An Agent Based Energy Efficient Local Monitoring

Rahul Dubey, Sanjeev Sharma

Abstract— Local monitoring is the one of the powerful technique for improving the security in multihope Wireless Sensor Network (WSN).Although it is a good technique for security purpose in WSN but it has a major drawback that it is costly in terms of energy consumption which make overhead for the energy constrained system such as WSN. In WSN environment, the scarce power resources are typically addressed through sleep-wake scheduling of nodes but sleep-wake technique is vulnerable even to simple attacks .In this paper a new technique is proposed that is not only energy efficient but also a secure technique which combine the sleep wake up scheduling with local monitoring which we call the OD –AEELMO (On Demand Agent Based Energy Efficient Local Monitoring).it enables sleep- wake management in secure manner even in face of adversarial nodes that choose not to awaken nodes responsible for monitoring their traffic.

Index Terms—Sensor networks, local monitoring, sleep/wake techniques, malicious node.

I. INTRODUCTION

The limited resources like memory, energy, bandwidth and the open communication medium and the deployment condition associated with WSN make these networks vulnerable to wide range of security attacks. Such as wormhole attacks, rushing, Sybil attacks [1], [2]. Cryptographic mechanisms alone are insufficient protection, since many of these attacks, such as wormhole and rushing attacks, can be launched without needing access to cryptographic keys or violating any cryptographic check. it has been demonstrated the local monitoring is a feasible mechanism to counter such attacks. Many techniques have been introduced that use the framework of local monitoring to achieve specific tasks such as intrusion detection [3], [4]. Building trust and reputation among nodes, [5], [6]. Protecting against control and data traffic attacks [9], [7], [2] and building secure routing protocols [1], [8], [7]. Though local monitoring has been demonstrated as a powerful technique for enhancing security of WSN. It results in a high energy cost since it requires the monitoring nodes to be constantly awake to oversee network activity. In this work, we propose sleep-wake protocol for optimizing the energy overhead of monitoring while main tanning the effectiveness of the monitoring service. The main challenge lies in ensuring that sleep-wake transition occur securely, so that an adversarial node cannot escape detection by causing its monitoring nodes to stay asleep. We propose the Agent Based Energy Efficient Local Monitoring (AEELMO) in WSN methodology, which consist of a set of mechanism that significantly reduce the node wake time required for monitoring.

These mechanisms derive from existing local monitoring techniques. Depending on the scenario, the proposed mechanisms either are modification of existing sleep wake used in network protocol henceforth referred to as the baseline sleep-wake scheme (SWS), or constitute a totally new protocol. In either case the goal of the protocol is to conserve energy while achieving the same level of security that was attained with the baseline local monitoring (LM). If a network does not have a baseline SWS, then Local monitoring is not modified since the goal is to reduce the impact of local monitoring on existing energy conservation schemes. This scheme don’t make the burden on hardware requirement At the highest level, the design of OD-AEELMO involves two steps-: First, developing the full list of malicious behaviors that a monitoring node needs to check (note that in baseline local monitoring, the malicious activities checked for are limited to drop, delay, modify, misroute, and fabricate) and second, defining the mechanism through which each check is to be performed. Additionally, the design of OD-AEELMO has to account for the fact that there is a delay incurred when the wake-up antennas awaken nodes upon receiving the control signal. Through a pipelined design we prevent this delay from accumulating over the hops between the communicating pairs of nodes. This results, for instance, in a constant delay independent of the number of hopes for the case in which the time to send a data packet is higher than that to send a control packet over one hop

II. BASELINE LOCAL MONITORING

Local monitoring is an intrusion detection scheme where a node monitors the control traffic going through neighbouring nodes this strategy mainly described in [2], but for this work it gives the basic knowledge of local monitoring which is useful in later section. For a node say α to be able to watch a node say N_2, α must be the neighbour of both N_2 and the previous hope N_1 then we call the α is a Agent of the link $N_1 \longrightarrow_{.} N_{2\,.}$ we use the R(N) to denote all the nodes those are with in the range of node N and $AG(N_1, N_2)$ to denote the all Agent of node N_2 over the link $N_1 \longrightarrow N_2$ Formally ,$AG(N_1, N_2) = R(N_1) \cap R(N_2) — N_{2\,,}$ where N_2 within the range of $R(N_1)$.

Manuscript Details received on October 30, 2011

Rahul Dubey School of information technology, Rajiv Gandhi technical university, Bhopal, India, 08103309269,(e-mail: rahul_nvision87@ yahoo.co.in)

Dr. Sanjeev Sharma, School of information technology, Rajiv Gandhi technical university, Bhopal, India 9407510528 (e-mail:sanjeev@rgtu.net)

*Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication*

For example in figure 1 AG(X, A) = {M, N, X}. Information from each packet sent from X to A is saved in a watch buffer at each Agent. The Agent expects that A will forward the packet toward the ultimate destination, unless A is itself the destination. Each entry in the watch buffer is time st-mp with a time threshold $T_{w,}$ by which A must forward the packet. Each packet forwarded by A with X as a previous hop is checked for the corresponding information in the watch buffer. The check can be to verify if the packet is fabricated, corrupted, dropped, delayed misrouted.

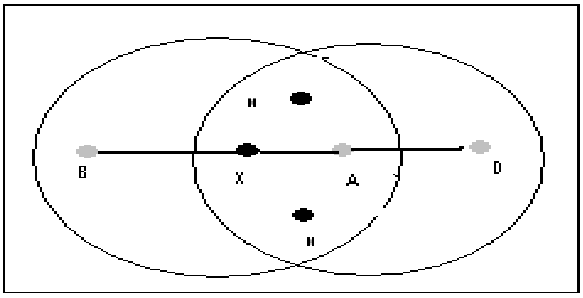

Fig 1 Local Monitoring

A malicious counter (MalC (i , j)) is maintained at each Agent node i, for a node j, at the receiving end of each link that i is monitoring over a sliding window of length T. (MalC(i ,j)) is incremented for any malicious activity of j detected by i. When (MalC (i, j)) value crosses a threshold rate Malcth over Twin , node i revokes j from its neighbor list (called direct isolation). The notion of enough number of alerts is quantified by the detection confidence index (DCI). A node becomes isolated when all its first-hop neighbors revoke it either directly or indirectly.

III. PROPOSED WORK

Aeelmo Protocol Description:

The primary goal or the AEELMO is to minimize the time a node has to be awake specially for the purpose of performing local monitoring. Nodes are of course also awakened for other purposes such as measurement and communications, but when the waking time intervals for various tasks overlap it is understood that the various node function are to be performed simultaneously. Local monitoring is used

To make sure that packets are not dropped delayed modified misrouted or forged along the path from source to destination [7]. AEELMO adds one more task to the list of events that an Agent node needs to check verifying whether the node being monitored actually wakes the relevant Agent or fails to do so due to malicious intent. Depending on the baseline SWS used in the network, AEELMO has three different mechanism for putting nodes to sleep in network with local monitoring these are the AEELMO protocols for following types of sleep-wake protocols:

1- Synchronized sleep-wake sensor networks
2- Continuously acting sensor networks
3- Triggered sensor networks

But my work is mainly dedicated to the triggered sensor network so the explanation of it is given below.

A. Triggered Sensor Network

We now developed a protocol that work on demand approach. It means here node activity is triggered (by any event or clock). Such type of network is referred the triggered sensor networks. We introduce a new sleep-wake protocol, called On-demand AEELMO, that enable the Agents to go to sleep when not required for monitoring. The approach we take is on-demand sleep-wake of the Agent rather than scheduling the sleep-wake periods. The defining characteristic of on-demand sleep-wake protocols is that any node in the network may, at random, initiate communication with other node in the network. The sleep-wake protocol does not rely on any fixed communication pattern in the network.

B. OD- AEELMO Approach

The basic idea in designing OD-AEELMO is for a node to wake up the requisite Agent nodes to perform local monitoring on the communication that is going out from that node. The challenge in the design comes from the fact that any of the nodes (except source) may be malicious and therefore, not faithfully wake up the suitable Agent. As shown in Fig 2 which explain the basic approach of OD- AEELMO. A source node S is sending data to a destination node D through an h-hope route $S \longrightarrow n_1 \longrightarrow n_2 \longrightarrow n_3 \longrightarrow n_4 \longrightarrow n \longrightarrow_5 n_6$ $\longrightarrow$........... $\longrightarrow$D. In a network where all the nodes are honest, S will wake up the next hope n_1 and the Agent of n1 (as shown in FIG2.) before sending the packet to n_1.in turn n_1 will wake up $n_2$and Agent node of n_2 (as shown in FIG2.) before sending the packet on the next hop and so on till the packet reaches D. Formally [2] the responsibility of a Agent node α of n_{i+1} over a link $n_i \longrightarrow n_{i+1}$ is to verify the following node behaviour.

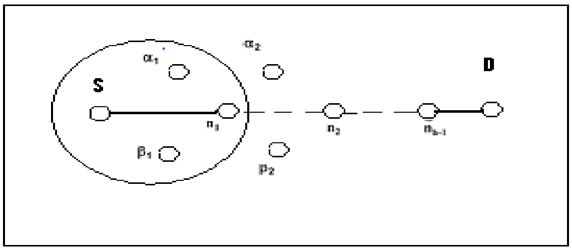

Fig 2. h-hopes route between S and D, neighbor of S and Agents of n_1 and n_2

In it condition 1-3 are known but 4 is added in OD-AEELMO

C1-Node n_{i+1} relay the packet to the correct next-hop in time $T_{w.}$

C2- node n_{i+1} does not illegally modify the packet it forwarding.

C3- node n_{i+1} only relays a packet if a packet is sent on $n_i \rightarrow n_{i+1.}$

C4- n_{i+1} should wake up the sufficient and suitable set of guard of n_{i+2} over $n_{i+1} \longrightarrow n_{i+2.}$

If one of the first three node behaviors above is not in effect, then the MalC value is incremented by the appropriate amount.

If the node behavior 4 is not in effect, the MalC increment is the maximum of the other MalC values because this violation may be used to mask the violations of any of the first there behaviors. In the following section the Agent scheduling algorithm followed by two variation of OD-AEELMO depending on the wake up mechanism.

C. Agent Scheduling Algorithm (AGS)

This algorithm is used to select the sufficient and suitable set of Agent (SSAG) that is required to monitor a certain communication link. Sufficient means enough number of Agents to completely isolate a malicious node. It depends on the detection confidence index (DCI) of local monitoring. The number of selected Agent should be at least equal to DCI. Suitable means the selection of the Agent that is capable to detect the malicious action even under the transmission power level control attack [21]. To illustrate the idea consider the Fig 3 The malicious node M tries to drop a packet while avoiding being detected by controlling its transmission power level (the dotted circle in Fig 3) to exclude the next-hop node D. Node M succeeds if the selected Agent is α and fails if the selected Agent is β. Node α is included within the reduced transmission range and falsely thinks that M faithfully relayed the packet. However, β is out of the reduced range and correctly accuses M of dropping the packet. Thus, the Agent β would be more suitable to monitor this link .S ——→M than α. AGS use the following parameters CS for current sender, CR for current receiver and LLOC (i) is used to represent the first hop, second hop neighbors, location of node i is Li ,NH is used to refer to the next hop node from the current receiver (CR). Assume S is executing the algorithm to wake up the sufficient and suitable set of Agent to monitor M over the link S ——→M then the input parameter to AGS are CS=S,CR=M,NH=D, and LLOC(s)={Lα, Lβ, Lm, Ld, Lz} where Li is the (X , Y) location of node i. The algorithm is shown below.

```
AGS (DCI, CS, CR, NH, LLOC(S))
{   SSAG= {};
**FIND THE LIST OF ALL POSSIBLE GUARD **
Return (SSAG) ;}
```

In next section the two variation of OD- AEELMO is explained

D. Master OD-AEELMO

The high level design goal in M-AEELMO is to minimize the energy wasted in waking up nodes that are not within the sufficient and suitable set of Agents. In M-AEELMO, a node wakes up only the set of sufficient and suitable Agents. For this, it is assumed that the wake-up antenna of each node is tuned to receive at its own code (as in [16]), which is distinct for all one-hop neighborhood nodes. On average only half of the nodes within a single transmission range may serve as Agents over a certain link ([2]). Of those Agents only a subset satisfies the sufficient and suitable criteria.

In Fig α and β are valid Agent of M over the link from S to M, while Z is not. Node β is a suitable Agent while α is not. Also, note that the energy spent in warm-up (transition between sleep mode and wake-up mode) is relatively high almost three times as much as the energy spent in listening for the antennas described in [16]). Thus, awakening the appropriate nodes saves considerable amount of energy. In Fig 2, a Agent of n_1 say α_1 knows the location of its neighbor n_1 and the location of all the neighbors of the common neighbors of n_1 (S, $\beta_1\beta_2\alpha_2$ and n_2) using this information α_1 knows as the Agents of n_2 over the link $n_1 \longrightarrow n_2$. Therefore, α_1 cannot be suitable Agent for n_2. A disadvantage of the M-AEELMO is that it requires the sophisticated wake-up hardware that can be identified using an id-attached beacon [16]. Steps of M-AEELMO

Assume that node S has a data to be sent for the destination D over the route S $\longrightarrow n_1 \longrightarrow n_2 \longrightarrow n_3 \longrightarrow n_4 \longrightarrow n \longrightarrow {}_5n_6 \longrightarrow$.......... $\longrightarrow$D

1- Node S runs AGS(DCI, S , n_1 , n_2, LLOC(S)) which returns SSAG(n_1) that contains the set of suitable and sufficient Agents to monitor its communication with the next hop node(n_1)

2- Node S sends a signal to wake up the first hop node (n_1)and the Agents in SSAG(n_1)(α_1, β_1) this is a multicast signal that contains the identities of and each member of this signal is to guaranteed to wake up the correct Agents of n_1 due to the assumption of honest source S.

3- Node S sends packets to n_1 following the timing schedule decided.

4- Nodes n_1, α_1 and β1 after being awakened continue to remain awake for T_w. Recall that is a parameter of local monitoring that capture the maximum time by which an entry in the watch buffer is evicted. Each time a new packet is sent from S to n_1, Tw is reinitialized. After T expires at a node, the node goes back to sleep.

5- Node n_1, after being awakened ,runs AGS(DCI, n_1 , n_2, n_3, LLOC(n_1))to get SSAG(n_2).then n_1 uses the timing schedule as decided to send a wake up signal to n_2 and the Agents in SSAG(n_2).

6- Node S and each of the Agents in SSG(n_1) run AGS(DCI, n_1 , n_2, n_3, LLOC(i)) where i €{S, SSAG(n_1)} to get SSAG(n_2). This is to verify that n_1 awakes the correct set of sufficient and suitable Agent for n_2 over the link n_1 $\longrightarrow n_{2.}$

7- Node n2does not accept packets from n1 if the wake up signal of n1 does not include the correct set of sufficient and suitable Agents node n2 gets the set by executing AGS (DCI, $n_{1,}$ n2, n_3, LLOC (n_2)).

8- If n_1 fails to send the wake up signal the Agent of n1 with the lowest ID send a two hop broadcast of the wake up signal it that Agent fails, the guard with the next smallest ID sends the signal and so on. This design ensures that if there is a chain of colluding malicious nodes then all these nodes will be suspected.

9- The process continues at each communication link along the route to the destination.

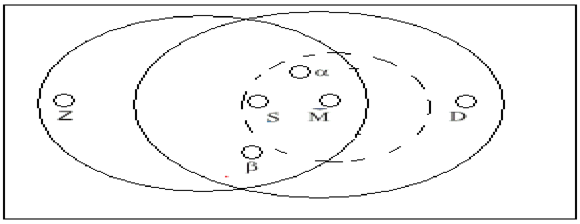

FIG 3 Transmission power control attack

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

E. Simple OD-AEELMO

The high level design goal of S-AEELMO is to simplify (hence the name) the wake-up hardware and wake-up signal. In S-AEELMO, a node that triggers the wake-up signal awakens all its first-hop neighbors through a simple one-hop broadcast. This simple awakening process simplifies the wake-up hardware and the wake-up signal since no coding is compared to M-AEELMO required. However, it increases energy consumption compare to M-AEELMO due to the needless awakening of the guards. Again referring to Fig. 2, S-AEELMO neighbors that are not within the set of sufficient and suitable Agents uses the following steps to awaken the Agents along the route from S to D:

1- Node S broadcasts the wake up signal to all its first hop neighbors (Z, W, n1, α1, and β1). The wake up signal includes the identity of both the current sender and next two hops ($n_{1,}$ n_2).

2- Each neighbor of S, after being awakened, decides whether to stay awake or go back to sleep based on the role that it may play on the ongoing communication 1) if that neighbor is the next hop , it stays awake to receive the data and to monitor the next hop from it (n2) 2) if that neighbor is a Agent for the next hop n_1 over the link s ——→$n_{1\ ,}$ it runs AGS (DCI, $n_{1,}$ n_2, n_3, LLOC (α_1)) to find SSAG(n_1).if α_1 is within the first DCI entries of SSAG(n_1), it stays awake to monitor the behavior of n_1, other wise it goes back to sleep. Note that the seed or the randomization process in the AGS algorithm is the same. 3) If none of previous two cases hold the node goes back to sleep immediately. Here the correct Agents are guaranteed to be awakened because S is assumed to be honest.

3- Node S sends packets to n_1 following the timing schedule as decided.

4- The nodes in SSAG(n_1) and n_1 after being awakened continue to stay awake for T_w after that they go back to sleep.

5- Node n_1 does the same step that S did to wake up the next hop n_2 and n_2's Agents. The nodes in SSAG (n_1) verify the wake up signal or n_1.

6- If n_1 fails to send the wake up signal the Agent of n1 with the lowest ID sends a two hop broadcast of the wake up signal it that Agent fails, the guard with the next smallest ID sends the signal and so on. This design ensures that if there is a chain of colluding malicious nodes then all these nodes will be suspected.

7- The process continues at each communication link along the route to the destination.

F. OD-AEELMO Wakeup Time Scheduling

In this section, we generate the timing schedules for Awakening the relevant nodes and Agent using OD-AEELMO. This is important because the wake-up antennas have a warm-up period that could increase the end-to-end delay of the Communication. We design the schedule such that the moreover, the increase in delay with the number of hops is Cumulative with the number of hops for Case II shown below. It is a constant independent of the number of hops. Additional delay due to the sleep-wake protocol is no small for Case I. In this case, the coefficient of increase per packet and the time to send a data packet over one hop is the difference between the time to send a control packet and the time to send a data packet over one hop. The following terms are used to derive the expression for the two cases. TOC is time to send the signal to wakeup antenna, TOW the time to fully wake up a node, TOD time to send a packet over one hope, TWAK time till the node awakened from the wake up.

Consider an isolated flow between S and D, separated by hops. The intermediate nodes are n_1 n_2 n_3..........n_{h-1} a_i represents the Agent of node n_i over the link n_{i-1}——→$n_{i.}$ Let v_i denote the set of neighbors of n_i that are not within the Agents of n_{i+1} over the link n_i——→$n_{i+1.}$ Consider the two disjoint cases based on the relation between (TOC + TOW) and TOD

$$\Gamma = (TOC+TOW) - TOD$$

CASE I -: (TOC+TOW) > TOD (Γ > 0)

As shown in Fig 4 a shows the timing schedule for a node in the route between the source and the destination. The node n_1 , awakes at T_3 and sleeps at $T_{8\ ,}$ where T_8-T_3 =TOD(time to receive data) +Γ(wait for the next hop to be ready to receive the data)+ TOD(send the data to the next hop) +{TOD+Γ}(as a Agent for n_2) = 3TOD+2Γ and Fig 2.4 b shows the timing schedule for Agent node that is= 2TOD+Γ . Fig 4.c shows the timing schedule for a node that is a neighbor to a node in the route from the source to the destination but is not a Agent node. The node $v_{1,}$ awakes at T_3 determines that it cannot be a Agent and thus goes back to sleep immediately.

CASE II -: (TOC+TOW) < TOD (Γ < 0)

As shown in Fig 5 a shows the timing schedule for this case. The node n_1 , awakes at T_3 and sleeps at $T_{9\ ,}$ where T_9-T_3 =TOD(time to receive data) + TOD(send the data to the next hop) +{TOD }(as a Agent for n_2) = 3TOD and Fig 5 b shows the timing schedule for Agent node that is= 2TOD. The timing schedule for a node that is a neighbor to a node in the route from the source to the destination but is not involved in monitoring is the same as its peer in Case I

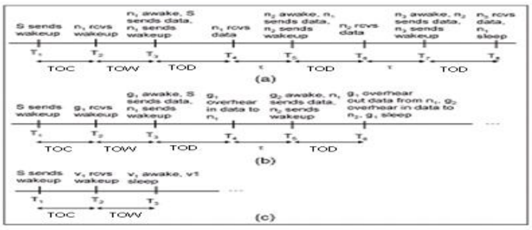
S sends wakeup
n₁ rcvs wakeup
n₁ awake, S sends data, n₁ sends wakeup
n₁ rcvs data
n₂ awake, n₁ sends data, n₂ sends wakeup
n₂ rcvs data
n₃ awake, n₂ sends data, n₃ sends wakeup
n₃ rcvs data, n₁ sleep
T₁
T₂
T₃
T₄
T₅
T₆
T₇
T₈
TOC
TOW
TOD
τ
TOD
τ
TOD
(a)
S sends wakeup
g₁ rcvs wakeup
g₁ awake, S sends data, n₁ sends wakeup
g₁ overhear in data to n₁
g₂ awake, n₁ sends data, n₂ sends wakeup
g₁ overhear out data from n₁, g₂ overhear in data to n₂, g₁ sleep
T₁
T₂
T₃
T₄
T₅
T₆
TOC
TOW
TOD
τ
TOD
(b)
S sends wakeup
v₁ rcvs wakeup
v₁ awake, v1 sleep
T₁
T₂
T₃
TOC
TOW
(c)

Fig 4 case I wake/sleep timing schedule a) node in the route. b) A Agent node. C) A neighbor to a node in the route that is not valid Agent

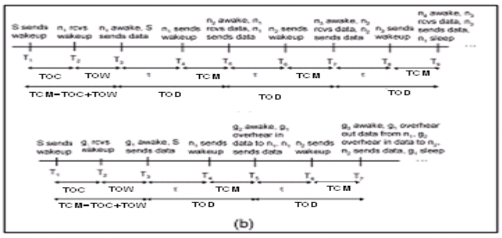

Fig 5 case II wake/sleep timing schedule for, a) node in the route. b) A, Agent node.

IV. PERFORMANCE ANALYSIS

Performance analysis of these algorithms can be done by the following parameter analysis in mathematical form.

A. End-to- End Delay Analysis.

To bring out the worst case End-to-End delay behavior of OD-AEELMO we compare it with local monitoring without Sleep-wake scheduling, which we call Baseline Local Monitoring [2]. In addition to the notations defined in previous Section, let ATRANS be the current to transmit (at the middle of the transmit range), which is 27 mA for Mica2. Let AOW be the current consumed during the transition from sleep to wake-up (warm-up), which is 30 mA for Mica2 motes [16]. Finally, let AACT be the current in the computationally active mode=the current in the idle listening mode =the current in receive mode.

Let us consider a flow between S and D separated by h-hops S $\longrightarrow n_1 \longrightarrow n_2 \longrightarrow n_3 \longrightarrow n_4 \longrightarrow n \longrightarrow {}_5n_6 \longrightarrow$........... $\longrightarrow n_{h-1.}$ The bounding box around Sand D covers all possible nodes including the forwarding and the Agent nodes that may be involved in the communication between S and D. The size of the bounding box is 2r (h+1) r=2r^2 (h+1) where r is the transmission range. For OD-AEELMO considering the wake up sleep scheduling cases of previous section.

CASE I -: (TOC+TOW) > TOD ($\Gamma > 0$)

From Fig 2.1it can be seen that the delay at the first link S $\longrightarrow n_1$ is TOC+TOW+TOD. Over each of succeeding links, the delay is TOC+TOW since delay due to data TOD gets exposed. This is due to the sleep wake schedule process that OD-AEELMO uses where the wake up signal is sent at the earliest opportunity. Therefore the end-to-end delay in OD-AEELMO is $\mu_{AEEELMO}$ (h) for the link from S to D is

$\mu_{AEEELMO}$ (h) = (TOC+TOW+ TOD) + (h-1) (TOC+TOW) (1)
= (h) (TOC+TOW) + TOD

The end-to-end delay in BLM

μ_{BLM} (h) = (h) TOD (2)

So from equation 1 and 2 the additional delay for OD-ELMO is

$\mu_{AEEELMO-AD}$ (h) =μ_{AEELMO} (h) - μ_{BLM} (h) =h. r+ TOD (3)

CASE II -: (TOC+TOW) < TOD ($\Gamma < 0$)

For this case the end-to-end delay in OD-AEELMO is exactly same as that for case I (1) after exchanging TOD with (TOC+TOW)

μ_{AEELMO} (h) = (TOC+TOW+ TOD) + (h-1) (TOD)
= (h) TOD + (TOC+TOW) (4)

Therefore using (2), (4) the additional delay for OD-AEELMO .$\mu_{AEELMO-AD}$ (h) =μ_{AEELMO} (h) - μ_{BLM} (h) = TOC+TOW (5)

V. SIMULATION RESULT

In previous section, we have explained our proposed protocol. In this chapter, we simulate and analyze the performance of our proposed protocol OD-AEELMO. Over the existing protocols Base Line Monitoring .There are various simulation tools available like NS-2, OPNET, OMNeT++, J-Sim, GlomoSim, Qualnet, TOSSIM etc., for simulation of Sensor Networks. These all simulators are working on object oriented concept and. We use the ns-2 to simulate a data exchange protocol over a network with local monitoring enabled. We conduct simulation experiments for a sensor network in two scenarios; one (the baseline) is without OD-AEELMO, and the other in which OD-AEELMO is incorporated. The OD-AEELMO scenario is built on the baseline scenario to control sleep-wake of the Agent. The nodes are distributed randomly over a square area with a uniform node density. Each node acts as a source and generates data according to a Poisson process of rate λ. destination is chosen at random and is changed using an exponential distribution of rate μ. A route is canceled if unused for a time period of T_{Route}. The results are averages over 14 runs. The malicious nodes are chosen at random. All sources and destinations are chosen so that any source/destination pair is more than two hops apart.

A. Simulation Setup

We have assumed a road map uses the parameter given below in the table 1 for the simulation and result analysis

Table 1 Simulation parameter

Parameter	**Value**
Propagation Model	Two ray model
Area	100 square meter
# of nodes	20
BW(kbps)	40
Fraction of data monitored (f_{dat})	0.6
Avg no neighbors (N_b)	8
Simulation time	120 s
T_{route}	50 s
# malicious node	4
Warm up time TOW	5 ms
Packet Generation Rate	0.1/sec

Initially, the nodes are randomly located in the area. The area is considered as 100 square feet. The total number of nodes considered for the simulation is 20 .The propagation model used is Two ray model and the bandwidth used is 40 kbps and packet generation rate is 0.1/s and node warm-up time is 5ms and no. of neighboring nodes are 8 where the no. of malicious node is 4 and the fraction of data monitored by the Agent node is 0.6 Table1 shows the simulation parameters used to evaluate the performance of the OD-AEELMO.

B. Performance evaluation

The simulation environment for the simulation and analysis of our proposed methodology is shown in figure 6

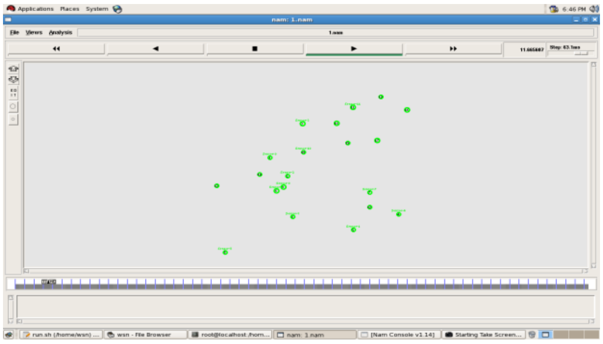

Fig 6 Simulation Environment

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

5.2.1 Input parameters

1. Fraction of data monitored (f_{dat})—each Agent node randomly monitors this fraction of the data packets.
2. Data traffic load (1/µ).
3. Number of malicious nodes (M).

5.2.2 Output parameters

1. Delivery ratio—the ratio of the number of packets delivered to the destination to the number of packets sent out by a node averaged over all the nodes in the network;
2. Percent wakeup time—the time a node has to be awake Specifically to do monitoring, averaged over all the nodes as a percentage of the simulation time;
3. Average end-to-end delay—the time it takes a data packet to reach the final destination averaged over all successfully received data packets;
4. Percent isolation—the percentage of the number of malicious nodes that is isolated as a fraction of the total number of malicious nodes;
5. Percent false isolation—the percentage of the number of nodes that is isolated due to natural collisions on the wireless channel as a fraction of the total number of nodes.
6. Isolation latency—the time between when a malicious node performs its first malicious action to the time of isolation, averaged over all isolated malicious nodes.

With the given parameter we analyze our technique with respect to the baseline technique.

5.2.3 Effect of Number of Malicious Nodes (M)

Fig 7 shows the variations of percent delivery ratio, percent isolation, and percent false isolation as we vary M. Fig 7(b) shows that the percent delivery ratio slightly decreases as M increases. This is due to the packets dropped before the malicious nodes are detected and isolated. As M increases, this initial drop increases and thus the delivery ratio decreases. Fig 7 c) shows that the percent isolation also slightly decreases as we increase M. This is because the number of available Agent in the network decreases as more and more nodes get compromised. These two metrics in OD-AEELMO are slightly lower than those of the baseline due to the erroneous extra sleep Fig 7 (a) shows that the percent false isolation decreases as we increase M. This comes from the fact that the number of good nodes decreases as we increase M.

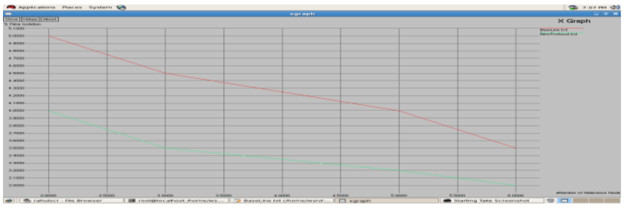

FIG 7 (a) False Isolation Vs Malicious Node

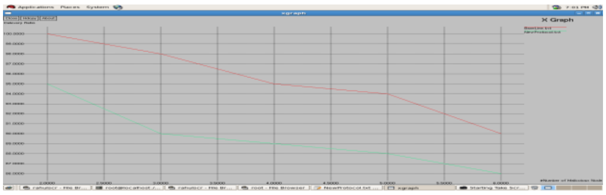

FIG 7 (b) percent delivery ration Vs Malicious Node

This in turn results in a decrease of the Indirect false isolation since a node may not have more than DCI good nodes to agree on falsely moreover, as M increases, the data traffic decreases since malicious nodes are not Again, this is due to the loss of some packets that may falsely identify a node as malicious due to the erroneous extra sleep in OD-AEELMO.

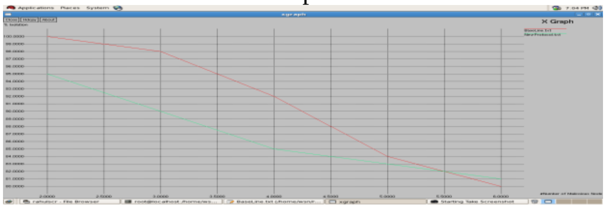

FIG 7 (c) Isolation Vs Malicious Node

5.2.4 Effect of data traffic load (1/µ).

Fig.5.6 shows the performance as we vary the data traffic load (1/µ). Fig.8 (c) shows that the percent false isolation increases as the traffic load increases. As the traffic load increases, the probability of collision increases. This in turn increases the possibility of false accusation since a Agent say A, may falsely accuse a node say N, of not forwarding a packet if either A has a collision when N forwards or N has a collision while receiving the packet Fig. 8 b) shows that the isolation latency increases as the traffic load increases. As the traffic load increases, we decrease the MalC increment to alleviate the increase in false detections. This causes the MalC threshold to be reached slower at an Agent node, which results in increasing the isolation latency of the malicious nodes. Also the higher traffic load lays it open to the possibility of some packets being missed due to natural collisions, and thereby. Preventing reaching the threshold faster. Note that the isolation latency in OD-AEELMO is higher than that of the baseline because of the additional packets missed due to the erroneous extra sleep. Fig. 8(a) shows that end-to-end delay increases as the Data traffic load increases due the higher contention for the channel. The end-to-end delay in OD-AEELMO is slightly higher due to the extra delay in packet forwarding in OD-AEELMO resulting from the warming up of the nodes. Note that as the traffic load increases the behavior of OD-AEELMO gets closer to that of the baseline. This is due to the increase in the wakeup time forced by the need to communicate more traffic.

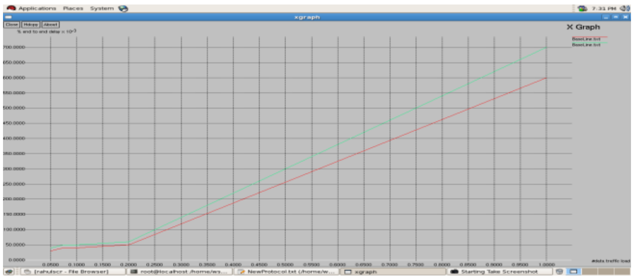

FIG 8 (a) End to End delay Vs Data Traffic Load

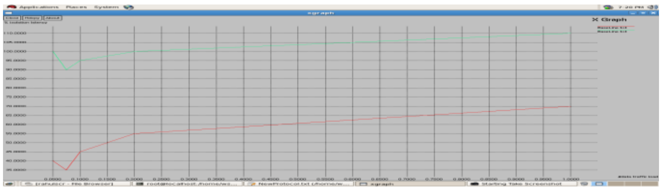

FIG 8 (b) Isolation Latency Vs Data Traffic Load

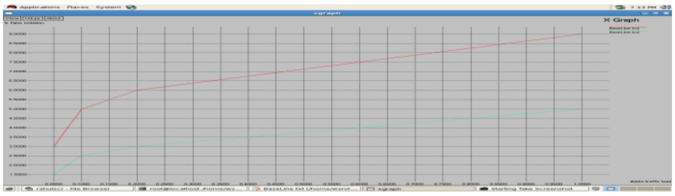

FIG 8 (c) False Isolation Vs Data Traffic Load

5.2.5 Wake-Up Time Variations

In this section we compare the effect of varying f_{dat}, M, and 1/µ on the percent wake-up time for OD-AEELMO with (DCI=3) Fig. 9 (a) shows that the percent wake-up time required for monitoring increases as f_{dat} increases due to the increase in the number of data packets that a node needs to overhear in its neighborhood. Fig. 9 (b) shows that the percent wakeup time decreases as we increase M. As M increases, the number of data packets in the system decreases since the malicious nodes are isolated and disallowed from generating data packets. Therefore, the number of packets that need to be monitored decreases, which results in a decrease in the average percentage of wake-up monitor time. Fig. 9 (c) shows that the percent wake-up time increases as the data traffic load increases due the increase of data packets that need to be monitored.

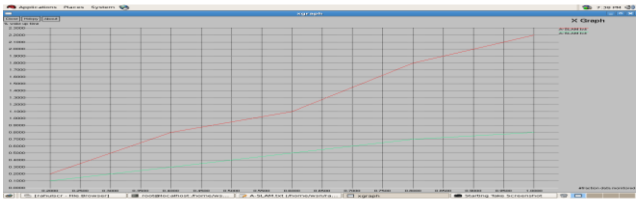

FIG 9 (a) Wake-up Time Vs Fraction of Data Monitored

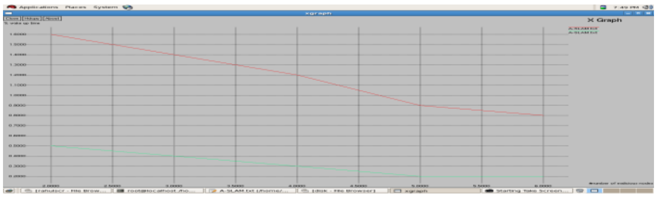

FIG 9 (b) Wake-up Time Vs Malicious node

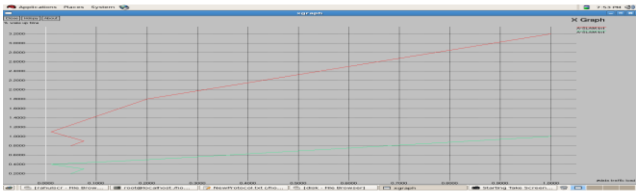

FIG 9 (c) Wake-up Time Vs Data Traffic Load

5.2.6 Protocol Delay Comparison in CASE I and CASE II

In the previous section we have discussed the overall concept of the proposed protocol in two cases and analysis is given below. As shown in Fig 10 we plot the extra delay of OD-AEELMO over that of BLM for case I (3) and case II (5) above with TOD=7ms and $\Gamma=1$ The figure shows that the additional delay due to OD AEELMO increases linearly with the number of hops for Case I while it remains constant for Case II

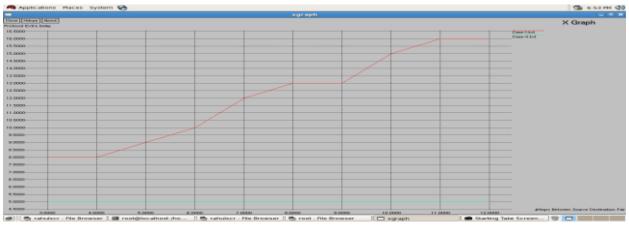

FIG 10 Protocol Extra Delay Vs No of hopes b/n Source & Destination

VI. CONCLUSION

As we saw in the previous work of the Energy Conservation main focus was on the node wake Up and sleep without considering the security during the data communication but in our work we not only considered the Energy of node but the security of data communication also by extending the concept of the Local Monitoring with the Energy Conservation scheme. Here we present the OD-AEELMO protocol this is called on demand because it considers the triggered network.. Our technique also requires special type of hardware support which consist passive or low power antenna. By the simulation and analysis it has proven that our protocol not only conserve the energy of the nodes but also satisfy the constraint of the Local Monitoring.

REFERENCES

1. S. Marti, T.J. GiulI, K.Lai, and Baker, “Mitigating Routing Misbehaviour in Mobile Ad Hoc networks,” proc MOBICOM, pp. 255-265, 2000.
2. I.Khalil, S. Bagchi, and N.B. , Shroff, “Design and Analysis of A Protocol for Detection and Isolation of the wormhole attack in Multihop Wireless Networks,” IEEE/Create Net Secure Comm. , pp. 89-100, sept. 2005.
3. Y. Haung and W. Lee, “A Cooperative Intrusion Detection System for Ad Hoc and Sensor Networks,” pp. 135-147, 2003
4. A. Silva, M.Martin, B.Rocha, A. Loureiro, L.Ruiz and H.wong,” Decentralize Intrusion Detection in Wireless Sensor Networks, Proc. First ACM Workshop Quality of Service and Security in Wireless and Mobile Networks, pp. 16-23, 2005.
5. A.A. Pirzada and C.Mcdonald,”Establishing Trust In Pure Ad Hoc Networks,” Proc. 27th Australasian Computer Science Conf.(ACSC ’04), pp. 47-54, 2004.
6. s. Buchegger and J.-Y. Le Boudec,” Performance Analysis of The CONFIDANT Protocol; Cooperation of NodesFareness in Distributed Ad Hoc networks,” Proc. Symp. Mobile Ad Hoc Networking and Computing (MOBIHOC), pp. 80-91, 2002.
7. I. Khalil, S. Bagchi, and C. Nina-Rotaru, “Dicas: Detection Diagnosis and Isolation of Control Attacks in Sensor Networks,”IEEE/Create Net Secure Comm., pp. 89-100, Sept. 2005.
8. S.J. Lee and M. Gerla, “Split Multipath Routing with Maximally Disjoint Paths in Ad Hoc Networks,” Proc. IEEE Int’l Conf. Comm. (ICC), pp. 3201-3205, 2001.
9. Abhisek Pandey and R.C. Tripathi”A Survey On WSN Security” proc IJCA/0975-8887/volume 3-No 2, June 2010.
10. B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, “Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks,” Proc. MOBICOM ’01, 2001.
11. W. Ye, J. Heidemann, and D. Estrin, “An Energy Efficient MAC Protocol for Wireless Sensor Networks,” Proc. IEEE INFOCOM, pp. 88-97, 2002.
12. A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, “Wireless Sensor Networks for Habitat Monitoring,” Proc. ACM Int’l Workshop Wireless Sensor Networks and Applications pp. 1567-1576, 2002.

13. R. Naik, S. Biswas, and S. Datta, “Distributed Sleep-Scheduling Protocols for Energy Conservation in Wireless Networks,” Proc. 38th Ann. Hawaii Int’l Conf. System Sciences (HICSS), pp. 285b-285b 2005.
14. J.W. Hui, Z. Ren, and B. Krogh, “Sentry-Based Power Management in Wireless Sensor Networks,” Proc. Second Int’l Workshop Information P rocessing in Sensor Networks (IPSN), pp. 458-472, 2003
15. S. Liu, K. Fan, and P. Sinha, “Dynamic Sleep Scheduling Using Online Experimentation for Wireless Sensor Networks,” Proc Analysis of Wireless Sensor Networks (Symmetric), 2005 Third Int’l. Workshop Measurement, Modelling and Performance.
16. L. Gu and J.A. Stankovic, “Radio-Triggered Wake-Up Capability ForSensorNetworks,”Proc. IEEE Real-Time and Embedded Technology and Applications Sym., (RTAS), pp. 27-36, 2004
17. Y.H. Chee, “Ultra Low Power Transmitters for Wireless Sensor Networks,” PhD dissertation, Univ. of California, 2006
18. B. Cook, A. Berny, S. Lanzisera, A. Molnar, and K. Pister, “Low-Power 2.4-GHz Transceiver with Passive RX Front-End and 400 mV Supply,” IEEE J. Solid-State Circuits, vol. 41, no. 12, pp. 2757 2766, Dec. 2006
19. N. Pletcher, “Ultra-Low Power Wake-Up Receivers for Wireless Sensor Networks” PhD dissertation, Univ. of California, 2008. AS3931Product_brief_0204.pdf, 2011
20. http://www.austriamicrosystems.com/03products/data.
21. I. Khalil and S. Bagchi, “MISPAR: Mitigating Stealthy Packet Networks,” Proc. Fourth ACM SecureComm. pp. 1-10, 2008 Dropping in Locally Monitored Multi-Hop Wireless Ad Hoc.

AUTHORS PROFILE

Rahul dubey received the M.Tech .degrees in Information Technology from School of information technology, Rajiv Gandhi technical university, Bhopal in 2011 and received the B.E degree in computer science from Samtrat Ashoka Technological Institute Vidisha in 2009.

DR. Sanjeev Sharma received his Doctor of philosophy in information technology from R.G.P.V BHOPAL. He is currently working with university teaching Department R.G.P.V BHOPAL as a Department head. He has a 20 year of experience in teaching field. His research interest includes network security system, Mobile Adhoc Network. He has various published national and international papers and journal.