is cut into several chains. Each chain has a leader which gathers data from its neighbours and sends aggregated data to the upper level leader. The delay may decrease with 3-level PEGASIS. However, 3-level PEGASIS raises the problem of wireless interference as it does not consider the relative location of nodes. Another problem is that unexpected long transmission may occur because the leader of a chain sends a packet to the upper leader or the sink node by one hop transmission.

[27] provide an algorithm for constructing the energy efficient chain called the minimum total energy (MTE) chain. These chain construction algorithms use centralized approaches for constructing the chain and elect the leader node for transmitting data back to the sink by taking turns. However, if the remaining energy of each node is not taken into account in the leader

election, the nodes with low remaining energy will easily run out of energy, leaving just a small number of survival nodes to perform the sensing task. From the viewpoint of network lifetime, this is not ideal.

Both PEGASIS and MTE approaches use centralized chain construction which has a number of disadvantages. Firstly, their transmission cost calculation based on distance may not reflect the exact cost in different practical environments due to radio irregularity as indicated in [28]. Secondly, these centralized approaches may not scale well for large network or large number of nodes. Moreover, after some time, nodes far away from the sink easily run out of battery since they consume more energy to transmit to the sink as a leader.

The chain oriented topology proposed in this paper is a multiple-chain oriented topology. In other words, multiple chains are constructed using the deployed sensor nodes in the target field. The chains are constructed in a way to solve the above-mentioned problems of different chain oriented protocols. Furthermore, a network management protocol is associated with the proposed logical topology, so that the network can be managed in such a way as to contend with the resource constraints of WSNs.

3. DESCRIPTION OF THE PROPOSED TOPOLOGY CONSTRUCTION SCHEME

This section describes the proposed multi-chain oriented logical topology in detail. The section is divided into several subsections.

3.1. Basic structure of the proposed logical topology

The features of the basic structure of the proposed logical topology are listed below.

i. All the deployed sensor nodes in the target field take part in the logical topology construction process.

ii. The proposed logical topology consists of multiple chains. Hence, the topology is called multi-chain oriented topology. These chains are called lower-level chains.

iii. All the chains of the proposed topology are simple chains, rather than complex chains. A simple chain is defined as a chain where each member node of the chain has, at the most, two neighbouring nodes. On the other hand, a member node may have more than two neighbouring nodes in a complex chain. Figure 2 shows an example of both simple chain and complex chain. Note that, in Figure 2 the member node *C2* has four neighbouring nodes - *C1*, *C3*, *C4*, and *C5*.

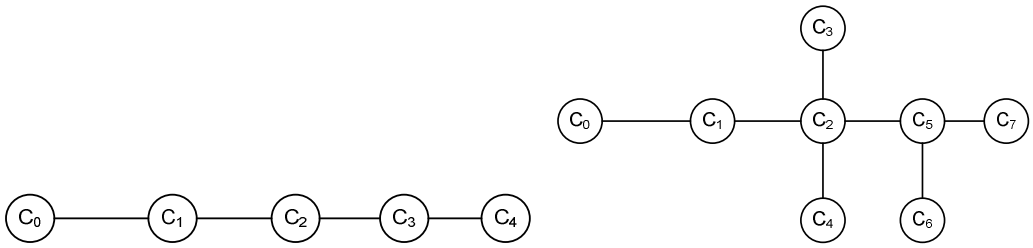
iv. In a lower-level chain, the distances between any two successive nodes are called links. Thus, a chain that consists of n number of sensor nodes has (n - 1) links. The sum of these (n - 1) links is the length of that chain.

v. The length of each chain of the proposed topology is similar. As it is assumed that the sensor nodes are deployed randomly in the target field, constructing multiple chains having exactly the same length may not always be possible. However, the proposed logical topology creates chains of similar lengths to avoid uneven energy consumptions by chains of dissimilar lengths.

vi. For each chain, a member node of the chain is elected as the leader of the chain. These leaders are called lower-level leaders.

vii. The lower-level leaders construct a higher-level chain. Similarly, a member node of the higher-level chain is elected as the leader of the chain. This leader is called the higher-level leader.

A sample architecture model of the proposed logical topology is depicted in Figure 3. This figure shows the logical topology using two hierarchical layers.



(a) Simple chain (b) Complex chain

Figure 2. Types of chains - simple chain and complex chain.

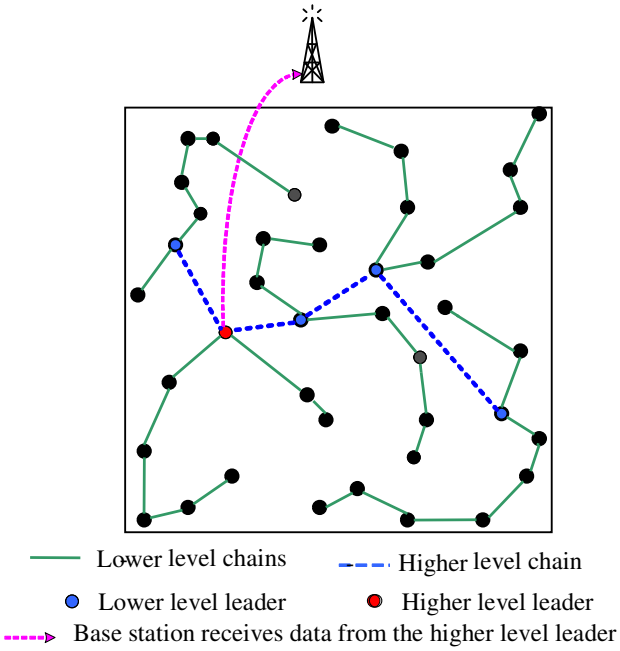


Figure 3. A sample model of the proposed topology

3.2. Different phases of the proposed topology

The proposed logical topology can be described using three phases, namely i) topology formation phase, ii) steady state phase, and iii) topology update phase. Figure 4 demonstrates these phases with respect to a timeline. Additionally, Figure 5 demonstrates the transitions among different phases.

At the initial stage of the sensor deployment in the target field, the topology formation phase starts. This phase takes place only once. The steady state phase and the topology update phase then follow.

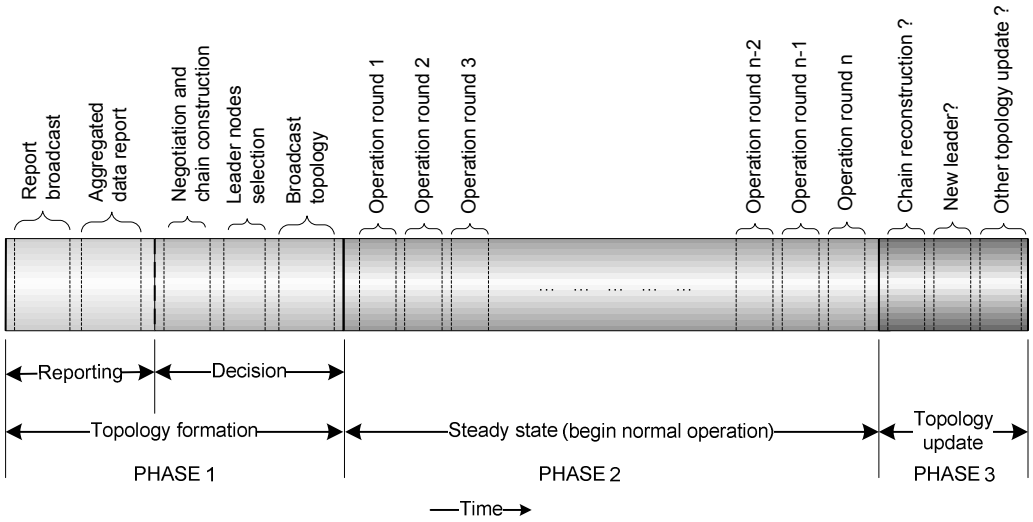


Figure 4. Timeline of the proposed topology

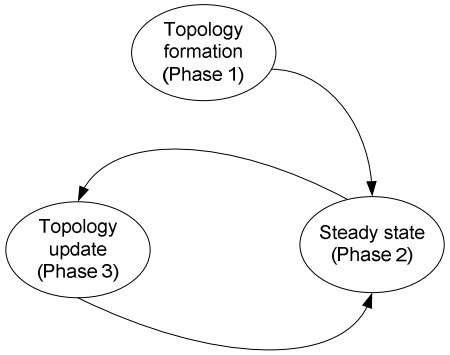


Figure 5. Transitions of different phases of the proposed topology

At the beginning of the topology formation phase no sensor nodes recognises any other sensor node in the target field. Each of the deployed sensor nodes then reports its individual characteristics to all of its neighbouring sensor nodes using broadcasting. A sensor node, receiving broadcasted messages by its neighbouring nodes, calculates the distances between itself and the neighbouring nodes. Additionally, each sensor node aggregates the reports it collects from its neighbouring nodes. After reporting, all the sensor nodes negotiate with their neighbours and construct several chains. When the chain constructions finish, lower-level leaders are elected for each chain. Each lower-level chain then broadcasts the topology, describing the member nodes, successor-predecessor lists, and time division multiple access (TDMA) allocations. At this point, the topology formation phase ends, and the deployed sensors are ready for their normal operation.

At the end of the topology formation phase, the steady state phase begins. In this phase, the sensor nodes start their normal operation. Without the loss of generality, it can be assumed that the sensors are deployed in the target field to collect some data. The steady state consists of several rounds. A round begins whenever the sensor nodes start their sensing. A round finishes when the higher-level leader collects all sensed data via the lower-level leaders, and then sends the data to the BS.

After the end of a fixed number of rounds in the steady state, the topology update phase takes place. The tasks of this phase are to maintain the topology, such as selection of new lower-level leaders, construction of a higher-level chain, selection of a higher-level leader, and reconstruction of chains, if necessary.

3.3. Chain construction algorithm for the proposed topology

The proposed chain construction algorithm consists of three steps, namely i) generating the shortest path chain, ii) link exchange, and iii) pruning. Step one generates an initial single chain which is derived using the Kruskal minimum spanning tree algorithm. This initial chain may not be optimized, because of the existence of some cross links. At steps two and three, these cross links are removed, the chain is reconstructed and pruned to multiple chains. The chain construction algorithm is depicted in Figure 6, and detailed descriptions of each step are provided below.

Step 1

$A = \{$ 1, 2, 3, …, N $\}$ // set of sensor nodes

$SH = \phi$ // set of links $L(i, j)$

Assign C[i][j] = Cij

for $(\forall i \in A)$ node[i].peer_leaf = i

repeat until A contains two elements // there would be two leaf nodes in the initial chain

Find i and j that minimize C[i][j] such that $((i, j \in A) \,\&\, (i \neq j) \,\&\,$ (node[i].peer_leaf $\neq j$))

construct_chain(i, j)

Procedure construct_chain(i, j)

place (i, j) in SH

node[node[i].peer_leaf].peer_leaf= node[j].peer_leaf

node[node[j].peer_leaf].peer_leaf= node[i].peer_leaf

if (node[i].peer_leaf ≠ i) remove i from A

if (node[j].peer_leaf ≠ i) remove i from A

// SH contains all the links that constitute the initial chain

Step 2

do

Start tracing the chain starting from any leaf node.

Find crossed links (w, x) and (y, z)

if ($C(x, y) + C(w, z) \leq C(w, x)$+C($y$, z))

$SH = SH - (w, x), (y, z)$

$SH = SH + (x, y), (w, z)$

until all nodes are traced

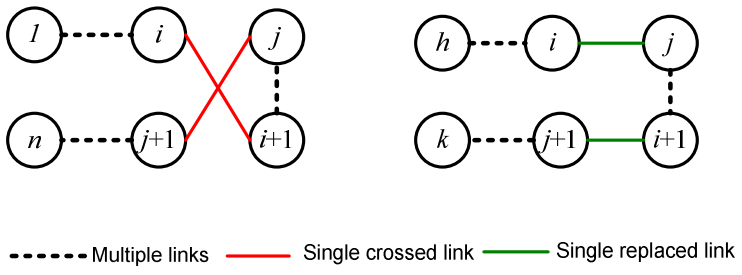
Step 3

Divide the chain constructed after step 2 into multiple chains with similar number of node in each chain

---

Figure 6. Chain construction algorithm

Step 1. Configuring the initial chain. This step generates an initial chain, which is derived from the Kruskal minimum spanning tree algorithm by giving an additional constraint of a maximum degree of 2. This algorithm selects a link, one by one, through a specified routine. Since links are selected as long as a loop does not occur, several complex chains (see figure 2(b)) can be generated during generating the chain. When some links are formed, the next link is the shortest link among links that connect those nodes whose degree is under 2. However, the two end nodes are not included in the same sub-chain.



(a) (b)

Figure 7. Link exchange. Crossed links(a) are replaced by new links(b).

Step 2. Link Exchange. For large number of nodes, there is a high possibility that the initial chain generated after step 1, includes some cross links (see Figure 7). In this step, cross links are removed, and the chain is pruned to multiple chains. The cross link removal process takes place when there are available links, whose lengths are shorter than that of the cross links. This process is called link exchange.

In Figure 7, the nodes are numbered from 1 to n. In this figure, dotted lines represent sub-chains that are consisted of several links. The solid lines in this figure represent a single link. When the process of link exchange occurs, the order sub-chain from i+1 to j is reversed. To exchange two links of the chain as from (i, i +1) and (j, j+1) to (i, j) and (i +1, j +1), the following condition should be satisfied:

$C(i,i+1)+C(j,j+1) \geq C(i,j)+C(i+1,j+1)$, where C(i, j) is the length of the link (i, j).

Step 3. Pruning. At the end of the link exchange, an optimal chain is generated. To create multiple chains from this optimal chain, each node of this chain is traced, starting from the farthermost end of the chain from the BS. The tracing process takes place from one node to its neighbouring node until the number of nodes traced is equal to C_N. Here, C_N is the optimal number of node in a chain. At this point all nodes which are which have already been traced are pruned from the initial chain. This pruning process continues until all the nodes of the initial chain are traced.

3.4. Selection of leader nodes

Suppose any node in a chain can be elected as a leader, and the leader is responsible to send the aggregated data to the BS. The maximum number of operational rounds that can be achieved before any node exhausts its power is analysed first. Without loss of generality, it can be assumed that nodes in the chain are numbered sequentially as 1, 2, …, n. Let e_i be the energy consumed by the node i in transmitting a data message to the BS. Let $\rho_{i,j} = kE_{elec} + k\varepsilon_{amp}((d(i,j))^{\alpha}$ be the energy consumed by the node i, and $e_r = kE_{elec}$ be the energy consumed by the node j when the node i transmits a k-bit message to the node j. When some node i is selected to be the leader, every node numbered $j < i$ (if any) expends energy $\rho_{j,j+1}$ in sending data to the node j+1, at which energy e_r is consumed to receive the data. Likewise, every node numbered $k>i$ (if any) expends $\rho_{k,k-1}$ to send data to the node k-1, where energy e_r is expended in receiving the data. The leader transmits the collected data to the BS, consuming energy e_i. Suppose that, every node i is scheduled to be the leader x_i times. Table 1 shows the energy expense of every sensor node in this case.

Table 1. Energy consumption by different nodes while acting as a leader

Node ID	In sending message to the BS	In sending message to neighbours	In receiving neighbour's message
1	$e_1 x_1$	$\rho_{1,2}\sum_{j=2}^{n} x_j$	$e_r x_1$
$i \in \{2,3,...,n-1\}$	$e_i x_i$	$\rho_{i,i-1}\sum_{j=1}^{i-1} x_j + \rho_{i,i+1}\sum_{j=i+1}^{n} x_j$	$e_r\left(\sum_{j=1}^{i-1} x_j + 2x_i + \sum_{j=i+1}^{n} x_j\right)$
N	$e_n x_n$	$\rho_{n,n-1}\sum_{j=1}^{n-1} x_j$	$e_r x_n$
x_i : the number of times node i is selected to be the leader e_i : the amount of energy consumed in transmitting message from node i to BS. $\rho_{i,j}$: the energy consumed by I in transmitting a message to j e_r : the energy consumed by any node in receiving a message			

Optimal leader scheduling problem is to find a positive integer values of x_i's as to maximize $\sum_i x_i$ subject to the following constraints:

$$E_1 \geq (e_1 + e_r)x_1 + \rho_{1,2}x_2 + \rho_{1,2}x_3 + \ldots + \rho_{1,2}x_n$$
$$\vdots$$
$$E_i \geq (\rho_{i,i-1} + e_r)x_1 + \ldots + (\rho_{i,i-1} + e_r)x_{i-1} + (e_i + 2e_r)x_i + (\rho_{i,i+1} + e_r)x_{i+1} + \ldots + (\rho_{i,i+1} + e_r)x_n$$
$$\vdots$$
$$E_n \geq \rho_{n,n-1}x_1 + \rho_{n,n-1}x_2 + \ldots + (e_n + e_r)x_n$$

where E_i denotes the amount of energy that node i initially has.

These constraints can be formulated as

$$A\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} \leq \begin{pmatrix} E_1 \\ E_2 \\ E_3 \\ \vdots \\ E_n \end{pmatrix}, \text{ where } A = \begin{pmatrix} e_1 + e_r & \rho_{1,2} & \ldots & \rho_{1,2} \\ \rho_{2,1} + e_r & e_2 + 2e_r & \ldots & \rho_{2,3} + e_r \\ \rho_{3,2} + e_r & \rho_{3,2} + e_r & \ldots & \rho_{3,4} + e_r \\ \vdots & \vdots & \ldots & \vdots \\ \rho_{n,n-1} & \rho_{n,n-1} & \ldots & e_n + e_r \end{pmatrix}$$

Thus, the problem becomes a linear programming problem. Round robin leader scheduling equalizes the values of x_i's, which is generally far from optimal. The authors of PEGASIS also proposed an improvement on round robin scheduling [29]. This approach sets up a threshold of distance, and nodes are not allowed to be leaders if the distance to their neighbours along the chain is beyond the threshold.

From the above discussion, the ability to achieve optimal results in leader selection is a computationally rigorous task. Thus, instead of finding an optimal solution, the proposed topology uses a simple rule called Maximum Residual Energy First (MREF) for leader selection. This simple algorithm gives near optimal results for a lower number of nodes. As in

the proposed topology, there are only a few lower-level leaders making this algorithm perfectly suits for selecting a higher-level leader. As the name suggests, MREF selects the node that has the maximum residual energy to be the leader for network operations. Residual energy information can be piggybacked with data messages as a part of the aggregated data. If every lower-level leader attaches its own energy level to data message and lets the BS find the maximum value, it will incur an additional $O(n)$ overhead on every message. A better approach is to let every lower-level leader compares its energy level with the energy level attached to the incoming data message (if any) and send only the large one. The message overhead in this process is only $O(1)$.

For the lower-level leaders, the same selection procedures can be followed. However, since the communications of the lower-level leaders are not as energy intensive as for higher-level leader, it is proposed not to change lower-level leaders as frequently as higher-level leader. The benefits of using a slightly longer duration for selecting lower-level leaders include: i) less communication overhead, ii) reduced required time for leader selection at every round, and iii) maximum utilization of the higher-level chain.

3.5. Design issues of the proposed logical topology

Design issues that need to be discussed in relation to the proposed logical topology include the number of chains in the system, the number of nodes in a chain, and the time when the leaders should be changed or the chains should be reconstructed/updated. Other issues regarding network management include the arrival of a new node, or dead / aberrant nodes. These issues are discussed below.

i. Total number of chains in the system

The system can determine, *a priori*, the optimal number of chains (lower-level) for a particular system. This depends on several parameters, such as the positions of the sensor nodes, and the relative costs of computation versus communication. The proposed topology was simulated for a data collection application using a network where 100 sensor nodes were randomly deployed. The value of the radio parameters of the transmitter and the receiver that were used in the simulation are $E_{tx\text{-}elec}=E_{Rx\text{-}elec}=E_{elec}=50$ nJ/bit. The transmit amplifier was assumed to be 100 pJ/bit/m^2. A computation cost of 5 nJ/bit/message to fuse 2000-bit messages was further assumed. In the experiment, the number of chains in the system was varied gradually in order to observe its impact on energy consumption, and delay. Figure 8 shows how the energy dissipation in the system varies with the number of chains in the system. Note that, a zero chain means that no lower-level chain, and thus no higher-level chain is constructed. In this situation, each sensor node directly transmits its sensed data to the BS. Also note that, 1 chain means there would be no higher-level chain, and 100 chains means there is actually no lower-level chain (because of only one member in each chain), only a single higher-level chain. Therefore, both 1 chain and 100 chains refer to the same system as PEGASIS. Figure 8 suggests that energy consumption would be lower if the number of chains can be kept below 10 or above 80. However, a large number of chains would cause more overhead . Thus, for the proposed topology, the number of chain is maintained at 6%-8% of the sensor nodes. Therefore, for a target field of 200 sensor nodes deployed, 12 to 16 chains would be constructed.

ii. Optimal number of nodes in a chain

The optimal number of sensor nodes in a chain, denoted as C_N, is the number of nodes that should be included in each chain during the chain construction phase. It can be argued that, if the number of nodes in a chain is fewer than C_N, both the required time and energy dissipation increase in the network. On the other hand, if the number of nodes is more than C_N, energy dissipation may decrease slightly, however the time requirement increases. Additionally, for the

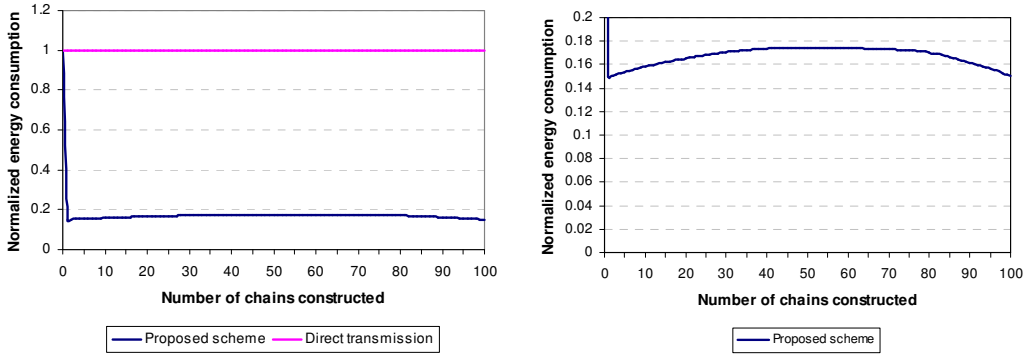


Figure 8. Normalized total energy dissipated vs the percent of leader nodess.

sake of even energy dissipation distribution, the lengths of the chains should be similar. Thus, in the proposed scheme, a similar number of sensor nodes are included for each chain. Since it is assumed that sensors are deployed randomly in the target field, creating chains of exactly the same number of sensor nodes may not be possible. However, the proposed scheme maintains a similar number of nodes in each chain. Thus for a target field of 100 nodes, the number of sensor nodes in each chain C_N would be = 12 to 17.

iii. Chain Reconstruction

It is important to reconstruct the chains whenever a significant number of sensor nodes in a chain expire. Otherwise, one chain may contain a higher number of sensor nodes, while others may contain a lower number of sensors. This affects the performance of the topology due to uneven energy dissipation by the chains. It is vital to maintain uniformity in the number of sensor nodes in all chains as only one sensor node (i.e. the higher-level chain leader) is responsible for sending the aggregated data to the BS, and it has to wait for aggregated data from different lower-level leaders. Thus, the uniformity of number of sensors in chains affects network lifetime. If a chain consists of a lower number of sensors, the probability of a sensor in that chain being selected as a local leader will be higher. Thus, a chain of short length is likely to lose sensors more often. It is obvious that if chains are reconstructed frequently, such as whenever only 4%-5% sensors of the chain die, it causes extra overhead. On the other hand, if the chain is reconstructed whenever 40%-50% sensors of the chain die, the uniformity among the chains is destroyed. To answer the question of when a chain should be reconstructed, simulation experiments were performed. To find the optimal value, chains were reconstructed varying the percentage of sensors' death in the chains, and its effects on total energy spent, lifetime of the network, and time required to complete 100 rounds were observed. Simulation results show that although the energy consumption increases when chains are reconstructed less frequently, the amount of energy difference is not extreme. Additionally, the lifetime of 95% of the deployed sensor nodes remains almost steady regardless of the percentage of expired sensors when a chain is reconstructed, with a small peak when approximately 20% of the deployed sensors have expired. Simulation results also show that time requirements decrease when chains are reconstructed less frequently. Time requirement falls sharply when between 4%-20% sensors die and then decreases slowly afterwards. Thus, after careful consideration, we conclude that it is best to reconstruct chains when approximately 20% of the sensors within a chain expire.

To track how many sensor nodes are expired in a chain, the following method can be used. When data are fused in every sensor of a chain, each sensor adds its tag to the data packet. For example, let node n_1 sends data to n_2, and n_2 fuses n_1's data and send it to n_3. However, if n_2 is

dead, n_1 sends data directly to n_3, and thus the node n_3 knows that n_2 is dead. In this way every lower-level leader can determine how many of its members are dead. In a similar fashion, when the higher-level leader collects data from all lower-level leaders, it can determine how many sensors in the network are dead. Subsequently, the higher-level leader sends instructions to all sensor nodes.

iv. Changing lower-level leaders

The lower-level leaders should be changed periodically to distribute the energy load. PEGASIS suggests changing the leader node in each round. However, for the proposed topology, if the lower-level leaders are changed at every round, it causes extra energy expenditure for negotiations to select leaders, as well as causing delay. In addition, the higher-level chain can be fully utilized if the lower-level leaders are changed after a number of rounds. Conversely, if the lower-level leaders are not swapped with other member nodes for a long time, they will quickly be drained of energy due to excessively long transmissions. Therefore, in the proposed logical topology, lower-level leaders are changed after R rounds, where the value R depends on the following criteria: i) total energy dissipation in the network, ii) maximum number of round before the first sensor node dies, and iii) the delay introduced in the network for different number of rounds.

We perform extensive simulation experiments to determine when the lower-level leaders should be changed. Simulation results show that there is no correlation between total energy consumption and R. In contrast, simulation results show that as the value of R increases, the network lifetime decreases. This is because, when the same sensor nodes are working as leaders for long periods, they deplete energy quickly compared to other sensor nodes. Additionally, the time delay decreases as the value of R increases. From the experimental result, we propos that the lower-level leaders are changed after every $C_N/2$ rounds.

v. Inserting additional nodes into the network

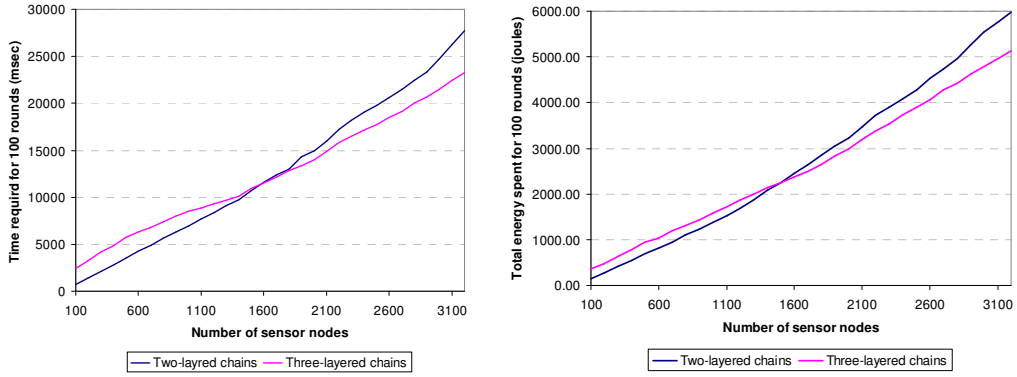
Additional nodes may be inserted into the network at any time. Before a node is inserted, the BS records and stores its unique ID and will insert the node into a nearby chain with the least number of nodes. This helps to minimize the chance of a chain monopolising a certain bandwidth if it contains a greater number of nodes than other chains which are communicating. The node will then organize itself within its chain.

vi. Identifying and isolating aberrant nodes

Sensor nodes that do not function as specified must be identified and isolated in order to continue the desired operation of the sensor network. An aberrant node may be the result of an attack or may act maliciously due to unexpected network behaviour. According to [30], an aberrant node is one that is not functioning as specified, and may cease to function as expected for the following reasons:

i. it has exhausted its power source or is damaged by an attacker,
ii. it is dependent upon an intermediate node and is being deliberately blocked because the intermediate node has been compromised,
iii. an intermediate node has been compromised and is corrupting the communication by modifying data before forwarding it, or,
iv. a node has been compromised and communicates fictitious information to the BS.

Therefore, the WSN should be maintained by identifying an aberrant node quickly and isolating it from the sensor network. The protocol named SecCOSEN [25] can be used for authentication purposes. This protocol perfectly suits the logical topology, as it was designed for a multi-chain



(a) (b)

Figure 9. (a) Timing and (b) energy consumption differences between 2-layer and 3-layer chains

oriented logical topology. Using this protocol, a node can authenticate the node from which it receives data/messages. If a node is not able to authenticate another node in the chain, the former node reports the incident to the chain leader. In addition, a node also maintains a timer for identifying any dead node with the help of timeouts and reports the incident to the leader node.

vii. Number of Layers

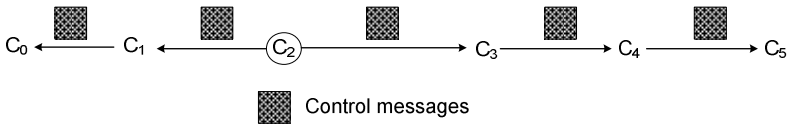
Although we describe the proposed multi-chain oriented logical topology using a two-layer model, the number of layers can be extended based on the number of sensor nodes in the target field. Figure 9(a) shows the simulation results and comparison between 2-layered and 3-layered chains with respect to the time required for 100 rounds. The figure demonstrates that 2-layered chains take less time to reach 100 rounds than 3-layered chains until the number of sensors is greater than 1600 when the reverse is true. The same situation arises for total energy consumption depicted in Figure 9(b). The 2-layered architecture saves more energy than the 3-layered architecture up until the number of sensor nodes exceeds 1500 when the reverse is true. Thus, it is concluded that, if the number of sensor nodes in the target field is less than 1500, two-layered architecture is used, while if the number of sensor nodes is equal to or greater than 1500, three-layered architecture is more suitable.

3.6. Communication abstraction of the proposed topology

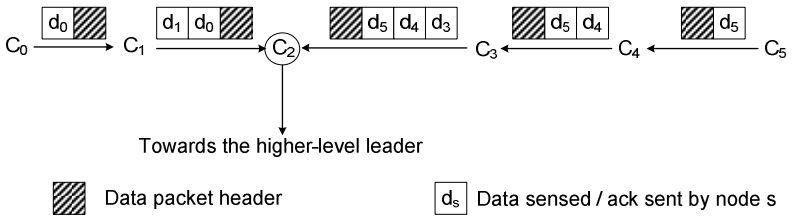
This section describes the communication abstraction for the proposed multi-chain oriented logical topology. Communication is fundamental to any logical topology of WSNs. The power of a WSN comes not from the capabilities of the individual devices, but from the collective capabilities achievable through wireless communication.

Addressing the intricacies of wireless communication can be a difficult, error-prone task. This is especially true of WSN applications, where the number of participating devices can be large, the communication patterns can be complex, and the network links are ad-hoc and unreliable. However, the proposed topology restricts the communications of a sensor node to only its successive nodes in its chain. Thus, the burdens of multicasting and broadcasting are removed from the sensor nodes.

The communication abstraction of the proposed topology can be divided into two parts, namely i) communications within a chain, and ii) communications between chains.



(a) Control message dissemination



(b) Sending data towards the lower-level leader

Figure 10. Communications in a chain.

i. Communications within a chain

Within a chain, sensor nodes communicate with each other to disseminate control information and sensed data. Communications among the sensor nodes are restricted to only the successive sensor nodes. Figure 10 shows the communication pattern inside a chain. In this figure, six sensor nodes (C_0 to C_5) construct a chain. C_2 is the lower-level leader of the chain. The lower-level leaders disseminate information and control messages to all the member nodes of their chains. These information and control messages are propagated hop-by-hop from one sensor node to its successive neighbouring node. For example, Figure 10(a) shows that the leader node C_2 sends the control information to the nodes C_1 and C_3. After copying the control message, the node C_1 sends the control message to the node C_0 and C_3 sends the message to C_4, which then sends it to C_5. As the nodes C_0 and C_5 are the end nodes of the chain, they refrain from sending the control message any further.

For sending the sensed data, each sensor node sends data to its successive node towards the leader of the chain. For example, in Figure 10(b), the node C_0 sends its sensed data to the node C_1, while the node C_1 merges its own data with C_0's data, and sends them to the leader node C_2. Similarly, the node C_5 sends its data to the node C_4, C_4 then sends C_5 's data and its own data to the node C_3. The node C_3 accumulates this data with its own data, and sends them all to the leader node C_2.

ii. Communication between chains

Different lower-level chains communicate with each other using the higher-level chain. The lower level leaders accumulate data sent by the member nodes of the chains, and transfer them to the higher level leader. The higher-level leader then sends the data to the BS.

If the BS, or the higher-level leader wants to send some information, or control messages to the chain members, the communication path remains the same, except the direction is opposite. In this case, the communication pattern is similar to hub-and-spoke topology.

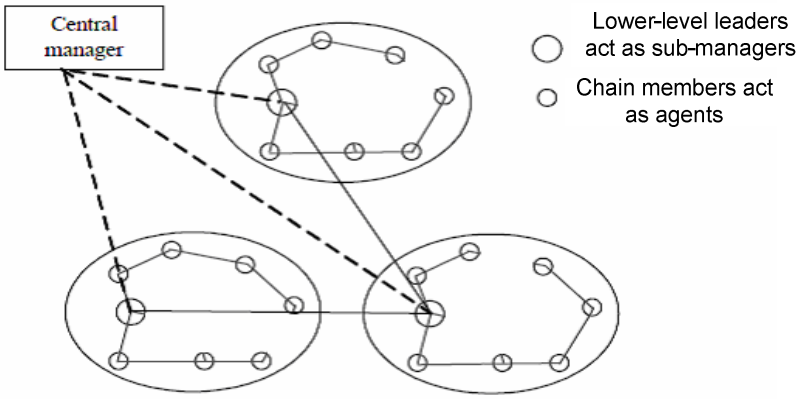


Figure 11. Different entities of the network management scheme for the proposed topology

4. NETWORK MANAGEMENT OF THE PROPOSED TOPOLOGY

This section presents the network management architecture and processes for the proposed logical topology. Network management is the process of managing, monitoring, and controlling the behaviour of a network. The management approach of WSNs differs from the traditional wired networks and mobile ad-hoc wireless networks due to the unique characteristics and restrictions of WSNs.

For the proposed multi-chain oriented topology, a three-layer hierarchical management architecture is proposed. Figure 11 represents the relationship between the different entities of the management architecture, namely the manager, the sub-manager and the agent nodes. The manager is in the highest level of the hierarchy, and is placed at the BS. The lower-level chain leaders of the proposed topology work as sub-managers, and the chain member nodes work as agent nodes. The sub-managers are used to distribute management functions, and to collect and collaborate management data. The manager has the global knowledge of the network states and gathers the global knowledge from the underlying network layers and sub-managers.

The proposed logical topology arranges the nodes into groups of chains and identifies a chain leader for each chain. This allows a subset of nodes to communicate with the sink nodes, conserving energy in the nodes that no longer need to send data to the sinks. Often sink nodes are farther away from many nodes in the network. Chaining procedure abandons these long paths required for communication for smaller hops since nodes will only be communicating with neighbour nodes (except for the chain leaders). Besides energy and bandwidth conservation, there are other advantages of clustering nodes in a WSN. One advantage is that it allows for spatial reuse of resources. If two nodes exist in different non-neighbouring clusters, it may be possible for the two nodes to share the same frequency or time slot. It is also beneficial in the presence of mobility. When using clustering and a node moves, it is often only necessary to update the information in the nodes sharing a cluster with the mobile node; all nodes in the network will not have to be updated. Clustering into chains can also facilitate network management and routing since many implementations require only the chain leader to participate in these functions. In this management architecture, the chain leaders (often called sub-manager) report the data to the manager on behalf of the entire cluster.

Three major aspects of the proposed network management, namely fault detection, performance management, and security management are discussed below.

4.1. Fault detection

Fault detection is the process by which the network manager identifies a node which is malfunctioning or almost dead and unable to sense or transmit data. If a normal sensor node dies, it does not create much of a problem except decreasing reliability. However, if a chain leader dies, the data of that chain are lost, and in the worst case, such a failure introduces network partition in the system.

In traditional IP networks, the usual way to determine whether a node is working properly is to receive periodic *keepAlive* messages from that node. However, for sensor network such message exchange is very costly. Therefore, fault detection operation in WSNs should be lightweight, and performed using passive information as much as possible.

The fault of normal sensors is detected by the sub-manager (i.e., by the lower-level leaders). If the sensors are supposed to send data periodically, then by analyzing the packets, the lower-level leader can identify the sensor node that is not responding. The lower-level leaders can also miss packets from member nodes caused by collisions. Inside the chain, each sensor maintains the state of its neighbours. If a sensor does not hear from any of its neighbour for a certain period of time, the node informs the lower-level leader about that particular sensor. The lower-level leader and the neighbours maintain a timer T for each of the neighbour sensors. If the lower-level leader or the neighbours hear a transmission from that sensor, then they reset the timer. If the timer of the lower-level leader expires, then it waits before declaring the alarm. If the timer of the neighbour expires, it piggybacks that information in the next data packet. If the lower-level leader receives packets from any of the neighbours of that node without any negative result, the leader waits for another random time. If there is no positive response before the timer expires, or random delay is extended three times, then the leader node generates an alarm, and decides that the node is dead. The leader then informs the manager about the dead node. For event driven sensor networks, the sensor sends a periodic *keepAlive* message to the sink in the absence of an event.

Lower-level leaders use timer T and reset it when fault detection of lower-level leaders is more important than that of a chain member node. In cases of periodic traffic, the central manager analyses the packets received by the sink. As the central manager knows the topology of the network, it knows the path of each chain leader to the BS. It maintains two timers (T_1 and T_2) for each chain leader and for gateway nodes. When the sink receives a packet from that node or through that node, the central manager restarts the timer. If the timer expires, then the central manager suspects that node is dead. As the fault should be detected immediately, the value of T_1 should not be very high. When the timer expires, the chain leader sends a query packet to the node and waits for another time T_2. If no response is received, it decides that the node is dead.

In event driven sensor networks, in the absence of events, the chain leaders or gateway send periodic message and the chain leader uses the same timer mechanism to detect faults.

4.2. Performance management

The performance management of WSNs monitors the performance of the network and keeps resource consumption as low as possible, especially the use of energy. One of the major performance issues of the WSN is event reliability, which is defined as the number of unique data packets received by the sink node. For optimum performance, the management system sets the data generation rate of the sensors and may also keep some nodes in the sleep state and others in the normal live state.

Performance management consists of monitoring network devices and links in order to determine utilization. Utilization may vary depending on the device and link; it may include

processing load, network card utilization, packet-forwarding rate, error rate, or packets queued. Monitoring utilization helps to ensure there is available capacity. Monitoring the network performance assists in identifying current and future bottlenecks and aids in capacity planning. Tracking the utilization of network resources by each user is the goal of accounting management. The primary function of this information is to bill users for their use of the network and its resources. This information can be used to establish metrics and quotas. The usage information also helps the network manager to allocate network resources properly. It is also helpful to see typical user behaviour as then atypical behaviour can be identified and addressed. Atypical behaviour may indicate a security breach or intrusion or may be an indication of a future device problem.

4.3. Security management

Due to the large number of sensor nodes and the broadcast nature of wireless communication, it is usually desirable for BS to broadcast commands and data to sensor nodes. The authenticity of such commands and data is critical for the normal operation of sensor networks. If convinced to accept forged or modified commands or data, sensor nodes may perform unnecessary or incorrect operations and cannot fulfil the intended purposes of the network. Thus, in hostile environments (e.g., battlefield, antiterrorists operations), it is necessary to enable sensor nodes to authenticate broadcast messages received from BSs.

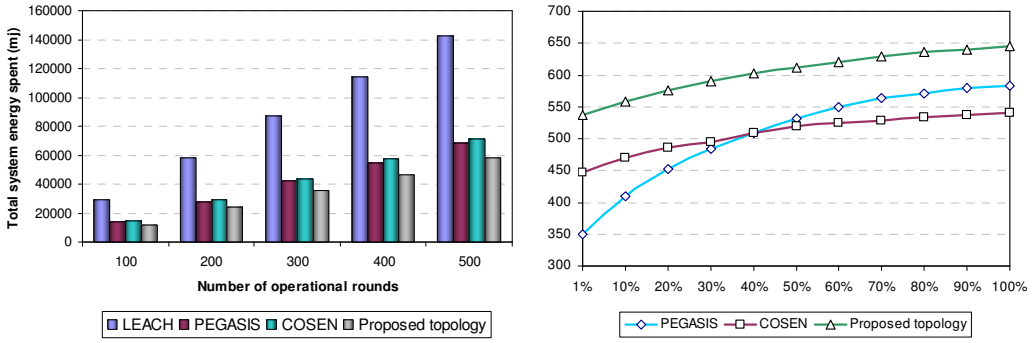
A protocol that can be adopted in the proposed logical topology is SecCOSEN, which has been proposed for authentication, and establishing secret keys in WSNs for multi-chain oriented logical topology. SecCOSEN uses partial key pre-distribution and symmetric cryptography techniques. While one version of the SecCOSEN protocol uses shared partial keys in a sensor chain, the other version uses private partial keys. Both versions of SecCOSEN show high resilience to different security attacks. The protocol outperforms other random key pre-distribution protocols as it requires less space, has lower communication overheads, and offers very high session key candidates.

5. PERFORMANCE EVALUATION OF THE PROPOSED TOPOLOGY

Several simulation experiments were carried out to evaluate the performance of the logical topology. The proposed logical topology was used for data collection, and its performance was measured against existing data collection protocols, namely LEACH [31], PEGASIS [29], and COSEN [32].

The simulation program was written in object oriented programming language C++. One hundred sensor nodes were assumed to be randomly distributed in the target field of 100m_100m, and the BS was located at (25, 150). Cartesian coordinates were used to locate the sensor nodes. It was further assumed that each sensor starts with one Joule of initial energy.

In practice it is difficult to model energy expenditure in radio wave propagation. Therefore, in order to measure the energy expenditure in the network, the same simplified radio model used in LEACH and PEGASIS was used. The value of the radio parameters of transmitter and receiver electronic that were used in the simulation are $E_{tx\text{-}elec}$=$E_{Rx\text{-}elec}$=E_{elec}=50 nJ/bit. The value of transmit amplifier ε_{amp} was assumed 100 pJ/bit/m^2. It was further assumed that, a computation cost of 5 nJ/bit/message to fuse 2000-bit messages. The bandwidth of the channel was set to 1 Mb/s. Thus the total transmission cost for a k-bit message is given by the equation: $E_{tx}(k, d) = E_{elec}k + \varepsilon_{amp} kd^2$. Here d is the distance between sender and receiver measured in meters. In the case of receiving a message, the energy consumption equation is given by the equation: $E_{rx}(k)= E_{elec}k$.



(a) (b)

Figure 12. (a) Energy consumption and (b) network lifetime comparisons

Multiple runs of the simulation for each protocol were performed and the average value was taken. The metrics that were considered to measure the performance of each protocol were i) overall energy expenditure in the network ii) lifetime of the network, iii) time to complete a fixed number of operational rounds.

The first experiment measured the total energy consumption by the system varying the number of operational rounds. Figure 12(a) shows the results. PEGASIS was found to be more energy conservative than LEACH and COSEN, however, the proposed topology outperformed PEGASIS by saving more than 10% of total energy for 500 data collection rounds. This is because of the optimal chain creation by the proposed algorithm, and efficient leader selection processes.

While the proposed topology was the most efficient in total energy consumed, the main success of the proposed topology is the even distribution of energy consumption. Uneven energy consumption by the sensor nodes adversely affects the system lifetime. Figure 12(b) demonstrates the lifetime patterns of PEGASIS, COSEN and the proposed topology. The figure shows that the death of the first node in PEGASIS occurs at an early stage compared to COSEN and the proposed topology. For PEGASIS, 10% of the nodes die at around 400 operational rounds, whereas for the proposed topology, 10% of the nodes die at around 550 rounds.

The definitive improvement of the proposed topology over PEGASIS is the latency in data collection. In the simulation, the required amount of time to complete different numbers of operational rounds for PEGASIS and the proposed topology was calculated. The pattern of the time requirement graph suggests that PEGASIS is not suitable for large-scale WSNs due to latency. For 100 operational rounds, the proposed topology requires approximately one-fifth of the time required by PEGASIS.

6. CONCLUSION

This paper presents a multi-chain oriented logical topology for WSNs. The design of the topology is governed by various factors including resource constraints such as energy, time, and computational complexity, as well as networking and architectural factors, and network management issues. We provide a detailed description of the construction of the proposed topology. Moreover, we propose a three-layer hierarchical management architecture for the multi-chain oriented topology. The network management scheme works in line with the proposed topology for managing different issues such as fault detection, performance management, and security management.

In designing the proposed multi-chain oriented topology, it is important to note that reducing the energy consumption will not always result in a longer system lifetime. Instead, balancing resources among sensors, and saving energy for those more resource-constrained sensors are very helpful in lengthening the overall system lifetime. Based on this principle, we construct the proposed topology and select the leader nodes.

Simulation results showed excellent results in favour of the proposed logical topology. The proposed logical topology outperformed LEACH, PEGASIS and COSEN not only in total system energy consumption, but also in system lifetime. The key reason behind this is the more even distribution of energy consumption. The proposed topology also solves the high delay problem of PEGASIS.

REFERENCES

[1] Nishiyama, H.; Ngo, T.; Ansari, N.; Kato, N. OnMinimizing the Impact ofMobility on Topology Control in Mobile Ad Hoc Networks. Wireless Communications, IEEE Transactions on 2012, 11, 1158 –1166.

[2] Chiwewe, T.; Hancke, G. A Distributed Topology Control Technique for Low Interference and Energy Efficiency in Wireless Sensor Networks. Industrial Informatics, IEEE Transactions on 2012, 8, 11 –19.

[3] Avidor, 760 D.; Mukherjee, S.; Onat, F. Transmit Power Distribution of Wireless Ad Hoc Networks with Topology Control. INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE, 2007, pp. 46 –52.

[4] Liu, S.Y.; Huang, C.C.; Huang, J.L.; Hu, C.L. Distributed and localized maximum-lifetime data aggregation forest construction in wireless sensor networks. Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on, 2012, pp. 655 –660.

[5] Hua, C.; Yum, T.S. Optimal Routing and Data Aggregation for Maximizing Lifetime of Wireless Sensor Networks. Networking, IEEE/ACM Transactions on 2008, 16, 892 –903.

[6] Yoo, H.; Shim, M.; Kim, D. Dynamic Duty-Cycle Scheduling Schemes for Energy-Harvesting Wireless Sensor Networks. Communications Letters, IEEE 2012, 16, 202 –204.

[7] Zairi, S.; Zouari, B.; Niel, E.; Dumitrescu, E. Nodes self-scheduling approach for maximising wireless sensor network lifetime based on remaining energy. Wireless Sensor Systems, IET 2012, 2, 52 –62.

[8] Hamida, E.; D'Errico, R.; Denis, B. Topology Dynamics and Network Architecture Performance in Wireless Body Sensor Networks. New Technologies, Mobility and Security (NTMS), 2011 4th IFIP International Conference on, 2011, pp. 1 –6.

[9] Hao, P.; Qiu, W.; Evans, R. Performance Evaluation of IEEE 802.15.4 MAC in Beacon-Enabled Tree-Topology Wireless Sensor Networks. Systems and Networks Communications (ICSNC), 2010 Fifth International Conference on, 2010, pp. 58 –63.

[10] Chen, H.; Tse, C.; Feng, J. Impact of Topology on Performance and Energy Efficiency in Wireless Sensor Networks for Source Extraction. Parallel and Distributed Systems, IEEE Transactions on 2009, 20, 886 –897.

[11] Shrestha, A.; Xing, L. A Performance Comparison of Different Topologies for Wireless Sensor Networks. Technologies for Homeland Security, 2007 IEEE Conference on, 2007, pp. 280 –285.

[12] Mamun, Q.; Ramakrishnan, S.; Srinivasan, B. An Efficient Localized Chain Construction Scheme for Chain Oriented Wireless Sensor Networks. Autonomous Decentralized Systems (ISADS), 2011 10th International Symposium on, 2011, pp. 3 –9.

[13] Ghidini, G.; Das, S. An Energy-Efficient Markov Chain-Based Randomized Duty Cycling Scheme for Wireless Sensor Networks. Distributed Computing Systems (ICDCS), 2011 31st International Conference on, 2011, pp. 67 –76.

[14] Toscano, E.; Lo Bello, L. The case for chain-based routing in industrial wireless sensor networks. Factory Communication Systems (WFCS), 2010 8th IEEE International Workshop on, 2010, pp. 189 –192.

[15] Wu, H.; ming Ding, Y.; Zhong, Z. A Chain-based Fast Data Aggregation Algorithm Based on Suppositional Cells for wireless sensor networks. Power Electronics and Intelligent

Transportation System (PEITS), 2009 2nd International Conference on, 2009, Vol. 1, pp. 106 – 109.

[16] Yu, Y.; Song, Y. An Energy-Efficient Chain-Based routing protocol in Wireless Sensor Network. Computer Application and System Modeling (ICCASM), 2010 International Conference on, 2010, Vol. 11, pp. 486– –489.

[17] Yuan, L.; Zhu, Y.; Xu, T. A Multi-Layered Energy-Efficient and Delay-Reduced Chain-Based Data Gathering Protocol for Wireless Sensor Network. Mechtronic and Embedded Systems and Applications, 2008. MESA 2008. IEEE/ASME International Conference on, 2008, pp. 13 –18.

[18] Pham, M.L.; Kim, D.; Doh, Y.; Yoo, S.E. Power aware chain routing protocol for data gathering in sensor networks. Intelligent Sensors, Sensor Networks and Information Processing Conference, 2004, pp. 107–112.

[19] Shin, J.; Suh, C. Energy-Efficient Chain Topology in Ubiquitous Sensor Network. The 10th International Conference on Advanced Communication Technology, (ICACT 2008), 2008, Vol. 3, pp. 1688–1693.

[20] Satapathy, S.; Sarma, N. TREEPSI: Tree based energy efficient protocol for sensor information. IFIP International Conference on Wireless and Optical Communications Networks, 2006, pp. 4–14.

[21] Mamun, Q.; Ramakrishnan, S.; Srinivasan, B. Muli-chain oriented logical topology for wireless sensor networks. Second International Conference on Computer Engineering and Technology, (ICCET 2010), 2010, pp. 367–372.

[22] Luo, H.; Tao, H.; Ma, H.; Das, S.K. Data Fusion with Desired Reliability in Wireless Sensor Networks. IEEE Transactions on Parallel and Distributed Systems 2011, 22, 501–513.

[23] Wu, H.; Ding, Y.M.; Zhong, Z. A Chain-based Fast Data Aggregation Algorithm Based on Suppositional Cells for wireless sensor networks. International Conference on Power Electronics and Intelligent Transportation System, (PEITS 2009), 2009, Vol. 1, pp. 106–109.

[24] Lindsey, S.; Raghavendra, C. PEGASIS: Power-efficient gathering in sensor information systems. IEEE Aerospace Conference, 2002, Vol. 3, pp. 1125–1130.

[25] Mamun, Q.; Ramakrishnan, S. SecCOSEN: A Key Management Scheme for Securing Chain Oriented Sensor Networks. The 6th Annual Communication Networks and Services Research Conference, (CNSR 2008), 2008, pp. 584–592.

[26] Lindsey, S.; Raghavendra, C.; Sivalingam, K. Data gathering in sensor networks using the energy*delay metric. The 15th International Symposium on Parallel and Distributed Processing, 2001, pp. 2001–2008.

[27] Du, K.;Wu, J.; Zhou, D. Chain-based protocols for data broadcasting and gathering in the sensor networks. International Parallel and Distributed Processing Symposium, 2003, 2003, pp. 8–13.

[28] Liu, R.; Rosberg, Z.; Collings, I.; Wilson, C.; Dong, A.; Jha, S. Overcoming radio link asymmetry in wireless sensor networks. IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications, (PIMRC 2008), 2008, pp. 1–5.

[29] Lindsey, S.; Raghavendra, C.; Sivalingam, K. Data gathering algorithms in sensor networks using energy metrics. IEEE Transactions on Parallel and Distributed Systems 2002, 13, 924–935.

[30] Fei, H.; Ziobro, J.; Tillet, J.; Sharma, N.K. Secure Wireless Sensor Networks: Problems and Solutions. Systemics, Cybernetics and Informatics 2005, 1, 90–100.

[31] Heinzelman, W.; Chandrakasan, A.; Balakrishnan, H. Energy-efficient communication protocol for wireless microsensor networks. Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, 2000, 2000, Vol. 2, pp. 10–20.

[32] Tabassum, N.; Mamun, Q.; Urano, Y. COSEN: A Chain Oriented Sensor Network for Efficient Data Collection. Third International Conference on Information Technology: New Generations, (ITNG 2006), 2006, pp. 262–267.

Author

Quazi Mamun is a Lecturer in the School of Computing and Mathematics, Charles Sturt University, NSW, Australia. He earned BSc Engineering degree in Computer Science and Engineering from Bangladesh University of Engineering and Technology (BUET), Masters degree in Global Information and Telecommunication Studies from Waseda University Japan, and PhD degree from Monash University, Australia. Quazi's research interests include, but not limited to, distributed systems, ad hoc and sensor networks, wireless networks, and network security. He is an active member of IEEE.