ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

SLEEP SCHEDULING FOR CRITICAL EVENT MONITORING IN WIRELESS SENSOR NETWORKS

S.Kavitha[1], S.Lalitha[2]

S.Kavitha[1], PG Scholar-M.E(CSE), Gnanamani College of Technology, Nammakal, T.N, India[1]

S.Lalitha[2], AP/CSE, Gnanamani College of Technology, Nammakal, T.N, India[2]

Abstract: Over the past decade, local monitoring has been shown to be a powerful technique for improving security in multihop wireless sensor networks (WSNs). However, local monitoring as it is currently practiced is costly in terms of energy consumption. Sleep-wake protocols are critical in sensor networks to ensure long-lived operation. However, an open problem is how to develop efficient mechanisms that can be incorporated with sleep-wake protocols to ensure both long lived operation and a high degree of security. To overcome this problem by using local monitoring, each node oversees part of the traffic going in and out of its neighbors to determine if the behavior is suspicious, such as, unusually long delay in forwarding a packet. Here, a protocol is used to make local monitoring parsimonious in its energy consumption and to integrate it with any extant sleep-wake protocol in the network. Analytically proved that the security coverage is not weakened by the protocol. Then perform simulations in GLOMOSIM to demonstrate that the performance of local monitoring is practically unchanged.

Key words: Energy efficiency, logical monitoring, critical events, sleep scheduling, sensor networks.

I. INTRODUCTION

Wireless Sensor Networks (WSN) is a collection of spatially deployed wireless sensors by which to monitor various changes of environmental conditions (e.g. forest fire, air pollutant concentration and object moving) in a collaborative manner without relying on any underlying infrastructure support. Recently, a number of research efforts have been made to develop sensor hardware and network architecture in order to effectively deploy WSNs for a variety of applications. Many network parameters such as sensing range, transmission range and node density have to be carefully considered at the network design stage, according to specific applications. To achieve this, it is critical to capture the impacts of network parameters on network performance with respect to application specifications. Since a distributed network has multiple nodes and services many messages and each node is a shared resource, many decisions must be made. There may be multiple paths from the source to destination. Therefore, message routing is an important topic. The main performance measures affected by the routing scheme are throughput (quality of service) and average packet delay (quantity of service).

II. DESIGN OVERVIEW

A. System Model And Assumptions

Consider a homogeneous, static sensor network, in which sensor nodes work in a duty cycling mode. In such a toggling period (TP), a node keeps active for TP*DC where DC is the duty cycle. Although the active period of neighbor nodes may be different, the communication among them can be guaranteed based on a MAC protocol.

In the active state, a node may detect targets within its sensing radius r, and communicate with other nodes within its communication radius R. assume that every node is aware of its own location and is able to determine a target's position at detection. In addition, assume that the sensor nodes are locally time synchronized using a protocol. In fact, as long as the distance between to target is more than two times of the communication radius of nodes, the sleep scheduling actions triggered by them will not overlap [1].

Components of sleep scheduling protocol:

- *Target prediction.* The proposed target prediction scheme consists of three steps: current state calculation, kinematics- based prediction and probability based prediction. After calculating the current state, the kinematics based prediction step calculates the expected displacement from the current location within the next sleep delay, and the probability models for scalar displacement and the derivation.
- *Awakened node reduction.* The number of awakened nodes is reduced with two efforts: controlling the scope of awakened regions, and choose a subset of nodes in an awakened region.
- *Active time control.* Based on the probabilistic models that are established with target prediction, schedules

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

an awakened node to be active, so that the probability that it detects to target is close to 1.

B. Traffic Model

Fig. 1 depicts the kinds of communication paths in the network, namely,

- *Forward direction* (downlink): The base station sends a message to one of nodes in the network.
- *Backward direction* (uplink): A regular node sends a message to the base station.

Several sensor nodes today, are often equipped with passive event detection capabilities that allow a node to detect an event even while it is in sleep mode. Still others provide ultra low-power, low-rate periodic sampling mechanisms for rare event detection. Upon the detection of an event, the sensor node is immediately woke up (within several µsec) and is ready to transmit a notification message to the base-station. Similarly, the base-station is often required to transmit imperative commands or queries to sensor nodes that may originate asynchronously. Messages in either direction, thus, originate at random times (asynchronously) and this implies that messages may potentially originate at an inopportune time when all other nodes in the network are in sleep mode and not ready to receive the message [5]. While these messages occur infrequently, they reflect urgency; as such their delivery demands non-negotiable worst case delay bounds. In this paper, delay is defined as the time duration between generations of a message at a node (base-station or a regular node) until its eventual delivery at the destination node.

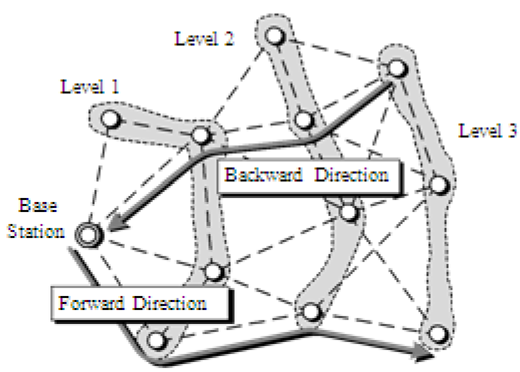

Fig. 1 Networks and Traffic Model

C. Critical Event Monitoring

Whenever a critical event occurs, the critical event is detected by the nearby sensor nodes. Immediately these sensor nodes should broadcast an alarm message to the entire network. Sleep scheduling is used to reduce the energy consumption which is leads to increase network lifetime. But it leads to the broadcasting delay, especially in large scale WSNs. So we need to balance both energy efficiency and delay aware. In this paper, a delay optimized sleep scheduling method is proposed to reduce the delay of alarm broadcasting in WSNs. When a critical event occurs, an alarm is immediately transmitted to a center node, and then it is quickly broadcasted by the center node to the entire network without any collision.

In sleep scheduling, sender nodes should wait until receiver nodes are active and ready to receive the message. Sleep scheduling should increase the network life time but it could cause transmission delay. Whenever the network scale increases, the broadcasting delays also increase. So a delay aware sleep scheduling method needs to be designed to provide low broadcasting delay from any node in the WSN. Most of sleep scheduling methods is introduced to minimize the energy consumption. To minimize the broadcasting delay in WSN, it is needed to minimize the time wasted for waiting during the broadcasting. The destination node is wake up immediately when the source nodes obtain the broadcasting packets. Here, the broadcasting delay is reduced. Sleep scheduling in wireless sensor network is shown in fig 2.2. Whenever a critical event occurs, it is detected by the nearby sensor nodes and immediately it should sent to its neighbor nodes [2].

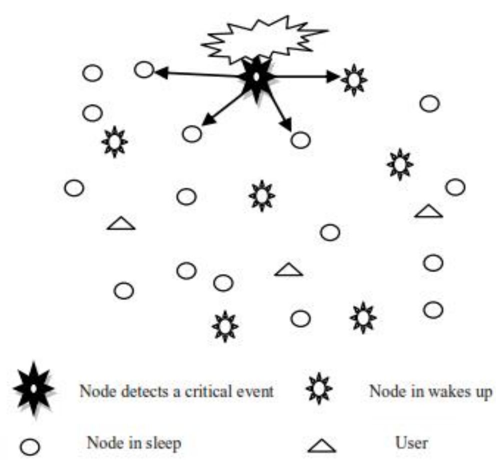

Fig. 2 Critical Event Monitoring with a WSN

III. METHODOLOGIES

A. Energy Efficient Local Monitoring In Sensor Networks

For a target tracking sensor network, the sensor coverage requirements at the surveillance state and at the tracking state. The surveillance state requires a homogeneous coverage on a wide area, but does not need full coverage. A wireless sensor network (WSN) is a set of sensor nodes deployed in a given space. The nodes form a network with the wireless links, and are capable of collecting, relaying and processing the sensor networks present a variety of network operation models for data delivery and processing. Three models are identified, namely, the continuous, event-driven and user initiated.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

- *Neighbor Discovery*. The neighbor discovery component at each individual node manages both the broadcasting of the working schedule and the maintenance of the neighbor table. To announce the existence of a node, the neighbor discovery component broadcasts the node ID and working schedule information with a configurable parameter that decides the number of retransmissions. While receiving a broadcasted working schedule announcement, the neighbor discovery component checks whether the source of the packet has been in its neighbor table. If the source does not exist in the neighbor table, the neighbor discovery component appends the source and corresponding working neighbor discovery component would just ignore the received broadcasting packet and does nothing. In short, the neighbor discovery component attempts to keep track of the schedules of all neighbors.
- *Link Quality Measurement.* To measure the pairwise link quality between a node and its neighbors, the link quality measurement component at each of its neighbors and utilizes the link layer acknowledgement from B-MAC calculate the pairwise link quality. Depending on the desired accuracy of measurement, the link quality measurement component provides a configurable parameter to set the number of message transmission between pairs of nodes. In order to minimize the impact of interference, serialize the link quality measurement process among nodes in the network; in other words, nodes in the network measure their link qualities in sequence. When the data forwarding component forwards packets, the link quality measurement component updates link quality information accordingly and triggers forwarding sequence optimization if necessary.
- *Forwarding Sequence Optimization.* Currently the heart of design, the forwarding sequence optimization component implements EDR and EED optimization faithfully. The two optimizations are necessary to create optimal forwarding sequence of EED for comparison with ETX in the following subsection. In the future, we also plan to complete the forwarding sequence optimization component with inclusion of EEC implementation.
- *Data Forwarding*. The data forwarding component is shared by both DSF and ETX. Whenever a node has a packet ready to send, according to the specified forwarding scheme, the data forwarding component at a node attempts a single packet transmission to the designated forwarding node when it is in the active state.

B. On-Demand Sleep-Wake Algorithm

Dynamic Sleep Schedule for Single Destination

When the sing initiates a query, the query packet reaches the corresponding destination, which forwards the data packet up this path towards the sink. Let the query packet be forwarded in the path A, B, C, D, and E towards the direction of the destination when a query is initiated. The one hop neighbors (region A), two hop neighbors (region B), three hop neighbors (region C) change their schedule dynamically. This change is applied till the destination is reached and data is forwarded back to the sink.

Case 1: Sometimes, the exact hop length to the destination is unknown to the sink, but it may know the appropriate region (e.g. temperature of a node in north). In this case, the sink can send the query to the cluster head and the cluster head (knowing the location) can send it to the destination. The sleep schedule if the destination's location is not known is shown in Fig.3.1. Te node which forwards the query packet will change the sleep schedule as fully 'on' till it get the data packet. Whichever intermediate node (A, B, C, D) forwards the query packet will change it's a radio status to be on. When the destination node E sends the data packet, the intermediate nodes will change their radio to their usual schedule, after forwarding the data packet, the intermediate nodes will change their radio to their usual schedule, after forwarding the data packet. This dynamic schedule avoids the delay in transmission.

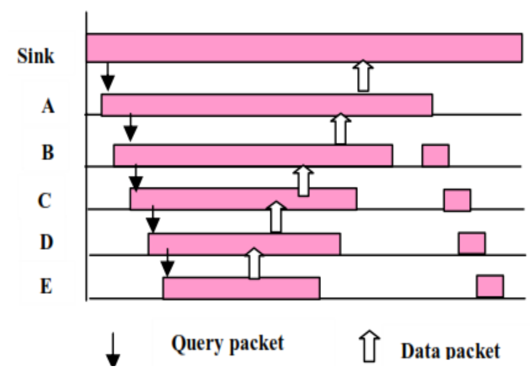

Fig. 3 Dynamic sleep scheduling if destination node's location is not known

Case 2: Sometimes the sing may know the destination's location in terms of hop length (e.g. current temperature of node X). The sleep schedule if the destination is known (i.e. how any hops away from a node), the schedule can be changed dynamically based on the arrival time of the data. The intermediate nodes calculate the time at which they have to forward the data using the following details: the time at which the query packet is forwarded, the distance of the destination node, time taken to transmit the data packet for one hop distance.

The partial data at each intermediate node flows up the tree towards the root. In this case, the nodes are activated only at the appropriate arrival time of the packets and energy saving is higher than case 1. However, the accurate wake up time is hard to estimate and each node is activated at the reserved time and is kept active for a timeout period. If the packet is not received within the timeout period, the node will switch off the radio.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

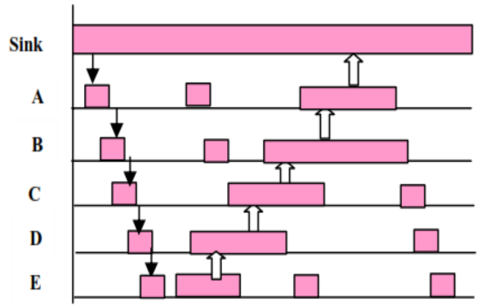

Fig. 4 Dynamic sleep scheduling if destination is known

Dynamic Sleep Schedule for Multiple Destinations

In some applications, the sink needs to gather the information from a set of sensor nodes instead of a single node. The sleep schedule if data should be collected from multiple destinations is shown in Fig 3.3. In this case, a multicast query is sent to the destination sensor nodes. There can be more number of packets to the sink for a single query, since many nodes send the sensed information. Hence, the radio of the intermediate nodes is made on, after a particular period of time. Unlike the previous scheme, the radio is not switched off after forwarding one data packet. After all the data packets are transmitted, a control packet is sent towards the sink, so that the intermediate nodes switch off the radio. The radio remains on until a control packet is received. The cluster head or a special node is responsible to send this control packet.

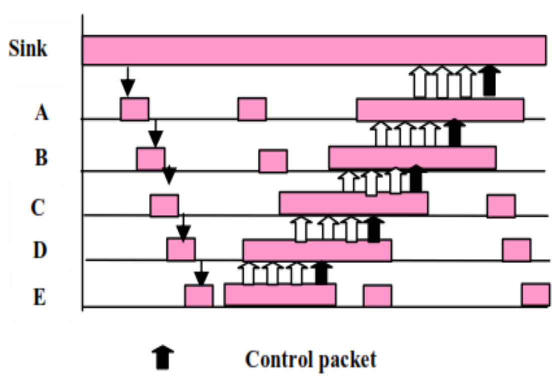

Fig. 5 Dynamic sleep schedule for multiple destinations

In many applications scenarios and network deployments, the network is dense and therefore most of the nodes at higher levels have many neighbors and they can communicate with many lower level nodes. Then take advantage of this fact in the multi-parent idea and exploit the full connectivity of the network. Instead of using a tree network topology where a single parent is assigned to each node in the network and the messages are always forwarded through the same fixed path, multiple paths and multiple parents with different wakeup schedules are associated with each node in the network.

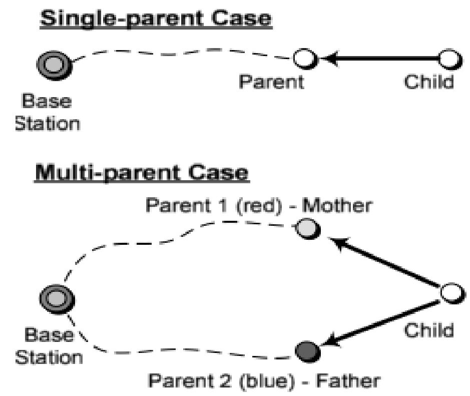

Fig. 6 Multi-Parent Idea

Basically, in the multi-parent idea when a message arrives to a node in the network, depending on its arrival time it chooses the fastest path in the network to get to its destination. When two parents (mother and father) are assigned to each node, if the mother is awake, the father can sleep and vice versa and the child node does not see any difference from a single-parent case. The base station belongs to all groups so it should wake up in all frames [5].
Procedure 1 is a handler for the event of detecting a target, which can be triggered by an interrupt that is raised on sensing something.

Procedure 1. OnDetectingTarget() - triggered when detecting a target
if (I am scheduled to be active) then
if (The target is *NOT* leaving the current awake region) then
return;
end if
end if
(Optional:) Run an alarm election algorithm;
if (I am selected as the alarm node) then
Calculate v_n and a_n;
Predict S_{n+1};
Compute $\mu S_{n+1}, \sigma S_{n+1}, \sigma_{\Delta n+1}$;
Compute $id_r, x_r, y_r, State(n), S_{n+1}, \mu S_{n+1}, \sigma S_{n+1}$ and $\sigma_{\Delta n+1}$;
end if
return;

For the formation frequency of awakened regions, use the target's motion trend as the criterion: when the target moves close to the edge of the current awake region, sensor node, which detects the target is leaving and is elected as the alarm node, broadcasts an alarm message to wake-up neighbors and form a new awake region.
Procedure 2 describes a sensor node's actions upon receiving an alarm message. This procedure can also be implemented as an interrupt handler.
Procedure 2 OnAlarmMessage() – Triggered when receiving an alarm message

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

1: Compute the distance d to the alarm node;
2: if ($d< \mu S_{n+1-} \sigma S_{n+1}$) then
3: return;
4: end if
5: Compute *§* with the alarm node position, my position S_{n+1};
6: Generate a random number *random=[0,1]*;
7: if (*random* > $P_{ss(§)}$) then
8: return;
9: end if
10: Compute t_{start} and t_{end};
11: SetTrackingTimer(t_{start});
12: return;

In step 2 of procedure 2, the node determines whether or not it is in the scope of the awaken region. In step 7, the node decides whether to be an awakened node or not. Finally, in step 11, the tracking timer is set so that the node can wake up at the scheduled time point.

Procedure 3 describes the tracking timer processing procedure, which controls the scheduled wake up/sleep and the mode switch.

Procedure 3. *OnTrackingTimer()* – Triggered when the tracking timer is out

1: if (mode == "*default*") then
2: mode = "*tracking*";
3: SuspendDefaultTimer();
4: SetTrackingTimer(t_{end});
5: else
6: mode = "*default*";
7: ResumeDefaultTimer();
8: end if
9: return;

IV. CONCLUSION

In this paper, a novel sleep scheduling method introduced which is based on the level-by-level offset schedule, to achieve low broadcasting delay in a large scale WSN. Novel sleep scheduling method also maintains long lived operation and high degree of security. In this paper, energy efficient Local Monitoring in Sensor Network (EELM) methodology, which consists of mechanisms that significantly reduce the node wake time required for monitoring. The performance of the generic on-demand sleep wake algorithm is evaluated via Glomosim simulator. Analytically proved that security coverage is not weakened by the protocol.

V. REFERENCES

[1] S.C.Ergen and P.Varaiya, "TDMA Scheduling Algorithms for Wireless Sensor Networks", Wireless networks, vol.16, no.4, pp.985-997, 2010.
[2] N.Bouabdallah, B.Sericola, "Continuous Monitoring using Event-Driven Reporting for Cluster-Based Wireless Sensor Networks", IEEE Trans. Vehicular Technology, vol.58, no.7, pp. 3640-3479, Sept. 2009.
[3] Y.Sun, O.Gurewitz, S.Du, L.Tang and D.B. Johnson, "ADB: An Efficient Multihop Broadcast Protocol Based on Asynchronous Duty-Cycling in Wireless Sensor Networks", Proc. Seventh ACM conf. Embedded Networked Sensor Systems, pp.43-56, Nov. 2009.
[4] M.I.Brownfield, K.Mehrjoo, A.S.Fayez and N.J.Davis, "Wireless Sensor Network Energy-Adaptive MAC Protocol", Proc. Third IEEE Consumer Comm. And Networking Conf. pp. 778-782, Jan. 2006.
[5] A.Keshavarzian, H.Lee, and L.Venkatraman, "Wakeup Scheduling in Wireless Sensor Networks", Proc. Seventh ACM Int'l Conf. Mobile Ad Hoc Networking and Computing, pp.322-333, May 2006.
[6] N.A. Vasanthi and S.Annadurai, "Energy Efficient Sleep Schedule for Achieving Minimum Latency in Query Based Sensor Networks", Proc. IEEE Int'l Conf. Sensor Networks, Ubiquitous and Trustworthy Computing, pp.214-219, June 2006.
[7] N.A. Vasanthi and S.Annadurai, "AWS: Asynchronous Wakeup Schedule to Minimize Latency in Wireless Sensor Networks", Proc. IEEE Int'l Conf. Sensor Networks, Ubiquitous and Trustworthy Computing, pp.144-151, June 2006.
[8] G.Lu, N.Sadagopan, B.Krishnamachari and A.Go Wireless Sensor Networks el, "Delay Efficient Sleep Scheduling in Wireless Sensor Networks", Proc. 24th IEEE Int'l Conf. Computer Comm., pp. 2470-2481, March 2005.
[9] T.Zheng, S.RadhaKrishnan and V.Sarangan, "PMAC: An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks", Proc. 19th IEEE Int'l Parallel and Distributed Processing Symp., pp. 224-231, April 2005.
[10] G.Lu, B.Krishnamachari, and C.Raghavendra, "An Adaptive Energy Efficient and Low-Latency MAC for Data Gathering in Wireless Sensor Networks", Proc. 18th IEEE Int'l Parallel and Distributed Processing Symp., pp. 224-230, April 2004.