Optimal sleep scheduling with transmission range assignment in application-specific wireless sensor networks

Rick W. Ha, Pin-Han Ho and Xuemin (Sherman) Shen*

Department of Electrical and Computer Engineering,
University of Waterloo,
ON, Canada N2L 3G1
E-mail: rwkha@bbcr.uwaterloo.ca
E-mail: pinhan@bbcr.uwaterloo.ca
E-mail:xshen@bbcr.uwaterloo.ca
*Corresponding author

Abstract: To extend the functional lifetime of battery-operated Wireless Sensor Networks (WSNs), stringent sleep scheduling strategies with communication duty cycles running at sub-1% range are expected to be adopted. Although ultra-low communication duty cycles can cast a detrimental impact on sensing coverage and network connectivity, its effects can be mitigated with adaptive sleep scheduling, node deployment redundancy and multipath routing within the mesh WSN topology. Prior research on this issue had led to the proposal of a new design paradigm called Sense-Sleep Tree (SS-Tree). The SS-Tree aims to harmonise 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 event-driven surveillance applications. This paper further expands and refines the SS-Tree approach by incorporating practical design considerations such as transmission range assignment, duty cycling and sensing coverage. This approach can achieve higher energy efficiency and satisfy various application requirements during WSN network planning and predeployment phase.

Keywords: Wireless Sensor Networks (WSNs); Sense-Sleep Tree (SS-Tree); transmission range assignment; sleep scheduling; temporal sensing coverage; ultra-low duty cycle.

Reference to this paper should be made as follows: Ha, R.W., Ho, P-H. and Shen, X.S. (XXXX) ‘Optimal sleep scheduling with transmission range assignment in application-specific wireless sensor networks’, *Int. J. Sensor Networks*, Vol. X, No. Y, pp.XXX–XXX.

Biographical notes: Rick W. Ha received a BASc in Computer Engineering and MASc in Electrical and Computer Engineering in 2000 and 2002, respectively, from University of Waterloo, Canada, where he is currently pursuing his PhD. His research interests include cross-layer design of wireless sensor networks, advanced wireless communication networks and mobile device ergonomics.

Professor Pin-Han Ho received a BSc and MSc from the Electrical and Computer Engineering Department of National Taiwan University in 1993 and 1995, respectively. He started his PhD study in 2000 at Queen’s University, Canada, focusing on optical communications systems, survivable networking and QoS routing problems. He finished his PhD in 2002, and joined the Electrical and Computer Engineering Department of University of Waterloo, Canada, as an Assistant Professor in the same year. He 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 a BSc (1982) from Dalian Maritime University (China) and his MSc (1987) and PhD (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. His research focuses on mobility and resource management in interconnected wireless/wired networks, UWB wireless communications systems, wireless security and ad hoc and sensor networks. He is a co-author of three books and has published more than 200 papers and book chapters in wireless communications and networks, control and filtering. He 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*. He 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. He is a registered Professional Engineer of Ontario, Canada.

1 Introduction

Sleep scheduling is an essential part of Wireless Sensor Network (WSN) design because battery-operated sensor nodes that are active 100% of the time will normally consume their entire power supply in a matter of days. For WSN applications that are designed to remain operational over several months or years, a duty cycle that resides below 1% is often needed to regulate the active and sleep times of the various hardware components. Such an ultra-low duty cycle puts undue pressure on the WSN application in maintaining adequate sensing coverage and network connectivity to satisfy sensing requirements. Therefore, devising a viable sleep scheduling scheme that takes into account both energy efficiency and sensing application guarantees becomes an interesting and challenging research topic.

Figure 1 shows the basic hardware architecture for a sensor node, where the emphasis on energy efficiency influences how WSN designers select hardware components to minimise power consumption while balancing processing power, memory requirements, sensor sensitivity and transceiver performance. A practical sleep schedule would selectively activate or deactivate certain components depending on the sensing task at hand, and Table 1 shows the useful assignment of sleep states (Sinha and Chandrakasan, 2001). In typical WSN implementations, the radio transceiver is often the most energy consuming hardware component, which suggests wireless communication should be conducted as sparingly as possible (i.e. minimise time spent in sleep states S_4, S_3 and S_2). However, the loss of communication capabilities not only affects network connectivity in the multihop WSN environment but also reduces the overall sensing coverage. For event-driven and timer-driven data reporting, collected sensor data cannot be timely forwarded to the data sink whenever nodes along the routing path have their transceivers turned off. Likewise, specific data requests issued by the data sink may not be delivered to the targeted nodes promptly due to prolonged sleep periods of intermediate nodes.

On the other hand, in single-hop or clustered WSN environment where network connectivity is usually dense, prolonged sleep periods of individual nodes have little effect on the packet forwarding capability assuming the data sink and clusterheads have extended power supply readily available. Therefore, the nodes can operate in either sleep state S_0 or S_1 for most of the time to minimise energy loss through transceiver activity. Nevertheless, realising dense network connectivity requires the support of longer transmission range relative to the physical distribution of the nodes, which entails greater energy consumption due to higher transmission power as well as increased likelihood of packet collision and overhearing in single-channel transmission.

Figure 1 Basic hardware architecture of a sensor node

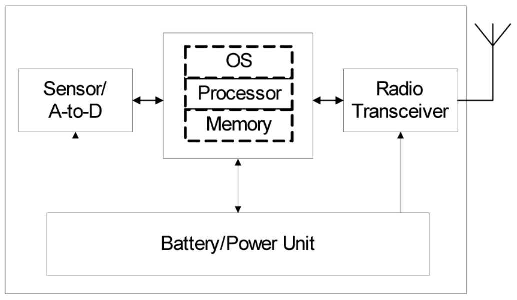


Table 1 Useful sleep state assignment

Sleep state	*Clock*	*Processor*	*Memory*	*Sensor/A-to-D*	*Transceiver*
S_4	Active	Active	Active	Active	*T*x, *R*x
S_3	Active	Idle	Sleep	Active	*R*x/listen
S_2	Active	Sleep	Sleep	Active	*R*x/listen
S_1	Active	Sleep	Sleep	Active	Sleep
S_0	Active	Sleep	Sleep	Sleep	Sleep

Besides sleep scheduling, other approaches to minimise energy consumption include topology control and power-aware routing (Xing et al., 2005). The former deals with finding the optimal transmission range of individual nodes, often dynamically after deployment, such that the resultant network topology is the most energy efficient, while the latter is associated with computing optimal packet delivery routes across the network based on communication costs and residual energy information such that overall system lifetime can be maximised. Even though both approaches offer some energy savings, energy loss due to transceiver idle listening still dominates in typical WSN applications that are characterised by prolonged monitoring and low data traffic if sleep scheduling is not implemented. Furthermore, such energy savings are achieved through extra hardware features such as transceivers with dynamically adjustable transmission power and fine-grained battery level monitors, implying additional design trade-offs in cost-conscious WSN applications.

Because ultra-low duty cycle sleep scheduling casts a far-reaching impact over different aspects of WSN design, a cross-layer methodology that jointly considers MAC design, routing strategies and application requirements should be advocated. Our previous research has brought up the concept of Sense-Sleep Tree (SS-Tree), whose ultimate goal is to balance, through a cross-layer organisation scheme, the sensing requirements, end-to-end data communication overhead, and network control effectiveness with energy efficiency for wide-area event-driven WSN surveillance applications. This paper further extends the SS-Tree concept and focuses on two areas that have not yet been studied in detail, namely determining the actual sleep schedules with respect to sensing requirements and implementing transmission range assignment in relationship to node distribution. The main objective is to analyse the effects of changing the various design parameters during predeployment phase on energy efficiency, network connectivity and sensing coverage.

The rest of this paper is organised as follows. Section 2 outlines some important system model considerations regarding the WSN design. Section 3 presents the SS-Tree concept and explains how it is connected to the various WSN design aspects. Section 4 follows with detailed numerical results to evaluate the validity and effectiveness of the proposed approach. Related work is provided in Section 5. Finally, Section 6 offers some concluding remarks and outlook for future work.

2 System model considerations

2.1 Spatial sensing coverage

During the WSN planning stage, engineers have to contend with a wide range of design alternatives to satisfy sensing application requirements. Consider the following scenario: A number of identical and immobile sensor nodes are deployed over an area $L \times W$ with the data sink located at the center of the sensing field. The method of node distribution around the data sink is either deterministic (grid or non-regular) with N nodes or Poisson with density λ. The exact manner and scale of node distribution depend on the sensing range of each node and the desire degree of overlapped spatial sensing coverage over the sensing field, where redundant deployment of nodes mainly serves the following three purposes:

1 provide multiple independent observations for a given event (i.e. *k*-coverage)

2 cover for failed nodes and

3 cover for sleeping nodes.

Intuitively, high spatial sensing coverage redundancy provides higher reliability in event detection and more flexibility in dealing with node failures. Hsin and Liu (2004) further propose to rotate the sleep schedules of overlapping nodes to extend battery lifetime while maintaining sufficient spatial sensing coverage at any given time. In other words, the entire node population is divided into disjoint sets, each provide adequate coverage while adhering to different sleep schedules.

In the system model for the SS-Tree concept, the sensing field can be *k*-covered, that is, every point of interest is covered by at least k nodes, depending on application requirements. However, sleep rotation of nodes with overlapping spatial sensing coverage will not be pursued. This is because the purported energy efficiency improvement comes at the expense of increased hardware and deployment costs as the degree of coverage overlapping required is at least several times than the minimum level for this sleep scheduling approach to work properly. Also, distributed computation of the optimal spatial sensing cover after node deployment would inevitably require accurate location information, which may not be available at the local level because of high costs and operational limitations with location-aware techniques such as GPS. Given that the spatial sensing cover is to be fixed over time, the main challenge therefore is to maximise energy efficiency, guarantee adequate sensing coverage and reduce the scale of node deployment so that large-scale WSN applications can be made more economically viable.

2.2 Temporal sensing coverage

Besides spatial sensing coverage, temporal sensing coverage which is related to the timing and promptness of data reports, is equally important in sensing applications. For event-driven WSN surveillance applications, the time it takes for the data sink to be notified of an event of interest appearing somewhere in the sensing field often has to be bounded by a threshold. Aside from satisfying application requirements, WSN designers have to take into account the various delays introduced during sensor sampling, processing and communication in determining the optimal threshold value. While delay metrics can be improved by selecting hardware components with the best timing performance, the necessity of enacting ultra-low duty cycle sleep management will inevitably drive up data communication latency significantly.

Consider the sensing field is 1-covered (i.e. every point of interest is covered by at least 1 node). Suppose the node that detected the event follows a global regular sleep schedule that only involves sleep states S_4 (i.e. every component remains active) and S_0 (i.e. every component except the clock sleeps), where data sampling and processing time, can be completed by the sensing unit within 1 active period. If the duration of the event is longer than 1 sleep cycle, then the following is the sequence of steps that may take place in event detection and notification:

1 node wakes up

2 node detects event

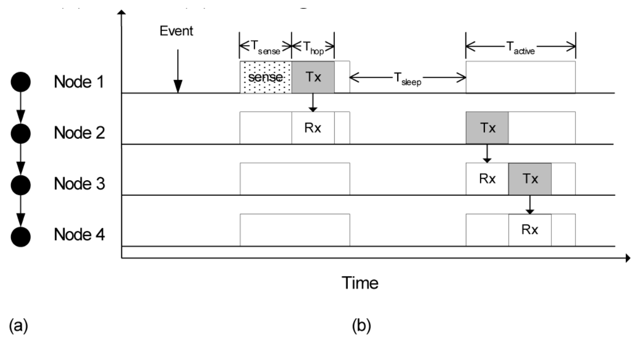
3 node generates notification packet

4 intermediate nodes forward the notification packet and

5 notification packet arrives at the data sink.

Figure 2(a) shows a linear chain of sensor nodes and Figure 2(b) illustrates the timing diagram of how an event is detected by node 1 and the corresponding notification packet is sent to node 4 through nodes 2 and 3. In the timing diagram, T_{sense} is the amount of time to perform data sensing and processing. T_{active} and T_{sleep} denote the time allocated for active and sleep periods, respectively. It is assumed that it takes T_{hop} on average to pass the notification packet over 1 hop, whose specific value depends on packet size, processing power, transmission bandwidth, MAC signaling, modulation schemes and general channel conditions. It is also assumed that the sensing to transmission switching times and packet processing times in intermediate nodes are negligible. Depending on the length of T_{active}, a node may be able to perform data sensing and processing, as well as forwarding the notification to the data sink over multiple hops in a single active period. For smaller T_{active} values, the end-to-end delivery process may span over several sleep cycles, thereby increasing end-to-end latency.

Figure 2 Event-driven data reporting timing: (a) route (b) timing



Let duty cycle, ρ , be defined as:

$$\rho = \frac{T_{\text{active}}}{T_{\text{active}} + T_{\text{sleep}}} \tag{1}$$

If the node is N_{hop} hops away from the data sink, then the expected elapsed time, $\overline{T_{\text{event}}}$, between initial appearance of the event, assuming uniform probability of event occurrence over time and the subsequent notification arriving at the data sink is approximately:

$$\overline{T_{\text{event}}} = \begin{cases} \dfrac{T_{\text{active}}}{2\rho} + N_{\text{hop}} T_{\text{hop}} & \text{for } N_{\text{hop}} <= \alpha \\ \dfrac{T_{\text{active}}}{2\rho} + \left\lfloor \dfrac{N_{\text{hop}} - \alpha}{\beta} \right\rfloor \dfrac{T_{\text{active}}}{\rho} - T_{\text{sense}} & \text{otherwise} \\ + \left[\left(N_{\text{hop}} - \alpha \right) \bmod \beta \right] T_{\text{hop}} & \end{cases} \tag{2}$$

where $\alpha = \left\lfloor T_{\text{active}} - T_{\text{sense}} / T_{\text{hop}} \right\rfloor$ is the average number of hops traversable in the initial active period the event is first detected, and $\beta = \left\lfloor T_{\text{active}} / T_{\text{hop}} \right\rfloor$ is the maximum number of hops traversable in 1 active period.

For time-critical sensing applications, the various input parameters can be manipulated to minimise $\overline{T_{\text{event}}}$. However, changes in parameter value will lead to profound ramifications on other WSN design aspects such as energy efficiency and cost. For example, according to Equation (2), $\overline{T_{\text{event}}}$ can be reduced by decreasing N_{hop}, which can be done either through installing more data sinks in the sensing field to shorten mean physical distance between nodes and their nearest data sink, or by increasing the transmission range of each node to produce denser network connectivity. The first approach would obviously push up deployment costs and network maintenance complexity, while the second approach would require higher energy consumption, potentially higher component costs, and potentially higher complexity in resolving the resultant increase in neighbourhood interference. More detailed discussions on parameter selection and trade-off will be provided later on.

Even for event-driven WSN applications, there still exists the need to provide both timer-driven and request-driven data reporting functions. The first two types are similar as they involve data packets traveling in the upstream direction to the data sink, provided that packet acknowledgement is done hop-by-hop rather than end-to-end. However, timer-driven data reporting often needs to be coupled with data aggregation schemes for energy efficiency, so the sequence of steps that would take place in timer-driven data reporting could resemble the following list:

1 node wakes up

2 internal timer triggers data reporting function

3 node furthest away from data sink generates initial data reporting packet from cached sensor values

4 intermediate nodes forward the data reporting packet, performing data aggregation along the way and

5 data reporting packet arrives at the data sink.

If the furthest node in the data aggregation chain is N_{hop} hops away from the data sink, then the expected elapsed time, $\overline{T_{\text{timer}}}$, between the initial transmission of the data reporting packet and the aggregated data reporting packet arriving at the data sink is approximately:

$$\overline{T_{\text{timer}}} = \left\lfloor \frac{N_{\text{hop}}}{\beta} \right\rfloor \frac{T_{\text{active}}}{\rho} + \left(N_{\text{hop}} \bmod \beta \right) T_{\text{hop}} \tag{3}$$

Note that the processing times for data aggregation in intermediate nodes are assumed to be negligible because of the expected small data packet size and reasonably capable sensor node microcontrollers. On the other hand, if the furthest node can only generate one timer-driven data reporting packet in each active period, then the minimum timing separation between consecutive timer-driven data reports, $T_{\text{min\_timer}}$, is simply:

$$T_{\text{min\_timer}} = \frac{T_{\text{active}}}{\rho} \tag{4}$$

Compared with the other two data reporting types, request-driven data reporting is perhaps the most time-consuming as it usually involves packet exchanges in both downstream and upstream directions. The typical messaging sequence could be:

1 data request arrives at data sink for querying data at a particular node

2 data sink forwards data request down the appropriate multihop path

3 targeted node receives data request

4 targeted node generates data reporting packet from cached sensor values

5 intermediate nodes forward the data reporting packet and

6 data reporting packet arrives at the data sink.

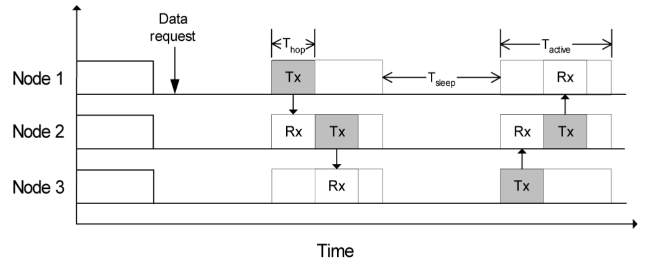
Notice that for both timer-driven and request-driven data reporting, cached sensor values instead of instantaneous data readings are sent in order to save time. As long as the cached values are being constantly updated and the readings remain valid at least over several sleep cycles, this arrangement should be acceptable in most sensing applications. Figure 3 shows a timing example for request-driven data reporting on the linear network chain depicted in Figure 2(a). Here, Node 1 assumes the role of the data sink and receives a data request from the WSN application that would like to query data at Node 3. In response, Node 1 sends out the data request at the next available active period towards Node 3 via Node 2. The selection of the length of T_{active} in this example allows the data request to be delivered to Node 3 within one active period. In the subsequent active period, Node 3 replies with the appropriate data report back to Node 1. Again, packet forwarding to the next hop may be relegated to the next active period if there is not enough time left in the current active period to complete one successfully.

Assuming the data requests arrive randomly with uniform probability over time, then the expected elapsed time, $\overline{T_{\text{req}}}$, for the data query to be completed at the targeted node that is N_{hop} hops away from the data sink is approximately:

$$\overline{T_{\text{req}}} = \begin{cases} P_1\left[\dfrac{(1-\rho)T_{\text{active}}+\rho T_{\text{hop}}}{2\rho}+\left\lfloor\dfrac{2N_{\text{hop}}}{\beta}\right\rfloor\dfrac{T_{\text{active}}}{\rho} \right. & \text{for} \\ \left. +\left(N_{\text{hop}} \bmod \beta\right)T_{\text{hop}}\right]+2P_2N_{\text{hop}}T_{\text{hop}} & N_{\text{hop}} <= \dfrac{\gamma}{2} \\ P_1\left[\dfrac{(1-\rho)T_{\text{active}}+\rho T_{\text{hop}}}{2\rho}+\left\lfloor\dfrac{2N_{\text{hop}}}{\beta}\right\rfloor\dfrac{T_{\text{active}}}{\rho} \right. & \\ \left. +\left[\left(2N_{\text{hop}}\right) \bmod \beta\right]T_{\text{hop}}\right]+P_2\left[\left\lfloor\dfrac{2N_{\text{hop}}-\gamma}{\beta}\right\rfloor\dfrac{T_{\text{active}}}{\rho}\right. & \text{otherwise} \\ \left. +\left[\left(2N_{\text{hop}}-\gamma\right) \bmod \beta\right]T_{\text{hop}}\right] & \end{cases} \tag{5}$$

where $P_1 = (1-\rho)T_{\text{active}} + \rho T_{\text{hop}} / T_{\text{active}}$ is the probability that the data request arriving at the data sink has to be processed in the next active period, $P_2 = \rho(T_{\text{active}} - T_{\text{hop}}) / T_{\text{active}}$ is the probability that the data request arriving at the data sink can be processed instantaneously, and $\gamma = \lfloor T_{\text{active}} + T_{\text{hop}} / 2T_{\text{hop}} \rfloor$ is the average number of hops traversable in the initial active period the data request is received.

Figure 3 Request-driven data reporting timing



So far, the average timing requirements for event-driven, timer-driven and request-driven data reporting types have been discussed. Since most sensing applications require all types of data reporting to be completed within a certain amount of time, the objective therefore is to minimise all of these measures through manipulating the various design parameters. Some of these parameters such as data sampling time by the sensing unit and data processing speed at the microcontroller are hardware-specific and their selection is beyond the scope of this paper. Instead, parameters related to communications and sleep scheduling such as N_{hop}, T_{hop}, T_{active} and ρ will be studied further later on.

2.3 Transmission range assignment

Besides spatial and temporal sensing coverage, another important consideration in WSN design is network connectivity. The measure of N_{hop} usually indicates how dense a network is, whose values are dictated by the physical location of the node in relationship to the data sink, node deployment density in the sensing field, and the level of transmission power. A dense network with low N_{hop} values can provide higher communication reliability through multipath routing over shorter hops, though it comes at the expense of increased wireless interference, route selection overhead and power consumption during data transmission. With sleep scheduling, on the other hand, nodes on redundant routes can enter sleep states, thereby reducing idle energy usage and incidences of packet overhearing. Choosing the appropriate N_{hop} is important for balancing temporal spacing coverage requirements as well as network connectivity.

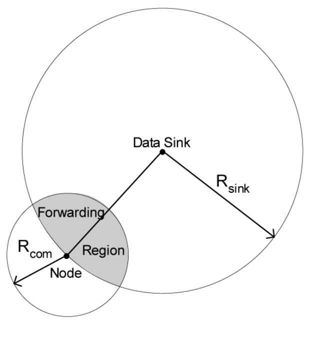
Optimising N_{hop} after node deployment is a rather straightforward task if the nodes are enabled to dynamically adjust their transmission ranges. It would first involve finding out the overall network connectivity and power consumption profile with every node transmitting at full power, and then try to converge to the optimal N_{hop} values by selectively or collectively reducing the transmission power of each node. In spite of its simplicity, the amount of computation and messaging overhead involved can become quite considerable. Also, installing transceivers with dynamically adjustable transmission power on every node can become relatively expensive compared with the case where a simple design with a fixed transmission range is adopted. Still, because of the lack of network configuration flexibility with fixed transmission ranges, it is important to determine the optimal R_{com} value during the node predeployment phase such that temporal sensing coverage, network connectivity and energy efficiency can be balanced.

The following derivation is inspired by the work presented by Takagi and Kleinrock (1984) on optimal transmission ranges for packet radio terminals. Assume that for certain applications nodes are Poisson-distributed with density λ over a two-dimensional space of area $L \times W$, and that each is equipped with a transceiver of fixed communication range R_{com}. Therefore, the expected number of nodes contained within a radius R is $\lambda\pi R^2$, while the probability of having i nodes distributed in an arbitrary area of size A on this sensing field is:

$$\Pr(X = i) = \frac{(\lambda A)^i}{i!} e^{-\lambda A} \tag{6}$$

If all the data packets traveling upstream from a node are always forwarded to the neighbour that is the closest to the data sink, then the forwarding region of a node is defined as the intersected area of two circles, one centred around the node with radius R_{com} and the other around the data sink with radius R_{sink}, as shown in Figure 4, where R_{sink} is defined as the distance between the node and the data sink. This greedy forwarding strategy may not always yield the best and the shortest end-to-end routing path in actual WSN implementation, but it serves as a sufficient model to analyse the relationship between R_{com} and N_{hop}, especially during the node predeployment phase.

Figure 4 Forwarding region of a node in relationship to data sink



A neighbour located in the forwarding region of a node is said to have provided a progress of x to the node if it is (R_{sink}–x) from the data sink. Then the progress of the node, z, is no greater than x if there is no node located in the lens-shaped combined striped and grayed areas in Figure 5. Note that if no neighbouring node is found in the forward direction (i.e. $x > 0$), then the least backward neighbour from the data sink will be selected. To find the size of the striped and grayed areas, we need to derive angles θ and φ from geometry principles:

$$\theta = \cos^{-1}\left[\frac{R_{sink}^2 + R_{com}^2 - (R_{sink} - x)^2}{2R_{sink}R_{com}}\right] \tag{7}$$

and

$$\varphi = \cos^{-1}\left[\frac{R_{sink}^2 + (R_{sink} - x)^2 - R_{com}^2}{2R_{sink}(R_{sink} - x)}\right] \tag{8}$$

Then the striped and grayed areas can be, respectively represented as:

$$A_{striped} = (R_{sink} - x)^2\left(\varphi - \frac{\sin 2\varphi}{2}\right) \tag{9}$$

and

$$A_{grayed} = R_{com}^2\left(\theta - \frac{\sin 2\theta}{2}\right) \tag{10}$$

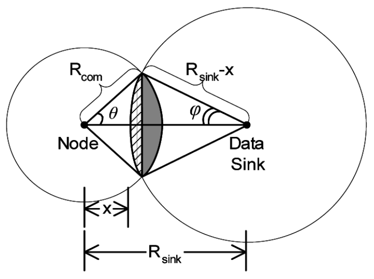
Let

$$A_x = A_{striped} + A_{grayed} \tag{11}$$

Then the probability that the progress of a node is no greater than x becomes:

$$\Pr(z \le x) = e^{-\lambda A_x} \tag{12}$$

Figure 5 Geometric relationships for calculating expected N_{hop} per node



It follows that the expected progress of a node, $E(z)$, is:

$$\begin{aligned} E(z) &= \int_{-R_{com}}^{R_{com}} x \Pr(x < z \le x + dx) \\ &= x e^{-\lambda A_x}\Big|_{-R_{com}}^{R_{com}} - \int_{-R_{com}}^{R_{com}} \Pr(z \le x)dx \\ &= R_{com}\left(1 + e^{-\lambda\pi R_{com}^2}\right) - \int_{-R_{com}}^{R_{com}} e^{-\lambda A_x} dx \end{aligned} \tag{13}$$

Once $E(z)$ is solved for one hop, then the expected hop count of a node R_{sink} away, $\overline{N_{hop}}$, can be computed by the algorithm EXPECTED_HOP_COUNT listed below. Note that the function *Progress* (R, R_{com}) returns the value $E(z)$ obtained by substituting the variable R in place of R_{sink} through Equations (7)–(13).

Programme EXPECTED_HOP_COUNT

```
1  hop_count ← 1
2  R ← R_sink
3  while (R > R_com) do
4          R ← [R – Progress(R, R_com)]
5          hop_count ← hop_count + 1
6  end while
7  N̄_hop ← hop_count
```

From Equations, (2), (3) and (5), the outcome of the hop count calculation is inherently tied to the timing performance of the three data reporting types, where a lower $\overline{N_{\text{hop}}}$ value translates into better data reporting times. Given R_{sink} and λ are fixed, such timing improvement can only be achieved by increasing R_{com}, which would lead to a list of problems associated with denser network connectivity as mentioned earlier. As a side note, throughout the above calculations it is assumed that the nodes are Poisson-distributed with density λ. In practical implementation, the actual node distribution on the sensing field may be highly variable, and the direct forwarding path may be affected the presence of physical obstacles such as mountains and lakes. Therefore, the derivations can be modified to take into account such effects, which will be relegated as part of our future work.

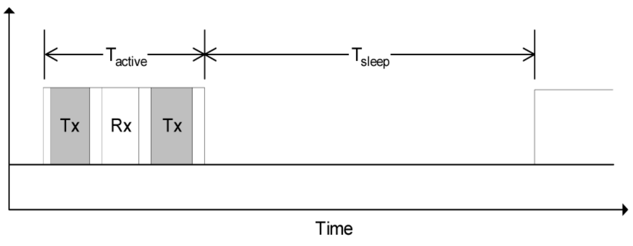
2.4 Sleep scheduling and active period dynamics

Increasing R_{com} places higher demands on energy consumption as transmission power changes exponentially in relationship to distance, but its effects are limited if the amount of data to be transmitted among nodes is low. A study by Reason and Rabaey (2004) had indicated that in practical implementation idle listening of the transceiver constitutes a significant part of energy consumption per node. Adjusting the amount of time allocated to each active period (i.e. T_{active}) may reduce the incidences of idle listening, though varying this parameter also affects data reporting efficacy as described in Section 2.2. Therefore, the energy consumption profile of each sleep cycle should be studied more closely to find the optimal balance among the various design considerations.

In addition to impacting on the energy consumption profile and data reporting timing requirements for sensor nodes, the parameter T_{active} also affects how sensing and time synchronisation are conducted. For instance, a high T_{sense} (i.e. the time it takes for the sensing unit to obtain data readings is long) prevents T_{active} to be set too low, which means both the processor and transceiver would have to endure long idle periods and expend additional energy. A simple remedy is to let the node first enter sleep state S_1 as listed in Table 1 and then move on to higher sleep states after data sensing has been completed. On the other hand, maintaining a low T_{active} places considerable pressure on the clocking and time synchronisation mechanism to provide accurate timing information. The choice of T_{active} needs to reflect the difficulty in to obtaining microsecond-level accuracy with currently available time synchronisation protocols for WSNs and the necessity of relying on inexpensive clock crystals in sensor nodes (Sundararaman et al., 2005).

With regards to communications, it is assumed that the node uses a single-channel simplex transceiver that does not allow transmitting and receiving data simultaneously over multiple channels. Figure 6 shows an example of typical activities undertaken by the transceiver over a sleep cycle which includes transmitting data (*T*x), receiving data (*R*x), idle listening and sleeping. The amount of time spent in data transmission and reception is directly related to packet sizes, bit rate, protocol signaling, coding effectiveness and general wireless channel conditions. In turn, such design choices would affect the measure of T_{hop}, which is desired to be as small as possible to minimise the timing requirements for data reporting. The first step in reducing T_{hop} is to decrease the amount of raw data to be transmitted, which involve compressing as much data into as few bits as possible, streamlining control messages, and simplifying network maintenance procedures. With raw data ready to be sent, increasing transmission bandwidth would certainly be beneficial in reducing T_{hop}, though bear in mind the possible increases in energy consumption and component costs involved.

Figure 6 Example of transceiver activity in sleep scheduling



Additional measures such as channel coding, CSMA/CA, packet acknowledgement and retransmission and spread spectrum techniques can be implemented to mitigate the effects of channel corruption and packet collisions. During network configuration, links with bad SNR characteristics can be rejected to reduce the amount of data corruption. However, in certain applications the above protective mechanisms can be deemed extraneous. For example, if the data traffic is consisted of small packets sent at infrequent times, then the benefits of using simple ALOHA for MAC outweigh those of protocol features such as RTS/CTS and end-to-end packet acknowledgement. A full evaluation on the methods in optimising T_{hop} is beyond the scope of this paper, though the reader is suggested to refer to Ha et al. (accepted-1) where an implicit acknowledgement mechanism had been proposed to expedite packet delivery by exploiting the multihop routing characteristics in WSNs to reduce control messaging overhead.

Summarising the different sources of energy expenditure during a sleep cycle, the average power consumption in communications for a node (i.e. total energy consumption over a sleep cycle), P_{com}, can then be represented by:

$$\begin{aligned} P_{\text{com}} &= \frac{\rho}{T_{\text{active}}}\left[\frac{D_{T\text{x}}}{B_{T\text{x}}}P_{T\text{x}} + \frac{D_{R\text{x}}}{B_{R\text{x}}}P_{R\text{x}}\right. \\ &\quad + \left(T_{\text{active}} - \frac{D_{T\text{x}}}{B_{T\text{x}}} - \frac{D_{R\text{x}}}{B_{R\text{x}}}\right)P_{\text{com\_idle}} \\ &\quad \left. + \frac{(1-\rho)}{\rho}T_{\text{active}}P_{\text{com\_sleep}}\right] \end{aligned} \tag{14}$$

where $D_{T\text{x}}$ and $D_{R\text{x}}$ refer to the expected amount of bits sent and received during an active period, respectively; $B_{T\text{x}}$ and $B_{R\text{x}}$ represent the average data rates at which data is sent and received during an active period, respectively; and P_{Tx}, P_{Rx}, $P_{\text{com\_idle}}$ and $P_{\text{com\_sleep}}$ stand for the amount of power consumed for sending data, receiving data, idle listening and sleep, respectively. Note that D_{Tx} and D_{Rx} account for both fresh and retransmitted data, as well as all necessary signaling packets and other control packets. Also, T_{active} is assumed to be set higher than the sum of the expected times needed for packet transmission and reception.

Suppose the node is supplied with an initial battery supply of E_{battery}, then the expected lifetime of the node, $\overline{T_{\text{lifetime}}}$, can be approximated by:

$$\overline{T_{\text{lifetime}}} = \frac{E_{\text{battery}}}{P_{\text{com}} + P_{\text{proc}} + P_{\text{sense}} + P_{\text{mem}}} \tag{15}$$

where P_{proc}, P_{sense} and P_{mem} represent power consumption for the processor, sensing unit and memory, respectively. Again, it is assumed that even with sleep scheduling the transceiver will still reign as the principal energy consuming unit in each node. For typical WSN applications, it is safe to treat P_{sense} and P_{mem} as negligible values because of their brevity and infrequency in operation. On the other hand, assuming the processor follows the same sleep schedule as the transceiver and remains active throughout each active period, then P_{proc} becomes:

$$\begin{aligned} P_{\text{proc}} &= \frac{\rho}{T_{\text{active}}}\left[T_{\text{active}}P_{\text{proc\_active}}\right. \\ &\qquad \left. + \frac{(1-\rho)}{\rho}T_{\text{active}}P_{\text{proc\_sleep}}\right] \end{aligned} \tag{16}$$

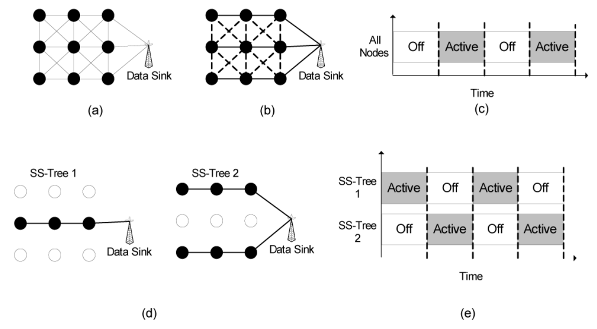
where $P_{\text{proc\_active}}$ and $P_{\text{proc\_sleep}}$ denote power consumed by the processor during active and sleep states, respectively.

3 The SS-Tree concept

Thus far, satisfying data reporting timing requirements and achieving energy efficiency through sleep scheduling have been discussed in detail. The remaining areas to be investigated are how to actually implement the sleep schedules and how the multihop WSN can be networked for maximum energy efficiency and data reporting effectiveness. For sleep schedules with ultra-low duty cycles, allowing nodes to wake up and sleep at random times is not recommended because of the possible excessive sleep slot synchronisation and beacon acquisition time between neighbouring nodes. Despite the need for setting up a common sleep schedule either globally or regionally, real-time changes to the application requirements, especially those that are pertinent to data reporting timing in response to environmental conditions in the sensing field, call for the ability to dynamically recalibrate sleep schedules without incurring considerable convergence times and messaging overhead. Implementing a global sleep schedule should not be very complex given a spanning tree structure is formed to connect all the nodes involved.

Figure 7(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 Figure 7(b) and (c), respectively. Then during the active period, considerable amounts of overhearing and packet collisions, represented as the dashed lines in Figure 7(b), would occur amongst neighbouring nodes. In contrast, none of the nodes will be capable of communicating during the sleep period, which renders 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.

Figure 7 SS-Tree concept (a) WSN topology, (b) logical spanning tree overlay, (c) global sleep schedule, (d) SS-Tree configuration and (e) interleaved sleep schedules



Suppose that the 9 nodes are divided into 2 groups of 3 and 6, respectively, in the manner shown in Figure 7(d), where each group follows its own sleep schedule such that the active periods of each group alternate, as illustrated in Figure 7(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 neighbours 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 SS-Tree or for short.

Besides achieving energy savings from simplifying the WSN topology, notice that in Figure 7(d) the sleeping nodes, coloured as white, are strategically located beside at least 1 branch of the other SS-Tree that is effectively a Connected Dominating Set (CDS). If each sleeping node in sleep state S_0 wakes up and enters sleep state S_1 whenever a neighbouring SS-Tree becomes active, 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 Figure 7(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 temporal sensing coverage using about the same amount of energy without making any substantial changes to the WSN. In general, the improvement attained in $\overline{T_{\text{event}}}$ if each node is adjacent to (N_{sst}–1) SS-Trees with evenly interleaved sleep schedules is:

$$\overline{T_{\text{event}}} = \begin{cases} \dfrac{T_{\text{active}}}{2\rho N_{\text{sst}}} + N_{\text{hop}} T_{\text{hop}} & \text{for } N_{\text{hop}} \le \alpha \\ \dfrac{T_{\text{active}}}{2\rho N_{\text{sst}}} + \left\lfloor \dfrac{N_{\text{hop}} - \alpha}{\beta} \right\rfloor \dfrac{T_{\text{active}}}{\rho} & \text{otherwise} \\ -T_{\text{sense}} + \left[\left(N_{\text{hop}} - \alpha \right) \bmod \beta \right] T_{\text{hop}} & \end{cases} \tag{17}$$

Despite this advantage in increasing sensing reliability for SS-Trees, one minor drawback is that both timer-driven and request-driven data cannot be simultaneously gathered from all SS-Trees each follows its own sleep schedule and only one SS-Tree may be active at any given time. For event-driven surveillance applications though, the impact of having unsynchronised periodic or request-driven 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 and request-driven data reporting, event-driven data is 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.

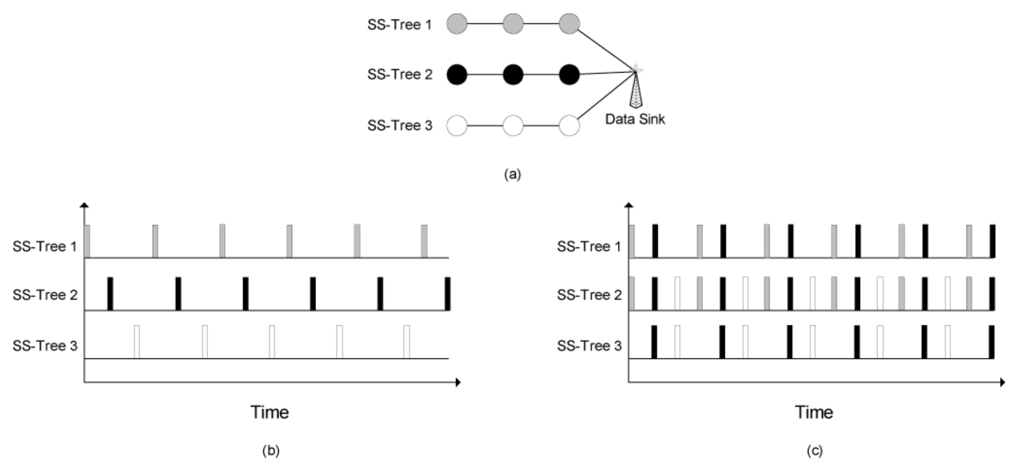
Another issue with SS-Tree is their potential inability to provide full spatial and temporal coverage at any given time, due to uneven network connectivity across the WSN such that at least one SS-Tree cannot form a true CDS. Figure 8(a) shows the same basic WSN topology as that used in Figure 7, but the 9 nodes are now arranged into three separate SS-Trees operating under the same low communication duty cycle with the sleep schedules arranged in a staggered manner, as shown in Figure 8(b). Because of sparse network connectivity, only 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 Figure 8(c). Allowing dynamic adjustment of transmission ranges would certainly be helpful in reducing the number of partial CDS, though it is more important to determine how many SS-Trees can be formed for a given R_{com} in the first place.

The key issue in realising the SS-Tree concept is the determination of how the sensor nodes can assigned to a certain number of SS-Trees on a given WSN topology. If transceivers with fixed transmission ranges are used, then the number of SS-Trees computable, N_{sst}, has to be inferred from the parameters R_{com} and λ since the full network topology is not known during predeployment phase. The fact that a node is adjacent to (N_{sst}–1) SS-Trees is equivalent to saying this node has N_{sst} different paths to the data sink. Assuming all such paths are disjoint, then N_{sst} for a single node can be approximated by the number of neighbours residing in the forwarding region as illustrated in Figure 4. Nevertheless, 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. To minimise the likelihood of producing shared nodes, the estimate of N_{sst} for the entire WSN should be made as conservative as possible. Therefore by setting $x = 0$ in Equations (7)–(11) and taking the minimum overall result, a conservative estimate of N_{sst} is given by:

$$\mathrm{N}_{\text{sst}} = \min_{R_{\text{com}} < R \le \sqrt{L^2 + W^2}/2} \left\lfloor \lambda \left[R^2 \left(\varphi' - \frac{\sin 2\varphi'}{2} \right) + {R_{\text{com}}}^2 \left(\theta' - \frac{\sin 2\theta'}{2} \right) \right] \right\rfloor \tag{18}$$

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, letting each node become aware of its neighbours' sleep schedules after Step 5 can ensure robustness against future SS-Tree breakages from upstream nodes. To further ensure

Figure 8 Impact of SS-Tree on spatial and temporal coverage: (a) SS-Tree assignment, (b) low duty cycle sleep scheduling and (c) event reporting windows



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 neighbourhood sleep schedules and reconnect to the data sink via the next neighbouring 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. More implementation details on SS-Tree computation and maintenance can be found in Ha et al. (2005), Ha et al. (accepted-1) and Ha et al. (accepted-2).

4 Numerical analysis

4.1 Sleep scheduling and temporal sensing coverage

This section investigates the relationship between sleep scheduling and temporal sensing coverage, and Table 2 summarises the list of system parameters involved. The first three parameters, namely $\overline{T_{\text{event}}}, \overline{T_{\text{timer}}}$ and $\overline{T_{\text{req}}}$, serve as performance benchmarks against parameters T_{active}, and N_{hop}. All timing information will be treated as relative to T_{hop} and T_{sense}, and both of them will assume the value of 1 in all the test cases. MATLAB is used to solve the various equations to obtain the results listed below.

Table 2 Summary of system parameters involved in analysing the relationship of sleep scheduling and temporal sensing coverage with SS-Trees

Symbol	*Description*
$\overline{T_{\text{event}}}$	Expected event-driven data reporting delay
$\overline{T_{\text{timer}}}$	Expected timer-driven data reporting delay
$\overline{T_{\text{req}}}$	Expected request-driven data reporting delay
T_{hop}	Per hop delivery time
T_{semse}	Data sensing and processing time
T_{active}	Length of active period
ρ	Duty cycle
N_{hop}	Hop count
N_{sst}	Number of SS-Trees

Figures 9 and 10 plot $\overline{T_{\text{event}}}$ versus T_{active} against different values of ρ and N_{hop}, respectively, whose relationship is defined by Equation (2) earlier. As shown in Figure 9, event-driven data reporting performs poorly respect to timing for low values of ρ as expected, where in general $\overline{T_{\text{event}}}$ is inversely proportional to ρ. Notice that for all any ρ, a low T_{active} to T_{hop} ratio (i.e. a packet can only traverse a few hops per active period) produces a much longer $\overline{T_{\text{event}}}$ than a ratio that permits more hop traversals within T_{active}. It implies that the popular MAC strategy of shutting off the transceiver right after transmitting one packet may not fare well with ultra-low duty cycle sleep scheduling in this regard. However, as this ratio increases further, the improvement in $\overline{T_{\text{event}}}$ eventually reaches its maximum, which is approximately at $T_{\text{active}} = T_{\text{hop}} N_{\text{hop}} / 2$ as indicated in Figure 10, before increasing linearly in proportion to T_{active}.

Figure 9 $\overline{T_{\text{event}}}$ versus T_{active} for various values of ρ (rho) at $N_{\text{hop}} = 10$, $T_{\text{hop}} = 1$ and $T_{\text{sense}} = 1$

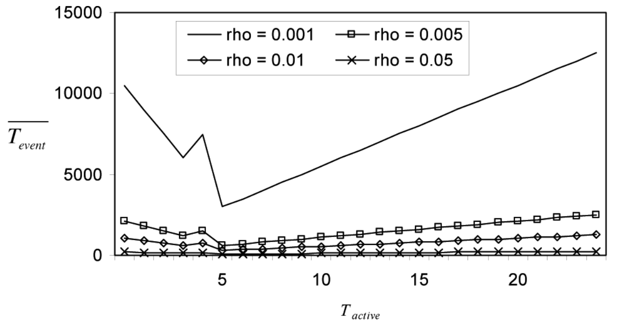
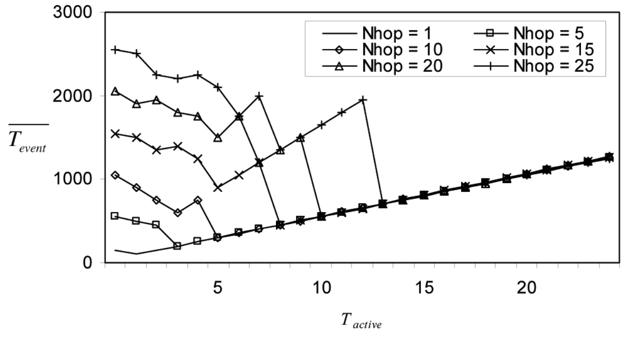


Figure 10 $\overline{T_{\text{event}}}$ versus T_{active} for various values of $N_{\text{hop}}(N_{\text{hop}})$ at $\rho = 0.01$, $T_{\text{hop}} = 1$ and $T_{\text{sense}} = 1$



The overall saw-tooth profile produced by smaller values of T_{active} in Figure 10 is due to the combined effects of data reporting packet being able to complete more hops in a single active period and increase in duration of each sleep cycle as T_{active} gets higher. Interestingly, although lowering N_{hop} generally leads to a proportional decrease in $\overline{T_{\text{event}}}$, the absolute timing improvement of shorter hop counts is significantly reduced when T_{active} is set such that the packet to be delivered end-to-end in one active period, which leaves the duration of the sleep action as the sole major factor contributing to event-driven data reporting latency.

Figures 11 and 12 graphically illustrate Equation (3), which relates $\overline{T_{\text{timer}}}$ and Tactive for different values of ρ and Nhop, respectively. Because of the similarities of the event-driven and timer-driven models, their results very much resemble each other, though at high Tactive values the effects of sleep cycle duration do not play any role in determining $\overline{T_{\text{timer}}}$. From both figures, it is obvious that the end-to-end completion time for timer-driven data reporting is very small if a high Tactive allows the packets to travel end-to-end in a single active period. However, high Tactive values would increase the minimum timing separation between consecutive timer-driven data reports, especially for low ρ values as shown in Figure 13 (see Equation (4)).

Figure 11 $\overline{T_{\text{timer}}}$ versus T_{active} for various values of ρ (rho) at $N_{\text{hop}} = 10$, $T_{\text{hop}} = 1$ and $T_{\text{sense}} = 1$

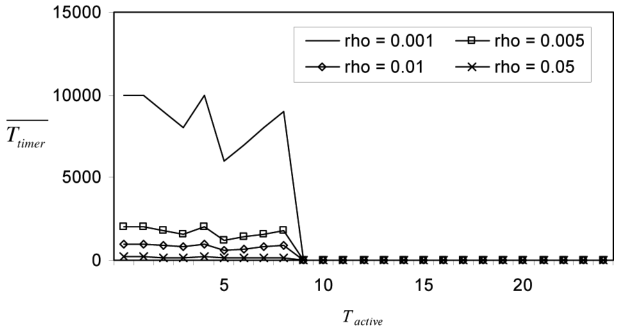


Figure 12 $\overline{T_{\text{timer}}}$ versus T_{active} for various values of N_{hop} (N_{hop}) at $\rho = 0.01$, $T_{\text{hop}} = 1$ and $T_{\text{sense}} = 1$

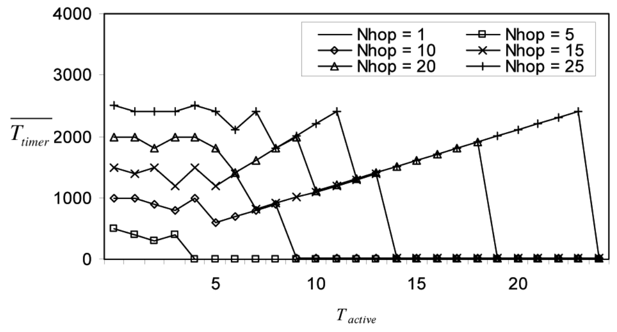
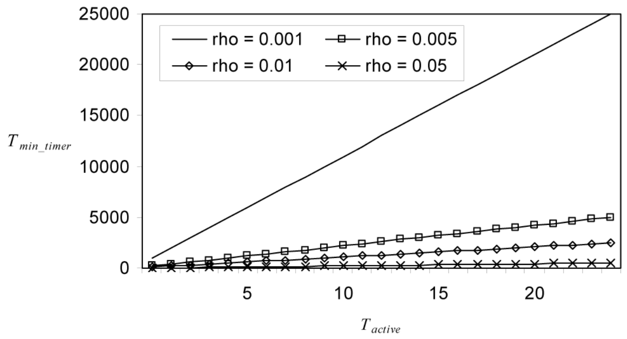


Figure 13 $T_{\text{min\_timer}}$ versus T_{active} for various values of ρ (rho)



The results for $\overline{T_{\text{req}}}$ versus T_{active} against different values of ρ and N_{hop} are shown in Figures 14 and 15, respectively, which in turn are given by Equation (5). Compared with the other two types of data reporting, request-driven data reporting requires at least twice the amount of time to complete, which can be exacerbated with a lower duty cycle and longer end-to-end hop count. As with the other two types, a higher T_{active} value that allows end-to-end data requests to be completed in a single active period produces a smaller $\overline{T_{\text{req}}}$. However, as T_{active} increases lower N_{hop} values retain their timing advantage of $\overline{T_{\text{req}}}$ better in request-driven data reporting than the other types.

Figure 14 $\overline{T_{\text{req}}}$ versus T_{active} for various values of ρ (rho) at $N_{\text{hop}} = 10$, $T_{\text{hop}} = 1$ and $T_{\text{sense}} = 1$

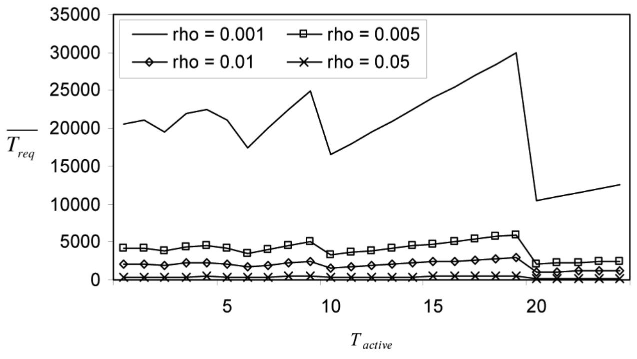
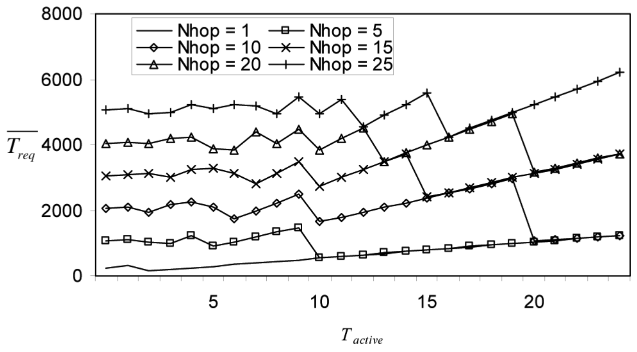


Figure 15 $\overline{T_{\text{req}}}$ versus T_{active} for various values of N_{hop} (N_{hop}) at $\rho = 0.01$, $T_{\text{hop}} = 1$ and $T_{\text{sense}} = 1$



4.2 *Transmission range assignment, temporal sensing coverage and impact of SS-Trees*

In all three data reporting types, a smaller hop count generally helps in reducing end-to-end delay, albeit to various degrees when coupled with the effects of ρ and T_{active}. As hop count is related to transmission range assignment and node deployment density, the interaction of these three aspects remains to be investigated. Also with the introduction of the SS-Tree concept in Section 3, it would be interesting to examine the timing performance of event-driven data reporting under different network configuration settings. Table 3 lists the main parameters involved in this analysis, where the values for λ, R_{sink} and R_{com} will all be taken as relative. Note that the values for R_{sink} are restricted by the dimensions of the sensing field (i.e. $L \times W$), though the exact area attributes will not be specified in the test cases. The various formulations presented in Section 3 as well as those for evaluating $\overline{N_{\text{hop}}}$ in Section 2.3 are solved using MATLAB, where the latter results are computed with the help of numerical integration techniques.

Firstly, the relationship between node deployment density and expected hop count with respect to the sink-to-node physical distance and communication range is examined. Results shown by Figure 16 suggest that to reduce $\overline{N_{\text{hop}}}$ by half, which in turn can lead to a potential 50% timing improvement for all types of data reporting, node deployment density has to be increased by at least three times if R_{com} is to remain the same. On the other hand, achieving a similar decrease in $\overline{N_{\text{hop}}}$ only requires increasing the transmission range by less than twice the original value as shown in Figure 17. So this demonstrates that improving temporal sensing coverage involves trading off deployment cost or per node energy use, and the ultimate choice depends on the impact of both options in the overall WSN design. Further discussions on energy efficiency with respect to transmission range will be provided in Section 4.3.

Table 3 Summary of system parameters involved in analysing the relationship between event-driven data reporting, transmission range assignment and SS-Trees

Symbol	*Description*
N_{sst}	Number of SS-Trees
$\overline{T_{\text{event}}}$	Expected event-driven data reporting delay
$\overline{N_{\text{hop}}}$	Expected hop count
N_{hop}	Hop count
λ	Sensor node deployment density
$L \times W$	Sensing field dimensions
R_{sink}	Physical distance between node and data sink
R_{com}	Communication range

Figure 16 $\overline{N_{\text{hop}}}$ versus R_{sink} for various values of λ (lambda) at $R_{\text{com}} = 1$

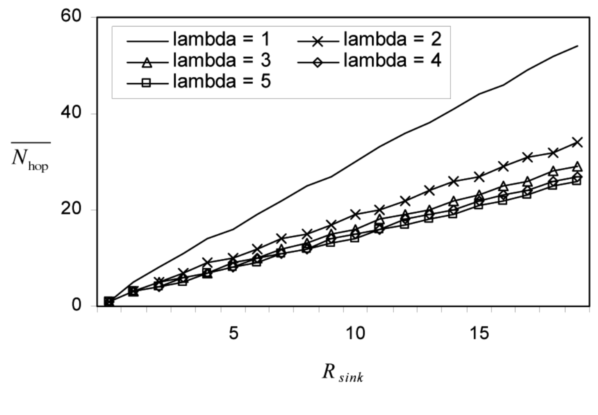
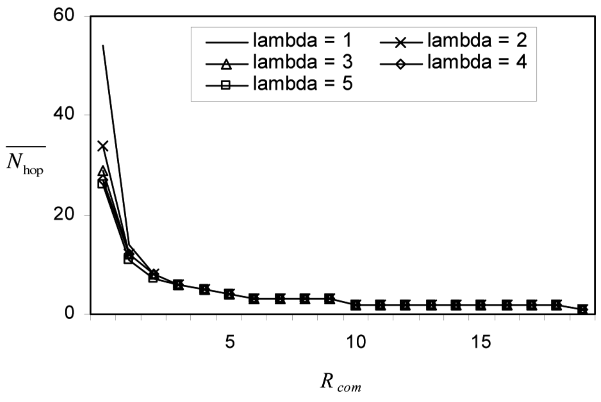


Figure 17 $\overline{N_{\text{hop}}}$ versus R_{com} for various values of λ (lambda) at $R_{\text{sink}} = 20$



If SS-Trees are to be implemented, then increasing node deployment density is projected to allow more SS-Trees to be computed, as shown in Figure 18. However, such increase in N_{sst} pales in comparison with those attained through increasing transmission range, as shown in Figure 19. While extending R_{com} to attain an N_{sst} estimate that reaches several hundred or even several thousand is a bit extreme, numerical results suggest that a slight increase in R_{com} would be enough to produce tens of SS-Trees.

Figure 18 N_{sst} versus R_{sink} for various values of λ (lambda) at $R_{\text{com}} = 1$

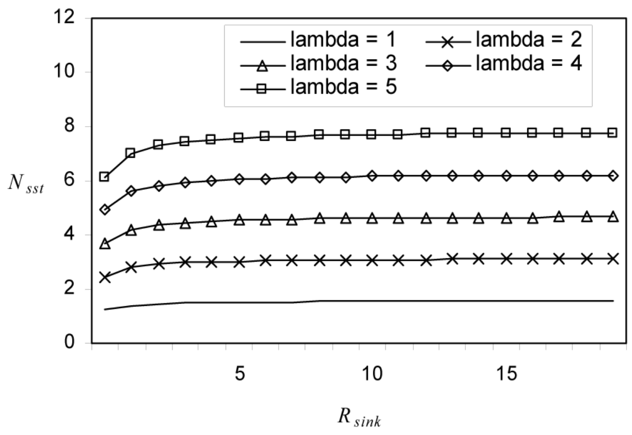
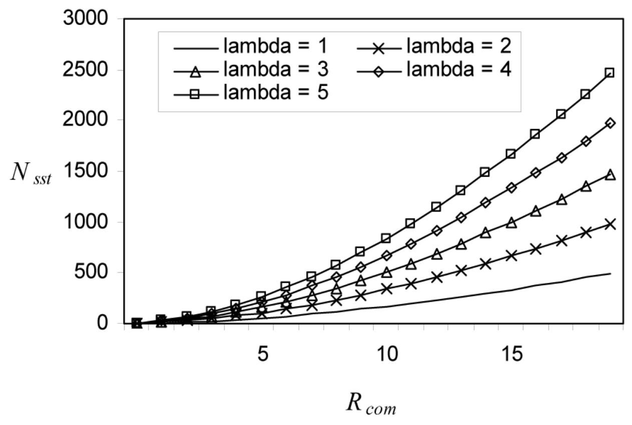


Figure 19 N_{sst} versus R_{com} for various values of λ (lambda) at $R_{\text{sink}} = 20$



Increasing the number of computable SS-Trees will in turn improve the timing performance of event-driven data reporting, which is essential in the majority of WSN applications. Figure 20 examines the effects of implementing more SS-Trees on $\overline{T_{\text{event}}}$ with respect to T_{active}. It is shown that the timing improvement in $\overline{T_{\text{event}}}$ as offered by having more SS-Trees becomes more pronounced when T_{active} is high. Also, the linear increasing profile of $\overline{T_{\text{event}}}$ after it reaches its minimum becomes more and more subtle as N_{sst} increases. Figure 21 looks at how $\overline{T_{\text{event}}}$ is affected by both changes in N_{sst} and ρ and discovers that for a given ρ, a high N_{sst} count dramatically decreases $\overline{T_{\text{event}}}$. In other words, by having more SS-Trees, the operational duty cycle required for guaranteeing a certain $\overline{T_{\text{event}}}$ threshold as required by the sensing application is effectively reduced, thereby significantly improving energy efficiency.

Figure 20 $\overline{T_{\text{event}}}$ versus T_{active} for various values of N_{sst} at $\rho = 0.01$, $N_{\text{hop}} = 10$, $T_{\text{hop}} = 1$ and $T_{\text{sense}} = 1$

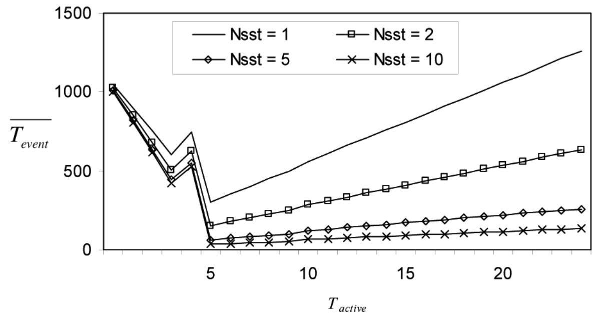
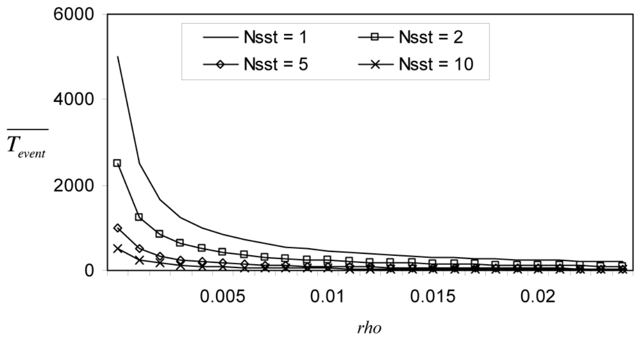


Figure 21 $\overline{T_{\text{event}}}$ versus ρ for various values of N_{sst} at $T_{\text{active}} = 10$, $N_{\text{hop}} = 10$, $T_{\text{hop}} = 1$ and $T_{\text{sense}} = 1$



4.3 Energy efficiency and temporal sensing coverage – a case study

Sections 4.1 and 4.2 have discussed in detail how temporal sensing coverage is affected by sleep scheduling, transmission range assignment and the SS-Tree concept. This section will tie the other major consideration in WSN design, which is energy efficiency, into the overall discussion. Table 4 contains the many parameters that will figure into the calculations, and most of them will take on or derived from values drawn specifications of a selected number of commercially available products as listed in Table 5. Additional parameters used in the numerical analysis are given in Table 6. Again, the numerical results for this section are computed using MATLAB.

In Table 5, the first two are transceivers, namely XBee-PRO from Maxstream Inc. (http://www.maxstream.net) and Z-Link from Atmel Corporation (http://www.atmel.com/), which conform to the IEEE 802.15.4/Zigbee standard. In contrast, the Crossbow Technology (http://www.xbow.com/) MICAz is a so-called mote, which is a standalone sensor node with a Zigbee-enabled transceiver installed. For the system lifetime analysis, it is assumed that all three transceivers can be interchanged to work with the same processor on the MICAz for energy consumption measurement purposes. Notice that despite the exponential relationship between transmission power and range, the actual current draw for transmission increases at a much smaller scale compared to transmission range (see comparison between XBee-PRO and MICAz). It is perhaps due to the fact that even for short-range transceivers, a sizable portion of power is consumed for maintaining components such as amplifiers and mixers while only a fraction is transmitted out to the wireless channel. In terms of cost, XBee-PRO is the most expensive option whereas Z-Link is the most economical. The cost of the transceiver onboard the MICAz module is not given, though it is assumed to be somewhere in between the other 2 transceivers.

Table 4 Summary of system parameters involved in analysing energy efficiency with temporal sensing coverage, sleep scheduling and transmission range assignment

Symbol	*Description*
$\overline{T_{\text{lifetime}}}$	Expected WSN system lifetime
$\overline{T_{\text{event}}}$	Expected event-driven data reporting delay
$\overline{T_{\text{timer}}}$	Expected timer-driven data reporting delay
$\overline{T_{\text{req}}}$	Expected request-driven data reporting delay
$L \times W$	Sensing field dimensions
E_{battery}	Initial battery supply
R_{com}	Communication range
T_{active}	Length of active period
ρ	Duty cycle
D_{Tx}	Expected amount of bits transmitted during an active period
D_{Rx}	Expected amount of bits received during an active period
B_{Tx}	Average data rate for data transmission
B_{Rx}	Average data rate for data reception
P_{Tx}	Power consumed by transceiver during data transmission
P_{Rx}	Power consumed by transceiver during data reception
$P_{\text{com\_idle}}$	Power consumed by transceiver during idle listening
$P_{\text{com\_sleep}}$	Power consumed by transceiver during sleep state
$P_{\text{proc\_active}}$	Power consumed by processor during active state
$P_{\text{proc\_sleep}}$	Power consumed by processor during sleep state

Table 5 Specifications for commercially available transceivers and motes

Specification	*Maxstream XBee-PRO*	*Atmel Z-Link*	*Crossbow MICAz*
Processor current (active)	–	–	8 mA
Processor current (sleep)	–	–	15 µA
Band	2.4 GHz	915 MHz	2.4 GHz

Table 5 Specifications for commercially available transceivers and motes (continued)

Specification	*Maxstream XBee-PRO*	*Atmel Z-Link*	*Crossbow MICAz*
Data rate	250 kbps	40 kbps	250 kbps
*T*x power output	60 mW/18 dBm	16 mW/12 dBm	1 mW/0 dBm
Supply voltage	3.3 V	3.3 V	3.3 V
*T*x current	270 mA	60 mA	17.4 mA
Idle/*R*x current	55 mA	14.5 mA	19.7 mA
Sleep current	10 µA	1 µA	1 µA
*T*x range (indoor/urban)	upto 100 m	upto 30 m	upto 30 m
*T*x range (outdoor/LOS)	upto 1.6 km	upto 100 m	up to 100 m
Price	$32 USD/chip	$6.75 USD/chip	$125 USD/unit

Table 6 Additional parameters for numerical analysis

Parameter	*Outdoor*	*Urban*
$L \times W$	3 km × 3 km	500 m × 500 m
λ	1 per 10,000 m^2	1 per 500 m^2
E_{battery}	2000 mAh	2000 mAh
ρ	0.01	0.01
D_{Tx}	10 bytes	10 bytes
D_{Rx}	10 bytes	10 bytes
T_{sense}	100 ms	100 ms
T_{hop}	5 bytes/B_{tx}	5 bytes/B_{tx}

The numerical analysis is divided into two cases, outdoor and urban, both of which assume each transceiver is able to transmit at the corresponding range specified in Table 5. In any case, multichannel selection is disabled and each transceiver can only rely on one radio channel for transmission. Sensing field dimensions and node deployment densities are set different for both cases, though in either case the data sink is assumed to be located in the middle of the sensing field. Initially, each node is supplied with 2 AA batteries which contain about 2000 mAh of capacity, and energy used by the other components besides the processor and transceiver is ignored. Delivering each data reporting packet over one hop is assumed to involve the exchange of five bytes in total, including messaging overhead and retransmissions. On the other hand, for each node it is assumed about ten bytes of data are to be transmitted and received, respectively, during each active period.

Figures 22–24 plot $\overline{T_{\text{event}}}$, $\overline{T_{\text{timer}}}$ and $\overline{T_{\text{req}}}$, respectively, for different transceivers at different transmission ranges while inherently factoring in the effects of SS-Trees. Under the conditions specified in Table 6 regarding sensing area dimensions and node deployment density, the number of SS-Trees computable for shorter range transceivers (i.e. Z-Link and MICAz) is one for outdoor and two for urban, while it is a remarkable 314 and 24, respectively, for the XBee-PRO. As longer ranges also lead to lower $\overline{N_{\text{hop}}}$, it is no wonder that the timing performance for XBee-PRO in all three data reporting types outperforms the other two transceivers for all $T_{\text{active}}/T_{\text{hop}}$ ratios in both outdoor and urban environments. On the other hand, because of Z-Link's slower transmission speed, its timing performance mostly registers in the second-level range, which is tens to hundreds of times of that achieved by XBee-PRO and MICAz.

Figure 22 $\overline{T_{\text{event}}}$ versus $T_{\text{active}}/T_{\text{hop}}$ for different transceivers at different transmission ranges

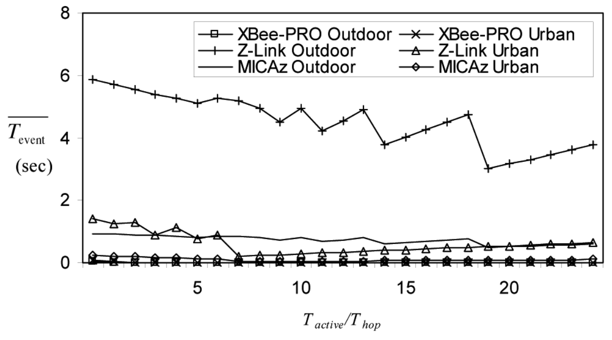


Figure 23 $\overline{T_{\text{timer}}}$ versus $T_{\text{active}}/T_{\text{hop}}$ for different transceivers at different transmission ranges

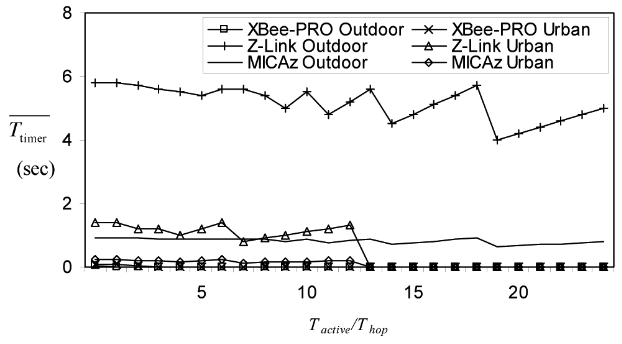
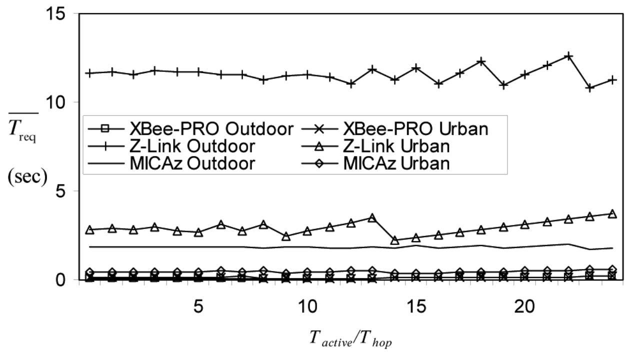
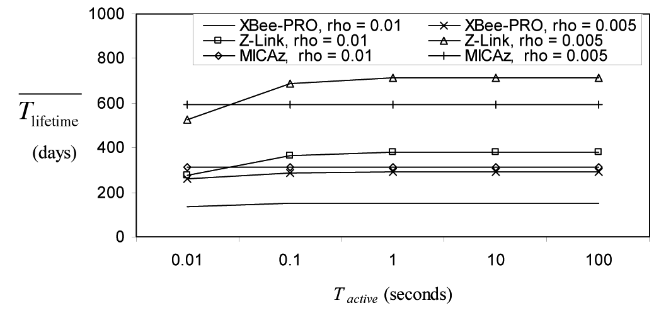


Figure 24 $\overline{T_{\text{req}}}$ versus $T_{\text{active}}/T_{\text{hop}}$ for different transceivers at different transmission ranges



Besides offering excellent timing performance, XBee-PRO also delivers solid energy efficiency results as demonstrated in Figure 25. Although for a given ρ, the expected operational lifetime for Z-Link and MICAz are about two times of that of XBee-PRO, the long-range chip can always operate at a lower duty cycle for better energy efficiency with only a minor degradation in data reporting timing performance. As a side note, the tailing off in $\overline{T}_{\text{lifetime}}$ as T_{active} decreases for Z-Link and to a lesser extent for XBee-PRO is due to the fact that energy consumption from packet transmission dominates when T_{active} is small. When the difference between transmission power and reception/idle power is as large as in the case for Z-Link and XBee-PRO, power consumption will actually increase as the active period duration gets smaller, thereby reducing expected system lifetime. However, such undesired decrease in $\overline{T}_{\text{lifetime}}$ can be more than made up by reducing the duty cycle, which underscores the fact that sleep scheduling is a bigger contributor to energy efficiency than transmission power regulation.

Figure 25 $\overline{T}_{\text{lifetime}}$ versus T_{active} for the three different transceivers under different duty cycles



5 Related concepts and research

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. 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 minimise the amount of energy lost due to needless transceiver idle listening (van Dam and Langendoen, 2003; Ye et al., 2004). 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 neighbours without prior notification, thereby forcing frequent packet reroutes. In addition, network maintenance functions such as neighbourhood discovery and time synchronisation 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 (Sichitiu, 2004).

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 (Ramanathan and Rosales-Hain, 2000; Xing et al., 2005). 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 characterised 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 (Aslam et al., 2003; Chang and Tassulas, 2004). This is to keep the best network connectivity and to increase the total transmission capacity. Other cost metrics, such as inter-node distances, transmission delays, channel conditions and route hop count, 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 organisational method for WSNs to minimise upstream and downstream communication costs is to interconnect 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 (Boukerche et al., 2003; Cetintemel et al., 2003). 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 utilising the tree structure for WSN formulation (Heinzelman et al., 2002; Manjeshwar et al., 2002), 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 (Du et al., 2003; Lindsey et al., 2002), 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 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 millisecond active time per 1 sec versus 1 sec active time per 1000 sec for a duty cycle of 0.1%). However, additional control overhead and hardware cost may be incurred in keeping time synchronisation tighter. 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 CDS, form a virtual backbone (Wan et al., 2004). In the graph theory terminology, a CDS in a graph is a set of connected vertices such that every vertex in the graph either is in the set or has a neighbour 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 minimising 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.

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 the application requirements. Many prior studies on the WSN connectivity, especially those stemmed from mobile ad hoc network (MANET) research such as Chen et al. (2002), have paid little attention on the importance of guaranteeing sensing coverage. As mentioned in Section 2.1, 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 (Hsin and Liu, 2004). 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 (Slijepcevic and Potkonjak, 2001), though network connectivity should also be considered to form a minimum connected sensor cover (Gupta et al., 2003). Therefore, both sensing coverage and network connectivity should jointly be considered when formulating an optimal sleep schedule that can balance all the application requirements (Xing et al., 2005; Zou and Chakrabarty, 2005).

7 Conclusions and future work

In this work, the relationship between energy efficiency, temporal sensing coverage and transmission range assignment is studied in detail. By summarising the results, it is found that the combination of a low duty cycle sleep schedule, long active period duration with respect to per hop delivery time, low traffic load, as well as a long transmission range, is able to best balance the various design considerations and achieve better performance in extending system lifetime and reducing data reporting latency. The SS-Tree concept also plays a significant role in improving timing performance in event-driven data reporting, which is essential in most of the WSN applications. However, these performance improvements may come at the expense of an increase in node manufacturing and deployment costs. Therefore, trade-offs in sensing application performance and affordability are worth serious deliberation during network planning and predeployment stage.

This paper covers quite a few unanswered questions regarding the SS-Tree concept regarding actual sleep scheduling and transmission range assignment. However, throughout this paper, it is assumed that the transmission range is already fixed at the predeployment phase, and spatial sensing coverage is adequately provided. Therefore, in the next stage of research, the following areas will be explored further:

- sleep scheduling with variable transmission range assignment
- sleep scheduling with spatial sensing coverage considerations
- more efficient SS-Tree computation methods and
- refined MAC design for efficient packet delivery.

Acknowledgement

The authors would like to thank Mr. Lawrence Wong for his insights.

References

Aslam, J., Li, Q. and Rus, D. (2003) 'Three power-aware routing algorithms for sensor networks', *Wireless Communications and Mobile Computing*, Vol. 2, No. 3, pp.187–208.

Boukerche, A., Cheng, X. and Linus, J. (2003) 'Energy-aware data-centric routing in microsensor networks', in *Sixth ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems*, San Diego, CA, pp.42–49.

Cetintemel, U., Flinders, A. and Sun, Y. (2003) 'Power-efficient data dissemination in wireless sensor networks', in *Third ACM International Workshop on Data Engineering for Wireless and Mobile Access*, San Diego, CA, pp.1–8.

Chang, J-H. and Tassulas, L. (2004) 'Maximum lifetime routing in wireless sensor networks', *IEEE/ACM Transactions on Networking*, Vol. 12, No. 4, pp.609–619.

Chen, B., Jamieson, K., Balakrishnan, H. and Morris, R. (2002) 'Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks', *Wireless Networks*, Vol. 8, pp.481–494.

Du, K., Wu, J. and Zhou, D. (2003) 'Chain-based protocols for data broadcasting and gathering in the sensor networks', in *International Parallel and Distributed Processing Symposium*, Nice, France, April.

Gupta, H., Das, S.R. and Gu, Q. (2003) 'Connected sensor cover: self-organization of sensor networks for efficient query execution', in *Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, June, Annapolis, Maryland, pp.189–200.

Ha, R.W., Ho, P.H. and Shen, X.S. (2005) 'A cross-layer organizational approach for mesh-based wide-area wireless sensor networks', *IEEE BroadNets*, October, Boston, MA, pp.885–894.

Ha, R.W., Ho, P.H. and Shen, X.S. (accepted-1) 'Cross-layer application-specific wireless sensor network design with single-channel CSMA MAC over sense-sleep trees', *Elsevier Journal: Computer Communications*.

Ha, R.W., Ho, P.H. and Shen, X.S. (accepted-2) 'Sleep scheduling for wireless sensor networks via network flow model', *Elsevier Journal: Computer Communications*.

Heinzelman, W.B., Chandrakasan, A.P. and Balakrishnan, H. (2002) 'An application-specific protocol architecture for wireless microsensor networks', *IEEE Transactions on Wireless Communications*, Vol. 1, No. 4, pp.660–669.

Hsin, C-F. and Liu, M. (2004) 'Network coverage using low duty-cycled sensors: random and coordinated sleep algorithms', *Third International Symposium on Information Processing in Sensor Networks*, April, Berkeley, CA, pp.433–442.

Lindsey, S., Raghavendra, C. and Sivalingam, K.M. (2002) 'Data gathering algorithms in sensor networks using energy metrics', *IEEE Transactions on Parallel and Distributed Systems*, Vol. 13, No. 9, pp.924–935.

Manjeshwar, A., Zeng, Q.A. and Agrawal, D. (2002) 'An analytical model for information retrieval in wireless sensor networks using enhanced APTEEN protocol', *IEEE Transactions on Parallel and Distributed Systems*, Vol. 13, No. 12, pp.1290–1302.

Ramanathan, R. and Rosales-Hain, R. (2000) 'Topology control of multihop wireless networks using transmit power adjustment', *Ninteenth IEEE INFOCOM, Tel-Aviv*, March, Israel, pp.404–413.

Reason, J.M. and Rabaey, J.M. (2004) 'A study of energy consumption and reliability in a multi-hop sensor network', *ACM Mobile Computing and Communications Review*, Vol. 8, No. 1, pp.84–97.

Sichitiu, M.L. (2004) 'Cross-layer scheduling for power efficiency in wireless sensor networks', *Twenty Third IEEE INFOCOM*, March, Hong Kong, China, pp.1740–1750.

Sinha, A. and Chandrakasan, A. (2001) 'Dynamic power management in wireless sensor networks', *IEEE Design and Test of Computers*, Vol. 18, No. 2, pp.62–74.

Slijepcevic, S. and Potkonjak, M. (2001) 'Power efficient organization of wireless sensor networks', *IEEE International Conference on Communications*, pp.472–476.

Sundararaman, B., Buy, U. and Kshemkalyani, A.D. (2005) 'Clock synchronization for wireless sensor networks: a survey', *Elsevier Ad Hoc Networks*, Vol. 3, No. 3, pp.281–323.

Takagi, H. and Kleinrock, L. (1984) 'Optimal transmission ranges for randomly distributed packet radio terminals', *IEEE Transactions on Communications*, Vol. 32, No. 3, pp.246–257.

van Dam, T. and Langendoen, K. (2003) 'An adaptive energy-efficient MAC protocol for wireless sensor networks', *ACM Conference on Networked Sensor Systems*, November, Los Angeles, CA, pp.171–180.

Wan, P-J., Alzoubi, K.M. and Frieder, O. (2004) 'Distributed construction of connected dominating set in wireless ad hoc networks', *Mobile Networks and Applications*, Vol. 9, No. 2, pp.141–149.

Xing, G., Lu, C., Zhang, Y., Huang, Q. and Pless, R. (2005) 'Minimum power configuration in wireless sensor networks', *Sixth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 25–28 May, Urbana-Champaign, IL, pp.390–401.

Xing, G., Wang, X., Zhang, Y., Lu, C., Pless, R. and Gill, C. (2005) 'Integrated coverage and connectivity configuration for energy conservation in sensor networks', *ACM Transactions on Sensor Networks*, Vol. 1, No. 1, pp.36–72.

Ye, W., Heidemann, J. and Estrin, D. (2004) 'Medium access control with coordinated adaptive sleeping for wireless sensor networks', *IEEE/ACM Transactions on Networking*, Vol. 12, No. 3, pp.493–506.

Zou, Y. and Chakrabarty, K. (2005) 'A distributed coverage- and connectivity-centric technique for selecting active nodes in wireless sensor networks', *IEEE Transactions on Computers*, Vol. 54, No. 8, pp.978–991.