Available online at www.sciencedirect.com
ScienceDirect
Computer Communications 29 (2006) 2469–2481
computer communications
www.elsevier.com/locate/comcom

Sleep scheduling for wireless sensor networks via network flow model

Rick W. Ha [a,*], Pin-Han Ho [a], X. Sherman Shen [a], Junshan Zhang [b]

[a] Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ont., Canada N2L 3G1
[b] Department of Electrical Engineering, Arizona State University, Tempe, AZ 85287, USA

Available online 3 March 2006

Abstract

The pervasiveness and operational autonomy of mesh-based wireless sensor networks (WSNs) make them an ideal candidate in offering sustained monitoring functions at reasonable cost over a wide area. There has been a general consensus within the research community that it is of critical importance to jointly optimize protocol sublayers in order to devise energy-efficient, cost-effective, and reliable communication strategies for WSNs. This paper proposes a cross-layer organizational approach based on sleep scheduling, called Sense-Sleep Trees (SS-Trees), that aims to harmonize the various engineering issues and provides a method to increase the monitoring coverage and the operational lifetime of mesh-based WSNs engaged in wide-area surveillance applications. An integer linear programming (ILP) formulation based on network flow model is provided to determine the optimal SS-Tree structures for achieving such design goals.

Keywords: Wireless sensor networks; Sleep scheduling; Cross-layer design; Integer linear programming; Network flow model

1. Introduction

Reliability has always been one of the top concerns in communications system design. With the advent of wireless sensor networks (WSNs) [1–3], the notion of reliability takes on a whole new meaning because of the unique operational characteristics introduced by WSNs that set them apart from other networks. First of all, the emphasis on *energy efficiency* is paramount in the majority of WSN applications because each sensor node is equipped with only a finite amount of battery supply. As the operational reliability of each node is completely compromised once its energy is depleted, the highest priority in WSN design is to prolong system lifetime as much as possible through the use of energy-efficient hardware components and power management techniques. Second, since each WSN is expected to be comprised of thousands of nodes or more, per unit *cost* hence becomes a major factor in sensor node design and component selection. A balance must be struck between choosing low-cost yet perhaps inferior components and maintaining sensing application reliability without making WSNs excessively expensive. In light of the influences of energy efficiency and cost, a few primary implications of reliability in WSN design are as follows:

- *Hardware reliability*. This measure is related to the propensity of the onboard hardware components in succumbing to failure during normal WSN operations. While the specifics on WSN hardware design for maximizing reliability are beyond the scope of this paper, engineering intuition suggests that it is a good idea to select hardware components that are as simple in architecture as possible.
- *Sensing reliability*. In WSN applications, all sensor nodes cooperate together to monitor physical phenomena of interest across the sensor field. As individual nodes can sense the appropriate physical phenomena within their sensing range only, important events may be missed by all sensor nodes because of possible inadequate sensing coverage. Therefore, providing comprehensive sensing coverage requires meticulous network planning and node deployment strategies. Redundant nodes may be introduced to the sensor field to offer

* Corresponding author.
E-mail addresses: rwkha@bbcr.uwaterloo.ca (R.W. Ha), pinhan@bbcr.uwaterloo.ca (P.-H. Ho), xshen@bbcr.uwaterloo.ca (X.S. Shen), junshan.zhang@asu.edu (J. Zhang).

0140-3664/$ - see front matter
doi:10.1016/j.comcom.2006.02.009

additional reliability for sensing coverage, though tradeoffs must again be weighed with respect to per node costs and network management complexity.
- *Communication reliability*. In most WSN applications, the overall traffic profile is very simple as packets only flow from sensor nodes to the data sink and vice versa, with very few inter-node exchanges. Despite this simplicity in traffic flows, the WSN is still expected to deliver sensor data and network control messages with high fidelity in a timely fashion. Aside from packet loss effects presiding over unstable wireless links, the inherent multihop nature of WSN communications present additional uncertainty in guaranteeing packet transport reliability that the common protocols used in the Internet paradigm are inadequate to handle. In particular, it remains a considerable challenge to preserve the network connectivity in conjunction with low-duty cycle sleep-scheduling strategies intended for maximum energy conservation.

Summarizing all the design concerns pertinent to guaranteeing WSN reliability, the real engineering challenge henceforth is to devise a comprehensive yet manageable network organization and communication paradigm that can harmonize all of the criteria without creating significant conflicts in optimization objectives. The ultimate design goal is to balance, through a cross-layer organization scheme, the sensing requirements, end-to-end data communication overhead, and network control effectiveness with energy efficiency. This paper concentrates solely on the WSN design for wide-area event-driven surveillance applications, and it is organized in the following manner. First, Section 2 describes related concepts and prior work in achieving energy-efficient and reliable WSN design. Section 3 proposes a new organizational methodology, called Sense-Sleep Trees (SS-Trees), that aims to maximize energy efficiency in WSN design. Next, a network-flow-model-based approach for computing SS-Trees is presented in Section 4. Evaluation on the validity and effectiveness of the proposed SS-Tree computation approach is provided in Section 5. Finally, Section 4 gives some concluding remarks and future research outlook.

2. Related concepts and prior work

Without enacting any energy saving technique during WSN operations, the radio transceiver would typically consume more energy than any other hardware component onboard a sensor node. The study in [4] examined the pattern of energy expenditure in wireless communications for a practical WSN and reported that the bulk of the energy is consumed during idle listening, while the dominant traffic type processed by individual nodes are overheard packets. Aside from avoiding the use of energy-inefficient and complex hardware components, sizable energy savings can be achieved through aggressive power management strategies in devising adaptive sleep schedules at the MAC layer to minimize the amount of energy lost due to needless transceiver idle listening [5,6]. While the main implication of sleep scheduling at the MAC layer is the shortening of the time the radio transceiver is engaged in idle listening, incidences of overhearing can also be reduced as sleeping nodes are no longer eavesdropping on the wireless medium.

For routing algorithms, however, link table entries would expire prematurely if an intermediate node sleeps and shuts off all links to its neighbors without prior notification, thereby forcing frequent packet reroutes. In addition, network maintenance functions such as neighborhood discovery and time synchronization must also operate within the short time frame that the transceiver is active, thereby competing for bandwidth and processing power with other data reporting functions. In the application layer, real-time data reporting functions are subject to constant and debilitating routing path breakages due to random sleeping nodes, while the spontaneity and variability of data generation in event-driven monitoring applications place undue demand on sleep-scheduling effectiveness. Because of these far-reaching effects, a cross-layer perspective should be taken in devising sleep schedules such that the various inter-layer issues can be tackled collectively [7].

Besides sleep scheduling, precious sensor node battery power can be saved by topology control measures that can dynamically adjust the radio transmission range of individual sensor nodes to balance energy efficiency, while maintaining adequate network connectivity [17,18]. However, the energy saving benefits of such topology control techniques can be offset by the messaging and processing overhead in determining the minimum transmission range per node, especially in the event-driven WSN-operating environment characterized by low data rates and long sleep periods. A third approach in reducing energy use for WSN communications is through power-aware routing where packets are routed onto paths with the most residual energy [15,16]. Other cost metrics, such as inter-node distances, transmission delays, channel conditions, and route hop counts, can also be taken into account when determining the minimum cost route across the WSN. One major drawback of this approach is the additional overhead required in disseminating the residual energy or cost information across the WSN for routing updates, which can be much reduced by exploiting the simple traffic profile and low data rates inherent in WSN applications.

For example, a legitimate organizational method for WSNs to minimize upstream and downstream communication costs is to network all the nodes with a large spanning tree structure that is rooted at the data sink. Forwarding messages in a shortest path spanning tree has the advantage of locating a lowest cost path between each of the nodes and the data sink, which enables minimum cost forwarding and source routing to perform effectively [9,10]. Also, junction points are ideal locations for performing data aggregation and in-network processing to reduce upstream traffic volume. Clustering is the best known example of utilizing the tree structure for WSN formulation [11,12], though it requires denser connectivity for proper cluster hierarchy formation. The star topology is also an example of a tree structure where all of the nodes are leaves connected to the data sink via 1 hop, which clearly is not applicable to all types of WSN topologies. A linear chain, used in chain-based routing protocols [13,14], is a special case of a tree where the number of descendents per node is 1.

Because sleep scheduling is an integral part of WSN design, compatibility issues of spanning tree management and sleep scheduling should be investigated with prudence. Random sleep scheduling is not recommended because it would exert a detrimental effect on network connectivity and topology maintenance efficiency under an ultra-low duty cycle (i.e., less than 1% active time per sleep cycle). On the other hand, while implementing a global coordinated sleep schedule for all of the nodes is feasible on a spanning tree structure, a network-wide communication blackout exists during the long sleep periods where none of the nodes would be active for packet forwarding. This lull in communication will adversely impact the monitoring effectiveness in temporal coverage (i.e., timely reporting of emergency events). One possible solution is to reduce the time scale of sleep schedules such that the length of each sleep cycle is shortened while maintaining the same transceiver duty cycle (e.g., 1 ms active time per 1 s vs. 1 s active time per 1000 s for a duty cycle of 0.1%). However, additional control overhead and hardware cost may be incurred in keeping time synchronization tighter [29]. Also, a shorter sleep cycle may cause multihop packet exchanges to span over multiple active periods, thereby complicating routing procedures and lowers communication reliability.

Another way to shorten the communication blackout period while maintaining the percentage of sleep time is via forming a spanning tree with a large number of leaves such that the non-leaf nodes, often referred to as the *connected dominant set* (CDS), form a virtual backbone [19,20]. In the graph theory terminology, a connected dominating set in a graph is a set of connected vertices such that every vertex in the graph either is in the set or has a neighbor in the set. By finding the minimum CDS (MCDS), the entire WSN can theoretically assign the fewest number of active nodes as the virtual backbone and groups of leaf nodes can be then turned on and off successively to provide interleaved coverage while minimizing energy usage through regular sleep scheduling. The main concern with this approach is that it requires the nodes belonging to the MCDS to remain in active mode for longer periods of time to accommodate the varying sleep schedules of the leaf nodes, thereby depleting their battery reserves much sooner.

Given the prevalence of mesh-based topologies in WSN applications, multiple disjoint CDSs (DCDS) can be identified so that through rotating the sleep and active times of all DCDSs, only one DCDS is needed to remain active at any given time to shoulder the communication and sensing responsibilities for the entire WSN. The base problem of partitioning all the nodes into disjoint dominating sets, not necessarily connected, is called a *domatic partition*, and the maximum number of disjoint dominating sets for a given topology is called is *domatic number*. Through maximizing the domatic number for a given WSN topology, each node can enjoy the most sleep time and therefore maximum energy efficiency while both network connectivity and sensing coverage can be preserved. The work in [21] showed that every graph has a domatic partition with $(1 - o(1))(\delta + 1)/\ln n$ dominating sets, where δ and n denote the minimum degree and number of vertices, respectively. The authors in [22] applied this concept to WSNs and proposed approximation algorithms to maximize system lifetime in a clustered environment.

Even though this new perspective on combining sleep scheduling and connectivity is promising, a couple of critical issues deserve further investigation. First, the original problem of maximum domatic partition (MDP) for a given graph does not require the nodes within each dominating set to be connected. Therefore, the results of prior work are only applicable to WSN design if the entire network is fully connected or arranged in a star topology around the data sink, both of which are not practical in most cases. Second, extending the MDP results for a connected topology into a maximum connected domatic partition (MCDP) solution is much more difficult than constructing a MCDS out of a minimum dominating set. Since the complexity of the MDP problem is already known to be NP-hard, solving the MCDP problem would prove to be even more complicated.

No matter how many nodes are to remain active at any given time in a sleep schedule, a minimum level of sensing coverage should be maintained for reliable sensing according to application requirements. Many prior works on the WSN connectivity, especially those stemmed from mobile ad hoc network (MANET) research such as [27,28], paid little attention on the importance of guaranteeing sensing coverage. As mentioned before in Section 1, redundant nodes can be deployed onto the sensor field to compensate for the sleeping nodes as well as to offer additional sensing reliability against node failures [8]. The problem of determining the minimum number of active sensor nodes to provide a certain degree of sensing coverage alludes to the classic set cover problem and its variants [23], though network connectivity should also be considered to form a minimum connected sensor cover [24]. In any case, both sensing coverage and network connectivity should jointly be considered when formulating an optimal sleep schedule that can balance all the application requirements [25,26].

3. System model and the SS-Tree approach

In light of the many design considerations and suggested approaches outlined in the previous section, the following list highlights the key concepts embedded in the current work.

- *Application-specific*. Given the application-specific nature of WSNs, it is rather not practical to seek a one-size-fits-all solution in the WSN design. While some WSN applications require *continuous monitoring* capabilities that generate a constant stream of delay-sensitive data, the current work focuses on *event-driven* WSNs for wide-area surveillance, which monitors infrequent but important events such as fire, intrusion, and other sudden environment abnormalities. The sensor field is assumed to be laden with a large number of identical and immobile nodes, collectively providing adequate sensing coverage with low degree of coverage redundancy. Nodes within a particular area report to a single data sink, which is assumed to possess a more stable energy supply and extra computing power. Event occurring frequency varies with time and changing environment conditions, and there potentially exist long periods of sensing inactivity and even the need for extended node hibernation. A mixture of heterogeneous control and data traffic would contend for limited bandwidth provided by the active nodes. While data reporting format is mostly event-driven, request- and timer-driven types should also be accommodated.
- *Coordinated sleep scheduling*. On average, nodes need to operate at an ultra-low communication duty cycle (<1%) for extending nodal lifetime to multiple years. Without sleep scheduling coordination, sensing and communication reliability cannot be guaranteed under an ultra-low duty cycle without depending on stochastic approaches that compensate the sleep randomness with the deployment of a large number of redundant nodes, which could increase hardware cost and management complexity tremendously. The length of each active period should minimize end-to-end packet propagation delay with reference to application requirements.
- *Near connected domatic partition*. If a node can just shut off its power-hungry transceiver during a sleep period, then it can still monitor events through its alert sensing unit without wasting energy via idle listening. Therefore, the active virtual backbone, along with a subset of inactive nodes connected to the virtual backbone, can provide the necessary sensing coverage during a particular active period. The notion of connected domatic partition presented in Section 2 can be applied in this case to find the optimal number of disjoint or near-disjoint connected dominating sets such that the energy load can be spread out across the WSN while sufficient network connectivity and sensing coverage can still be maintained.
- *Spanning tree structure*. Since packets are assumed to flow either from sensor nodes to the data sink or vice versa with very few direct end-to-end inter-node exchanges, the virtual backbone can take the form of a spanning tree to connect all the active nodes. The main advantage of using a spanning tree is that each node does not need to maintain full link state information for the entire WSN in order to forward the packets correctly. Instead of relying on exquisite routing protocols that require frequent routing table updates, simple flooding procedures that allow the packets to flow either upstream to or downstream from the data sink can be used on the spanning tree without incurring substantial messaging costs. While a tree-like structure is susceptible to breakage caused by failed upstream nodes, communication reliability can still be maintained via other CDSs within the domatic partition provided that each node is aware of the CDS assignment of its 1-hop neighbors.
- *Centralized approach*. Many prior works advocated for the use of distributed and localized approaches to determine everything including sleep schedules, topology management, routing setup, and sensing coverage in WSN design. This way of thinking has its roots in the development of large-scale heterogeneous networks such as MANETs and the Internet, which regard each node to be unique and autonomous. For WSNs, however, all nodes need to cooperate together to perform joint monitoring tasks, and the presence of a central authority (e.g., the data sink) is vital to provide critical information such as application requirements and accurate global time. Therefore, it is logical to let the data sink be actively involved in WSN management for better communication and sensing reliability. Issues of scalability and robustness, often brought up in centralized approaches, will be discussed later on.
- *Cross-layer design*. Finally, because of the use of ultra-low duty cycle sleep scheduling, all the necessarily control and data packet exchanges, no matter how infrequent they may be, are to be conducted within the tiny fraction of the time allotted as active period. In order to complete all packet transmission and forwarding activities within a single and short active period, the data reporting process should also be carefully choreographed, instead of letting the nodes transmit packets at will to further reduce management overhead. In addition, aggressive data aggregation and duplicate suppression should be enforced on the spanning tree structure so that the processing load can be distributed as evenly as possible across the sensor field to prevent premature battery depletion, especially for nodes located closer to the data sink. Such measures require cross-layer collaboration among application requirements, routing procedures and MAC design.

In addition to the key concepts listed above, the current work also makes a couple of assumptions regarding hardware selection. First of all, each node is equipped with very simple transceivers that only permit a fixed transmission range. Second, a single-channel CSMA-based MAC is used for wireless channel access. Third, no geographical location information through GPS or radio ranging techniques is readily available at sensor nodes. All three assumptions are made to reduce overall hardware costs, though the proposed scheme would still work well, maybe even better, under more favorable operating conditions with variable range transceivers, multi-channel MAC, and location identification chipsets available.

Fig. 1(a) shows a simple WSN with a data sink and 9 nodes arranged in a square grid pattern. Suppose that a spanning tree with 3 branches is logically overlaid on top of the original topology, and all 9 nodes follow the same global sleep schedule, as shown in Fig. 1(b) and (c), respectively. Then during the active period, considerable amounts of overhearing and packet collisions, represented as the dashed lines in Fig. 1(b), would occur amongst neighboring nodes. In contrast, none of the nodes will be capable of communicating during the sleep period, which render the WSN useless if it were to detect signs of abnormality occurred during that span. Only until when the next active period appear can the node pass on the urgent notification to the data sink.

Now suppose that the 9 nodes are divided 2 groups of 3 and 6, respectively, in the manner shown in Fig. 1(d), where each group follows its own sleep schedule such that the active periods of each group alternate, as illustrated in Fig. 1(e). The nodes of each group are arranged to form a tree rooted at the data sink with much sparser branches such that nodes on separate branches cannot communicate with one another. With fewer neighbors per active node, incidences of overhearing and packet collisions would be drastically reduced even with the use of only a single wireless channel, which translates into energy savings. Since the nodes on each tree share the same sensing and sleeping cycle, the tree itself is named as *Sense-Sleep Tree*, or *SS-Tree* for short.

Besides achieving energy savings from simplifying the WSN topology, notice that in Fig. 1(d) the sleeping nodes, colored as white, are strategically located beside at least 1 branch of the other SS-Tree that is effectively a CDS. As mentioned before, although the nodes turn off their radio transceivers when they enter sleep periods, they can remain on alert for signs of abnormality or emergency events in the meantime. Since each sleeping node is connected to an active SS-Tree, then whenever signs of abnormality emerge, the node can instantly switch on their transceivers and forward the emergency notification to the data sink via the active SS-Tree. As different SS-Trees rotate in time to act as the virtual backbone, they avoid overburdening any set of nodes from being the sole virtual backbone. In Fig. 1(e), the SS-Tree formation allows the nodes to remain on alert and capable of event reporting 100% of the time even though the transceiver is functioning at a 50% duty cycle, thereby providing twice the level of sensing coverage using about the same amount of energy without making any substantial changes to the WSN.

Despite this advantage in increasing sensing reliability for SS-Trees, one minor drawback is that timer-driven data cannot be simultaneously gathered from all SS-Trees since each follows its own sleep schedule and only one SS-Tree may be active at any given time. For surveillance applications though, the impact of having unsynchronized periodic reports of ambient conditions and operational status of sensor nodes is far less significant than experiencing any delays in alerting the data sink of signs of abnormality and other emergency events. Therefore, besides requiring the WSN application to tolerate a higher delay in timer-driven data reporting, event-driven data are to be given a higher priority in packet delivery that allows it to be expedited to the data sink on the active SS-Tree when both types of data coincide.

Another issue with SS-Trees is their potential inability to provide full spatial and temporal coverage at any given time, due to uneven network connectivity across the WSN such that at least one SS-Tree cannot form a true CDS. Loss of temporal coverage is also caused by the excessive length of each sleep period as mentioned in Section 2, which can be remedied by shortening the sleep cycle or increasing the duty cycle. Fig. 2(a) shows the same basic WSN topology as that used in Fig. 1, but the 9 nodes are now arranged into 3 separate SS-Trees operating under the same low communication duty cycle with the sleep schedules arranged in a staggered manner, as shown in Fig. 2(b). Because of sparse network connectivity, only

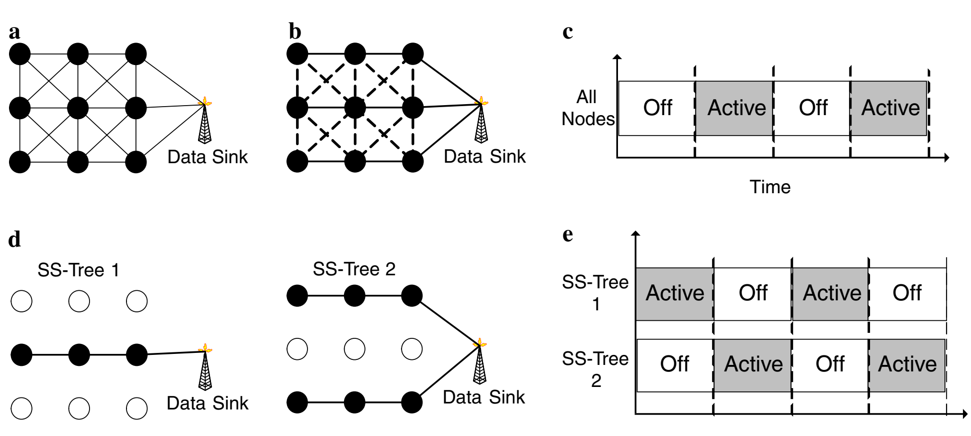


Fig. 1. SS-Tree concept. (a) WSN topology. (b) Logical spanning tree overlay. (c) Global sleep schedule. (d) SS-Tree configuration. (e) Interleaved sleep schedules.

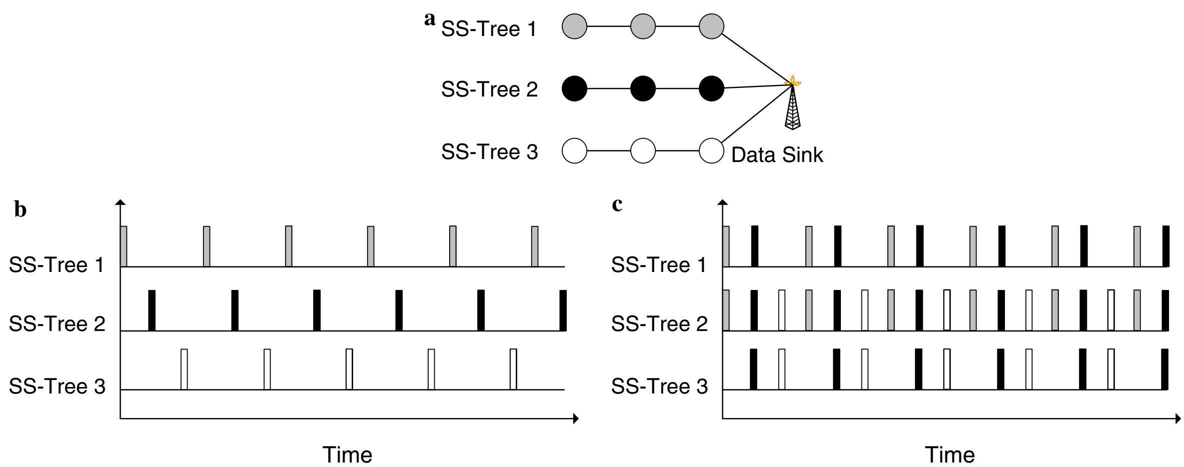


Fig. 2. Impact of SS-Tree on spatial and temporal coverage. (a) SS-Tree assignment. (b) Low duty cycle sleep scheduling. (c) Event reporting windows.

SS-Tree 2 can form a CDS while the other two can be seen as partial CDSs at best. Therefore, there exists an uneven distributing of event reporting windows as illustrated in Fig. 2(c). Allowing dynamic adjustment of transmission ranges would certainly be helpful in reducing the number of partial CDS, and it is relegated as part of future research on SS-Trees.

To realize the benefits of SS-Trees, it is important to devise an efficient method for determining and disseminating the sleep schedule to all of the nodes. The following list of steps concisely describes the process in determining the initial sleep schedules at network initialization.

1. Each node learns of its 1-hop neighbors.
2. Each node forwards local link state information to data sink.
3. Data sink computes optimal SS-Tree structures and sleep schedules with respect to the global connectivity map and application requirements.
4. Data sink disseminates computed sleep schedules to every node through source routing.
5. Each node exchanges sleep schedules with all 1-hop neighbors.
6. Each node follows its received sleep schedule to rotate between active and sleep states.

Since the data sink obtains global knowledge of network connectivity and link costs after Step 3, any rescheduling commands issued by the data sink from then on can be delivered to the nodes swiftly with direct source routing or relying on SS-Trees as efficient broadcast trees. Also, the fact that each node is made aware of its neighbors' sleep schedules after Step 5 ensures robustness against future SS-Tree breakages from upstream nodes. To ensure network integrity, the data sink periodically broadcasts a probing message down each SS-Tree to confirm the well-being of individual nodes. A missed periodic broadcast indicates a high probability of an upstream breakage on a particular SS-Tree branch, and the corresponding nodes can refer to the stored neighborhood sleep schedules and reconnect to the data sink via the next neighboring node that is scheduled to become active. More importantly, the data sink would assume a central role in permanently repairing SS-Trees with the help of the global connectivity and sleep-scheduling information it possesses.

4. Network flow model for computing SS-Trees

The core problem for realizing the SS-Tree concept is the determination of how the sensor nodes can assigned to a predetermined number of SS-Trees on a given WSN topology. With multiple SS-Trees coexisting in a WSN comes the possibility of tree overlapping, where a selected number of sensor nodes may have to belong to multiple SS-Trees to maintain tree connectivity. Such nodes, called *shared nodes*, need to follow multiple sleep schedules since each SS-Tree maintains its own sleep schedule. Therefore, the shared nodes constitute the weak points of the network where they would deplete their batteries much sooner than the rest of the WSN population. While a true MCDP does not contain any shared nodes, the complexity in finding such an ideal solution in reality is rather high, especially for large WSNs that are sparsely or unevenly connected. Therefore, the SS-Tree problem formulation is slightly different than that of MCDP in order to satisfy the practical aspects of WSN design.

Symbols, Let V be the set of all sensor nodes plus the data sink; E be the set of all bidirectional links between elements in V; K be the set of all SS-Trees; s be the symbol representing the data sink in V.

Problem definition. Given an undirected connected graph $G = (V, E)$ with node s denoted as the data sink, form $|K|$ connected subgraphs (SS-Trees), all rooted at node s, with the following main objectives:

1. Minimize the number of shared nodes (i.e., nodes belonging to multiple SS-Trees).
2. Maximize the exposure of each node to other SS-Trees.

One approach to search for an optimal SS-Tree solution is via an integer linear programming (ILP)-based model, which formulates the SS-Tree computation criteria into a set of constraint equations with a minimization goal. The basic idea behind the ILP-based approach, called ILP-Multicommodity Flow (ILP-MF), is to compute the SS-Tree assignments based on the network flow theory with special emphasis on multicommodity flows. First of all, consider Fig. 3, where the 9 nodes in a WSN have been assigned to 2 SS-Trees identified by their respective k values. The fact that each SS-Tree is rooted at the data sink can be viewed as having a flow of type k streaming down from the data sink to the assigned nodes, where each SS-Tree is identified by a distinct flow type as if it carries a separate commodity. Whenever a node is traversed by a flow of type k means that the node belongs to SS-Tree k, and shared nodes are created when flows of different types traverse the same node. The data sink acts as the common source for all of the multicommodity flows and it is designated by the name s in the following ILP formulation. In Fig. 3(a), each node is traversed by flows of either type 1 or 2, which means that no shared nodes exist. However in Fig. 3(b), the center node is traversed by both types of flow, thereby rendering it a shard node.

When working with ILP network flow models, it is important to assign one or more destination nodes to guarantee flow conservation. Simply assigning nodes at the fringes of the WSN as destination nodes is insufficient because this places undue influence on how the SS-Tree flows traverse the network, which would in turn affect the quality of the solution. On the other hand, it is difficult to account for the SS-Tree assignment of each node in the subsequent ILP computation through using a simple network flow model. To satisfy both needs, a virtual node called the *supersink*, denoted as t, is be added to the WSN (i.e., add node t to set V) that connects to every node in the WSN except for the data sink. Its purpose is to become the destination of all flows and its implications on variable declaration will be explained later.

Constants. The following binary constant, named, $a_{i,j}$, is defined to describe the node adjacency map of G:

$$a_{i,j} = \begin{cases} 1 & \text{if node } i \text{ is adjacent to node } j, \\ 0 & \text{otherwise (including when } i = j), \end{cases}$$

where $i,j \in V$. In this model, 2 nodes are said to be adjacent to each other if they can establish a bidirectional radio link. At network initialization, it is at each node's discretion to

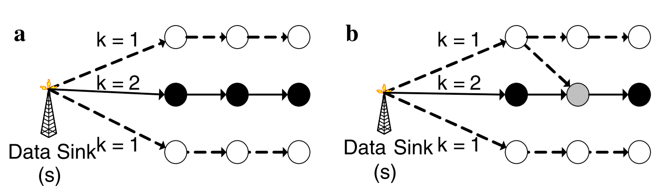


Fig. 3. ILP-Multicommodity Flow concept. (a) No shared nodes. (b) With 1 shared node.

determine the validity of a link with respect to certain predetermined signal-to-noise ratio (SNR) and bit error rate (BER) thresholds.

Parameters. Two integer parameters are defined in the ILP-MF formulation, which are:

$N_{\max}$, maximum number of nodes allowed in any SS-Tree,
$C_{\max}$, maximum number of co-SS-Tree neighbors allowed per node,

where $N_{\max}$ controls the SS-Tree size such that each SS-Tree is to be assigned with a balanced proportion of nodes out of the entire nodal population. The other parameter, $C_{\max}$, serves the purpose of realizing the second objective of maximizing the exposure of each node to other SS-Trees.

Variables. Two types of flow variables are defined in this network flow model. The first type, $f^k(i,j)$, denotes the amount of flow of type k traveling from node i to node j where $i,j \in V$-$\{t\}$ and $k \in K$, and it takes on integer values only. For regulating the flow on the bidirectional link between 2 nodes, *skew symmetry* is defined where for an edge $(i,j) \in E$ and $k \in K$, $f^k(i,j)$ equals $-f^k(j,i)$. Here, the convention is:

$$f^k(i,j) \begin{cases} > 0 & \text{if there is a net positive flow traveling from node } i \text{ to node } j, \\ < 0 & \text{if there is a net positive flow traveling from node } j \text{ to node } i, \\ = 0 & \text{if there is no net flow traveling between nodes } i \text{ and } j, \end{cases}$$

where $i,j \in V$ and $k \in K$. Obviously, $f^k(i,j)$ is equal to 0 if nodes i and j are not adjacent in the underlying network topology, and as well it is defined that the net flow from a node to itself (i.e., $i = j$) is 0.

The second type of flow variable, $f^k(i,t)$, is restricted for representing the flows between every node $i \in V$-$\{s,t\}$ and the supersink t, and its purpose is to define which nodes belong to which SS-Tree:

$$f^k(i,t) = \begin{cases} 1 & \text{if a flow of 1 of type } k \text{ passes from node } i \text{ to supersink } t, \\ 0 & \textit{otherwise}, \end{cases}$$

where $i \in V$-$\{s,t\}$ and $k \in K$. In short, the binary $f^k(i,t)$ variable is specialized for the unique characteristics in the network flow model such that it uses the binary flow concept to manage the SS-Tree assignment of each node. On the other hand, the integer $f^k(i,j)$ variables just account for all the flows of all types destined to the supersink and play a lesser role in satisfying the ultimate minimization objectives.

To illustrate the relationship between the variable types $f^k(i,j)$ and $f^k(i,t)$ in the overall network model of the ILP-Multicommodity Flow approach, Fig. 4 shows a simple 2-node WSN connected to the data sink s and the virtual supersink t. Suppose both nodes belong to SS-Tree 1 (i.e., $K = \{1\}$), then both $f^1(i,t)$ and $f^1(j,t)$ would be equal to 1 according to the above variable type definition. To compensate for this flow demand and maintain flow conservation, a net flow of 2 of type 1 must be supplied by the data sink to node i and then onward to node j, which

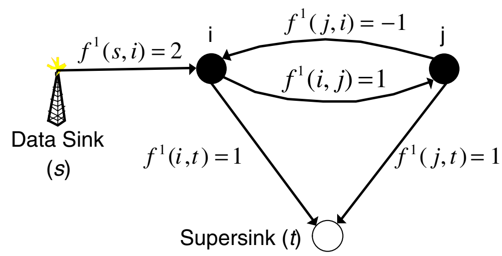


Fig. 4. Flow variable relationship in ILP-Multicommodity Flow approach.

means $f^1(s,i)$ equals 2 and $f^1(i,j)$ equals 1. Therefore, for a given node $i \in V$ (including the data sink and the supersink), the net flow exiting the node is:

$$\sum_{j\in V} f^k(i,j)\begin{cases} >0 & \text{for the data sink (i.e., } i=s),\\ <0 & \text{for the supersink (i.e., } i=t),\\ =0 & \text{for all other nodes,}\end{cases}$$

where $k \in K$. Also according to the skew symmetry property, $f^1(j,i)$ becomes -1 in Fig. 4.

Fig. 5 further explains the use of multicommodity flow variables in determining the nodal assignments for multiple SS-Trees. Suppose that another 2-node WSN is connected to the data sink in a slightly different manner than the previous example, and this time nodes i and j are assigned to SS-Trees 1 and 2, respectively (i.e., $K = \{1,2\}$). Flow variables of type 1 are collectively shown in Fig. 5(a) to show the association of each node to SS-Tree 1. Likewise, Fig. 5(b) displays the flow variables of type 2 to demonstrate each node's relationship to SS-Tree 2. If node i is to be assigned to SS-Tree 2 in addition to SS-Tree 1 as well, then both variables $f^1(i,t)$ and $f^2(i,t)$ would become 1. Therefore, a shared node i is identified by the fact that the sum of all its associated $f^k(i,t)$ variables for $k \in K$ is greater than 1.

In terms of the problem size, the number of variables, N_{var}, involved in the ILP-MF approach, is

$$N_{\text{var}} = |K||V\text{-}\{s,t\}| + 2|E||K|, \tag{1}$$

where the first ($|K||V\text{-}\{s,t\}|$) term accounts for the type of binary variables $f^k(i,t)$ and the latter ($2|E||K|$) term refers to the integer $f^k(i,j)$ variable type.

Objective function. Minimize the number of shared nodes:

$$\min \sum_{k\in K}\sum_{i\in V\text{-}\{s,t\}} f^k(i,t). \tag{2}$$

Constraints. Each node belongs to at least 1 SS-Tree:

$$\sum_{k\in K} f^k(i,t) \geqslant 1, \quad \forall i \in V\text{-}\{s,t\}. \tag{3}$$

The number of nodes per tree is restricted:

$$\sum_{i\in V\text{-}\{s,t\}} f^k(i,t) \leqslant N_{\max}, \quad \forall k \in K. \tag{4}$$

The data sink is the source of all flows:

$$\sum_{k\in K}\sum_{j\in V\text{-}\{s,t\}} f^k(s,j) \geqslant |V\text{-}\{s,t\}|. \tag{5}$$

Flow conservation:

$$\sum_{j\in V} f^k(i,j) = 0, \quad \forall i \in V\text{-}\{s,t\}, \ \forall k \in K. \tag{6}$$

Skew symmetry:

$$f^k(i,j) = -f^k(j,i), \quad \forall i,j \in V, \ \forall k \in K. \tag{7}$$

A node belongs to SS-Tree k if it is traversed by a flow of type k:

$$f^k(i,j) \leqslant f^k(i,t)|V|, \quad \forall i \in V\text{-}\{s,t\}, \quad \forall j \in V\text{-}\{t\}, \ \forall k \in K. \tag{8}$$

The number of co-SS-Tree neighbors per node is restricted:

$$\sum_{j\in V\text{-}\{s,t\}} a_{i,j} f^k(j,t) \leqslant f^k(i,t)C_{\max} + (1 - f_i^k(i,t))|V|, \quad \forall i \in V\text{-}\{s,t\}, \ \forall k \in K. \tag{9}$$

Constraint declarations (3)–(5) are straightforward, and flow conservation and skew symmetry properties of the network flow model are maintained by Constraints (6) and (7), respectively. However, special consideration is to be given to Constraint (8), which requires that a node must belong to SS-Tree k if a flow of type k traverses the node.

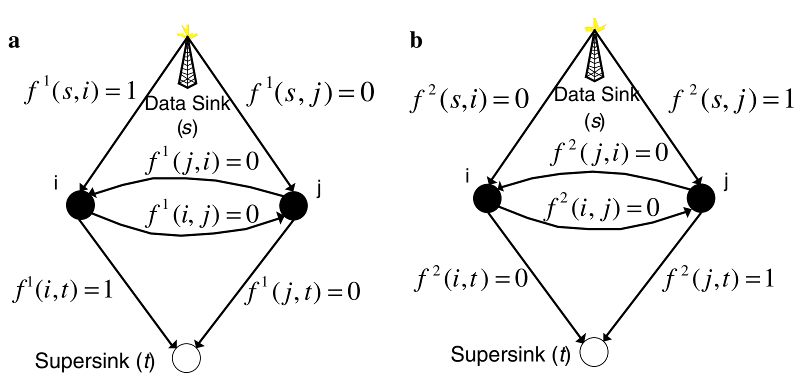


Fig. 5. ILP-Multicommodity Flow model for multiple SS-Tree assignment. (a) Flow variables for SS-Tree 1. (b) Flow variables for SS-Tree 2.

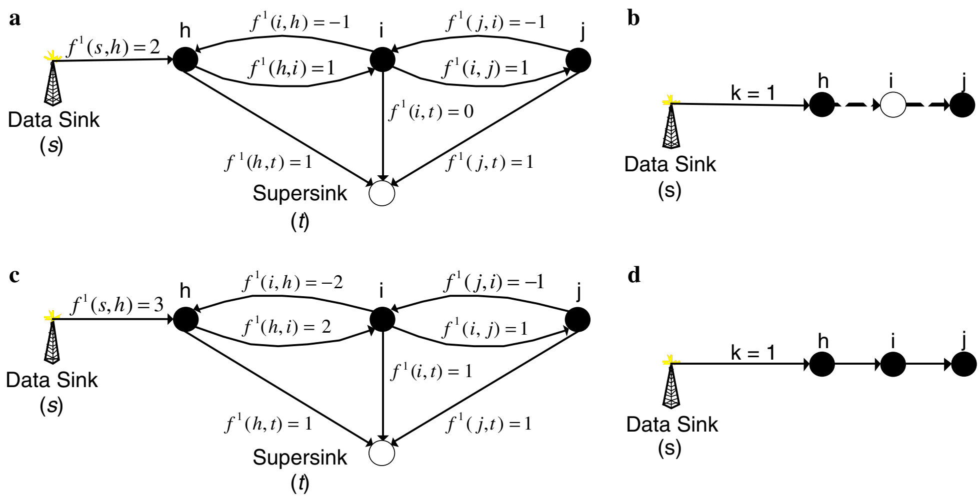


Fig. 6. Explanation on the SS-Tree flow traversal constraint. (a) Incorrect flow assignment. (b) Physical meaning of incorrect flow assignment. (c) Correct flow assignment. (d) Physical meaning of correct flow assignment.

The reasoning behind this constraint is given in Fig. 6, where a 3-node chain WSN is connected to the data sink as well as the virtual supersink. If nodes h and j are assigned to SS-Tree 1, then without the restrictions placed by Constraint (8), the flow variables could be allotted values in the manner shown in Fig. 6(a), where $f^1(i,t)$ is equal to 0 with a flow of type 1 passing through node i. This flow allocation may be perfectly valid in other types of network flow systems and none of the other ILP constraints have been violated. As illustrated in Fig. 6(b), however, the physical meaning of this SS-Tree assignment is that node i would not be sharing the same sleep schedule with nodes h and j, thereby severing the multihop link during the latter nodes' respective active periods. To rectify this assignment error, Constraint (8) limits all net inflow into a node to less than 0 unless $f^2(i,t)$ is 1 for all SS-Trees and for node i not being the data sink or the supersink. Therefore, the combination of flow conservation and this constraint would guarantee the correct flow assignment, as demonstrated in Figs. 6(c) and (d).

Furthermore, Constraint (9) refers to the goal of minimizing the number of co-SS-Tree neighbors per node. The left side of the constraint gives the number of neighbors of node i that belong to SS-Tree k. The right-hand portion of the constraint simply provides a condition that if node i also belongs to SS-Tree k, then the summation on the left side must be less than or equal to $C_{\max}$, which is the maximum number of co-SS-Tree neighbors allowed per node. Otherwise, that summation is less than $|V|$, which essentially means that the number of neighbors of node i that belong to SS-Tree k is inconsequential for the time being.

5. Computation results

The proposed ILP-Multicommodity Flow formulation for computing SS-Trees is solved using CPLEX 9.0 optimization software running on a Dell Precision 450 machine with two 2.8 GHz Pentium 4 Xeon processors and 1 GB RAM under the following assumptions:

1. Two types of WSN topology are used: 8-neighbor square grid (8-*N*, see Fig. 1(a)) and 4-neighbor planar square grid (4-*N*, based on the 8-*N* topology but without diagonal links). The number of nodes per grid edge ranges from 3 to 10.
2. The number of SS-Trees to be computed on each test topology ranges from 2 to 4 (*2-SST* to *4-SST*).
3. The data sink is represented as a node located at or near the center of the grid. Specifically, given n is the number of nodes per grid edge, then the coordinates of the data sink is $(0.5n, 0.5n)$ if n is even, $(0.5(n+1), 0.5(n+1))$ otherwise.
4. The link cost between adjacent nodes is 1, which implies that the total shortest path cost from each node to the data sink is simply its corresponding hop count.
5. The maximum number of co-SS-Tree neighbors allowed per node, $C_{\max}$, is set at 3.
6. The maximum number of nodes allowed per SS-Tree, $N_{\max}$, is empirically set at $\left\lceil 1.2\frac{|V|}{|K|} \right\rceil$.

In the course of the SS-Tree computations, it is evident that CPLEX cannot solve the ILP-MF problem efficiently in all of the test scenarios. Since computation times must be kept within reasonable limits in practical WSN application, a particular test case will be prematurely terminated if it does not return with a solution within 200 s. Table 1 lists the best computation times and network size achieved for each test setup before premature stoppage or after the number of nodes per grid becomes 10. Since the number of variables involved in the SS-Tree computation increases along with the number of SS-Trees as shown in Eq. (1), it is not surprising to see that computation efficiency degrades when more SS-Trees are to be computed for a given

Table 1
Computation times for ILP-Multicommodity Flow approach

Test setup	Best solution time (s)	Nodes per grid edge
4-*N*, 2-SST	78	10
4-*N*, 3-SST	5	7 (premature)
4-*N*, 4-SST	93	6 (premature)
8-*N*, 2-SST	25	10
8-*N*, 3-SST	5	8 (premature)
8-*N*, 4-SST	4	7 (premature)

WSN topology. In any case, there exists a close correlation between the density of the WSN topology, the number of SS-Trees involved, and the actual computation time needed. More in-depth analysis into this relationship is needed as part of future work. In addition, future research work will need to explore into heuristic approaches to expedite the ILP computation process.

Despite its long solution times in general, ILP-Multicommodity Flow formulation does yield the best computation results in terms of minimizing the number of shared node despite its long solution times. In all of the test cases, this approach successfully eliminates the occurrence of shared nodes. With aggressive data aggregation and duplicate suppression, the resultant WSNs should be able to offer a tremendous increase on the amount of monitoring coverage at the same communication duty cycle depending on the number of SS-Trees involved. Another measure of how well SS-Trees are computed is the number of *protected* nodes, which refers to nodes with at least 1 non-co-SS-Tree neighbor. This measure is important as it indicates how large the event reporting window of each node will be and how well nodes can recover from failures with help from such non-co-SS-Tree neighbors. Fig. 7 shows the proportion of nodes protected in each case, and the test cases with 8-neighbor grids all achieved 100% node protection. On the other hand, test cases with sparser 4-neighbor grids and 2 SS-Trees produced the worst protection, though they also incurred the fastest computation time than other 4-*N* counterparts. In all test scenarios, most of the unprotected nodes lie at the fringes of the WSN, and the number of protected nodes will be expected to increase as the nodal population gets larger.

If a protected node is connected all other SS-Trees, then this node is deemed *fully protected*, which means it can offer the highest sensing coverage while enjoying the maximum protection from neighbors for failure recovery. Fig. 8 shows the proportion of fully protected nodes in each test case. For a given topology, the degree of full protection decreases as the number of SS-Trees to be computed increases. On the other hand for a fixed number of SS-Trees, an increase in network density would also increase the degree of full protection. Therefore, careful design consideration should be given when balancing the optimal sensing coverage, network connectivity and the number of SS-Trees involved. A good reference on the optimal number of SS-Trees is the bound on the number of domatic partition on a graph as given in Section 2, though a more definitive bound is still up in the air because of the affinity of the SS-Tree approach to the much involved MCDP problem.

Besides causing an exponentially increase in the solution time, the elimination of shared nodes by the ILP-Multicommodity Flow approach also comes at a price of increased path cost. Assume before computing SS-Trees, the node furthest away from the data sink has a shortest path cost of PC_{max} as determined by Dijkstra's algorithm. Because the ILP-MF formulation does not restrict the path cost of the longest branch on each SS-Tree, the cost of delivering packets from the network fringes to the data sink could become very high under certain conditions. Fig. 9 shows such an effect, which plots the ratio of the PC_{max} values before and after SS-Tree computation. As shown in this figure, the highest path cost after computing SS-Trees could reach as high as 4.33 times of the highest path cost recorded on the original WSN topology. An excessively long end-to-end multihop delivery time would profoundly affect how sleep schedules are devised, which eventually leads to a decrease in the monitoring sensitivity provided by the WSN. Therefore, intuition suggests that some form of adjustable restriction on path cost should be

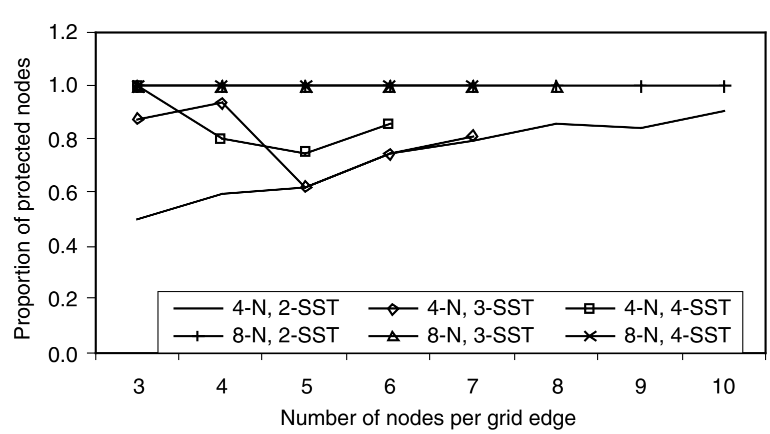


Fig. 7. Proportion of nodes protected using the ILP-Multicommodity Flow approach.

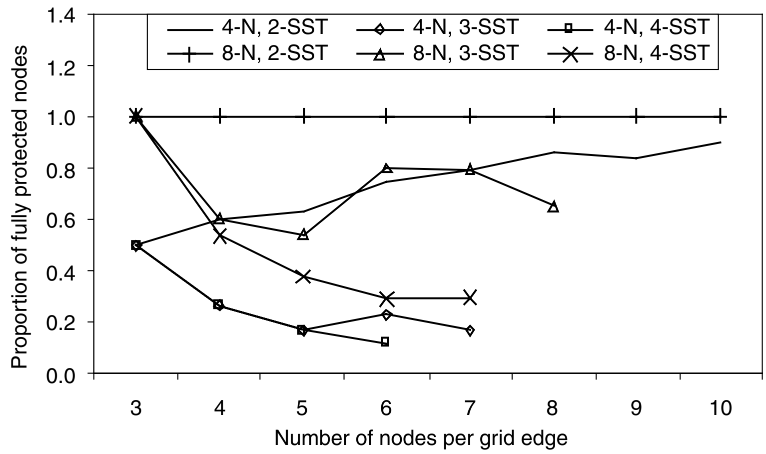


Fig. 8. Proportion of nodes fully protected using the ILP-Multicommodity Flow approach.

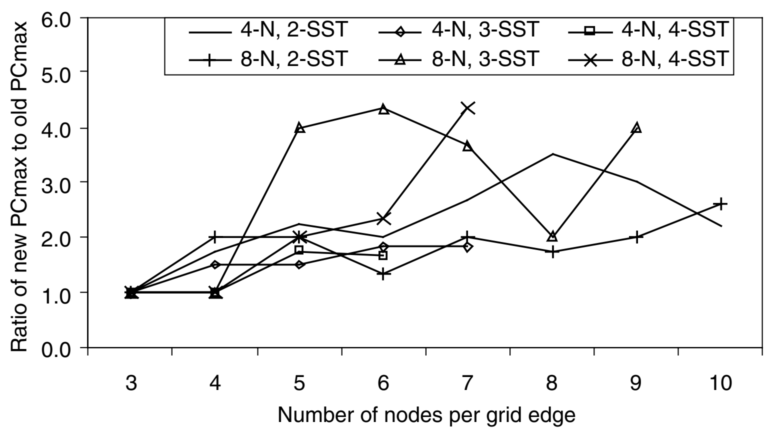


Fig. 9. Increase in path cost after computing SS-Trees using ILP-Multicommodity Flow approach.

implemented in the ILP formulation, which will become part of the ongoing research work.

6. Conclusions and future work

Determining the optimal sensing coverage for energy efficiency mainly depends on the sensing range, network connectivity and sleep schedules of individual nodes. In light of the constraining effects of an ultra-low communication duty cycle in the range of 1% or lower, a sleep-scheduling-based organizational approach called SS-Trees is suggested to minimize the energy usage while providing sufficient monitoring capabilities. An ILP formulation based on the network flow model is proposed to tackle the core problem of determining how the sensor nodes can assigned to a predetermined number of SS-Trees on a given WSN topology. Computation results have shown that the proposed ILP-Multicommodity Flow approach is capable of computing SS-Trees that achieve an increase in sensing coverage, albeit to various degrees. Specifically, the ILP-MF approach completely eliminates shared nodes, though at a price of exponential solution times, increased path cost and low inter-node protection compared to the MCDP ideal.

In the next stage of research, the ILP-MF formulation will serve as a basis of optimality in devising less complex algorithmic approaches for computing optimal SS-Tree structures. One such algorithm is proposed in [30], which achieved faster computation times based on a shortest path tree approach, though the number of shared nodes is not minimized. In addition, the following areas will be explored further:

- Optimal node distribution and transmission range for near-MCDP results using SS-Trees.
- Actual sleep scheduling after SS-Tree computation.
- Refined MAC design for expedited packet delivery.
- Improved neighborhood discovery and failure recovery strategies.

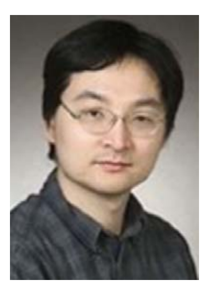
Acknowledgments

The authors sincerely thank the anonymous reviewers for their constructive comments and suggestions.

References

[1] C.-Y. Chong, S.P. Kumar, Sensor networks: evolution, opportunities, and challenges, Proc. IEEE 91 (8) (2003) 1247–1256.
[2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, A survey on wireless sensor networks, IEEE Commun. Mag. 40 (8) (2002) 102–114.
[3] G.J. Pottie, W.J. Kaiser, Wireless integrated network sensors, Commun. ACM 43 (5) (2000) 51–58.
[4] J.M. Reason, J.M. Rabaey, A study of energy consumption and reliability in a multi-hop sensor network, ACM Mobile Comput. Commun. Rev. 8 (1) (2004) 84–97.
[5] W. Ye, J. Heidemann, D. Estrin, An energy-efficient MAC protocol for wireless sensor networks, in: Proceedings of the IEEE INFOCOM, vol. 3, 2002, pp. 1567–1576.
[6] T. van Dam, K. Langendoen, An adaptive energy-efficient MAC protocol for wireless sensor networks, in: Proceedings of the ACM Conference on Networked Sensor Systems, 2003, pp. 171–180.
[7] M.L. Sichitiu, Cross-layer scheduling for power efficiency in wireless sensor networks, in: Proceedings of the IEEE INFOCOM, 2004.
[8] C.-F. Hsin, M. Liu, Network coverage using low duty-cycled sensors: random and coordinated sleep algorithms in: Proceedings of the IPSN'04, 2004, pp. 433–442.
[9] A. Boukerche, X. Cheng, J. Linus, Energy-aware data-centric routing in microsensor networks, in: Proceedings of the ACM MSWiM'03, 2003, pp. 42–49.
[10] U. Cetintemel, A. Flinders, Y. Sun, Power-efficient data dissemination in wireless sensor networks, in: Proceedings of the ACM MobiDE'03, 2003.
[11] W.B. Heinzelman, A.P. Chandrakasan, H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, IEEE Trans. Wireless Commun. 1 (4) (2002) 660–669.
[12] A. Manjeshwar, D. Agrawal, TEEN: a routing protocol for enhanced efficiency in wireless sensor networks, in: Proceedings of the IEEE IPDPS 2001, pp. 2009–2015.
[13] S. Lindsey, C. Raghavendra, K.M. Sivalingam, Data gathering algorithms in sensor networks using energy metrics, IEEE Trans. Parallel Distrib. Syst. 13 (9) (2002) 924–935.
[14] K. Du, J. Wu, D. Zhou, Chain-based protocols for data broadcasting and gathering in the sensor networks, in: Proceedings of the International Parallel Distributed Processing Sym., 2003.
[15] J. Aslam, Q. Li, D. Rus, Three power-aware routing algorithms for sensor networks, Wireless Commun. Mobile Comput. 2 (3) (2003) 187–208.
[16] J.-H. Chang, L. Tassulas, Maximum lifetime routing in wireless sensor networks, IEEE/ACM Trans. Network. 12 (4) (2004) 609–619.
[17] G. Xing, C. Lu, Y. Zhang, Q. Huang, R. Pless, Minimum power configuration in wireless sensor networks, in: Proceedings of the ACM MobiHoc, 2005, pp. 390–401.
[18] R. Ramanathan, R. Rosales-Hain, Topology control of multihop wireless networks using transmit power adjustment, in: Proceedings of the IEEE INFOCOM, 2000, pp. 404–413.
[19] P.-J. Wan, K.M. Alzoubi, O. Frieder, Distributed construction of connected dominating set in wireless ad hoc networks, Mobile Networks Appl. 9 (2) (2004) 141–149.
[20] J. Wu, H. Li, On calculating connected dominating set for efficient routing in ad hoc wireless networks, in: Proceedings of the DIALM, 1999, pp. 7–14.
[21] U. Feige, M.M. Holldorsson, G. Kortsarz, A. Srinivasan, Approximating the domatic number, SIAM J. Comput. 32 (1) (2000) 172–195.
[22] T. Moscibroda, R. Wattenhofer, Maximizing the lifetime of dominating sets, in: Proceedings of the IEEE IPDPS, 2005.
[23] S. Slijepcevic, M. Potkonjak, Power efficient organization of wireless sensor networks, in: Proceedings of the IEEE ICC, 2001, pp. 472–476.
[24] H. Gupta, S.R. Das, Q. Gu, Connected sensor cover: self-organization of sensor networks for efficient query execution, in: Proceedings of the ACM MobiHoc, 2003, pp. 189–200.
[25] Y. Zou, K. Chakrabarty, A distributed coverage- and connectivity-centric technique for selecting active nodes in wireless sensor networks, IEEE Trans. Comput. 54 (8) (2005) 978–991.
[26] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, C. Gill, Integrated coverage and connectivity configuration in wireless sensor networks, in: Proceedings of the ACM SenSys, 2003, pp. 28–39.
[27] A. Cerpa, D. Estrin, ASCENT: Adaptive self-configuring sensor networks topologies, in: Proceedings of the IEEE INFOCOM, 2002, pp. 1278–1287.
[28] B. Chen, K. Jamieson, H. Balakrishnan, R. Morris, Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks, Wireless Networks 8 (2002) 481–494.
[29] S. Rhee, S. Liu, Wireless sensor nets demand trade-offs, EE Times 31 (2003).
[30] R. Ha, P.H. Ho, X.S. Shen, SS-Trees: A cross-layer organizational approach for mesh-based wide-area wireless sensor networks, in: Proceedings of the IEEE BroadNets, 2005, pp. 885–894.

Rick W. Ha received his B.A.Sc. in Computer Engineering and M.A.Sc. in Electrical and Computer Engineering in 2000 and 2002, respectively, from University of Waterloo, Canada, where he is currently pursuing his Ph.D. degree. His research interests include cross-layer design of wireless ad hoc and sensor networks.



Professor Pin-Han Ho received his B.Sc. and M.Sc. degree from the Electrical and Computer Engineering Department of National Taiwan University in 1993 and 1995, respectively. He started his Ph.D. study in the year 2000 at Queen's University, Canada, focusing on optical communications systems, survivable networking, and QoS routing problems. He finished his Ph.D. in 2002, and joined the Electrical and Computer Engineering Department of University of Waterloo, Canada, as an Assistant Professor in the same year. Professor Ho is the first author of more than 40 refereed technical papers and book chapters, and the co-author of a book on optical networking and survivability. He is a recipient of Early Researcher Award (ERA) in 2005.

Xuemin (Sherman) Shen received his B.Sc. degree (1982) from Dalian Maritime University (China) and his M.Sc. (1987) and Ph.D. degrees (1990) from Rutgers University, New Jersey (USA), all in Electrical Engineering. From September 1990 to September 1993, he was first with the Howard University, Washington DC, and then the University of Alberta, Edmonton (Canada). Since October 1993, he has been with the Department of Electrical and Computer Engineering, University of Waterloo, Canada, where he is a Professor and the Associate Chair for Graduate Studies. Dr. Shen's research focuses on mobility and resource management in interconnected wireless/wireline networks, UWB wireless communications systems, wireless security, and ad hoc and sensor networks. He is a coauthor of two books, and has published more than 200 papers and book chapters in wireless communications and networks, control and filtering. Dr. Shen was the Technical Program Co-Chair for IEEE Globecom'03 Symposium on Next Generation Networks and Internet, ISPAN'04, IEEE Broadnets'05, QShine'05, IEEE WirelessCom, and is the Special Track Chair of 2005 IFIP Networking Conference. He serves as the Associate Editor for IEEE Transactions on Wireless Communications; IEEE Transactions on Vehicular Technology; ACM/Wireless Networks; Computer Networks; Wireless Communications and Mobile Computing (Wiley); and International Journal of Computers and Applications. He also serves as Guest Editor for IEEE JSAC, IEEE Transactions Vehicular Technology, IEEE Wireless Communications, and IEEE Communications Magazine. Dr. Shen received the Outstanding Performance Award from the University of Waterloo in 2004, the Premier's Research Excellence Award (PREA) from the Province of Ontario, Canada for demonstrated excellence of scientific and academic contributions in 2003, and the Distinguished Performance Award from the Faculty of Engineering, University of Waterloo, for outstanding contribution in teaching, scholarship and service in 2002. Dr. Shen is a registered Professional Engineer of Ontario, Canada.

Junshan Zhang received his Ph.D. degree from the School of Electrical and Computer Engineering at Purdue University in 2000. He joined the Department of Electrical Engineering at Arizona State University in August 2000, where he is currently an Associate Professor. His research interests fall in the general area of wireless networks, spanning from the networking layer to the physical layer. His current research focuses on fundamental problems in wireless ad-hoc networks and sensor networks, including cross-layer optimization and design, resource management, network information theory, and stochastic analysis. Prof. Zhang is a recipient of the ONR Young Investigator Award in 2005 and the NSF CAREER award in 2003. He has also received the Outstanding Research Award from the IEEE Phoenix Section in 2003. He was chair of the IEEE Communications and Signal Processing Phoenix Chapter from Jan. 2001 to Dec. 2003. He has served as a member of the technical program committees of INFOCOM, SECON, GLOBECOM, ICC, MOBIHOC, BROADNETS, and SPIE ITCOM, and TPC co-chair for the 25th International Performance Computing and Communications Conference. He has served as an Associate Editor for IEEE Transactions on Wireless Communications since 2004.