Network for Research and Development in Africa

International Journal of Information, Technology & Innovation in Africa
ISSN: 2360-9772. Volume 11, Number 4
Pages 01-16 (April, 2019)
www.arcnjournals.org

A Secured Routing Protocol for Wireless Sensor Networks Using Q-Learning

Fagbohunmi Griffin Siji[1] and Eneh, I. I.[2]
[1]Computer Science Department, Abia State Polytechnic Aba, Abia State | Email: fagbohunmisiji@yahoo.com
[2]Enugu State University of Science and Technology, Enugu, Enugu State

Abstract: *Wireless sensor networks (WSNs) consist of spatial distribution of sensors which co-operatively monitor the environment for certain phenomenon of interest such as temperature, humidity, pressure etc, and send their sensed data through multi-hop route to the sink.. These nodes may vary between hundreds to thousands depending on the size and nature of data (signals) to be detected. Wireless sensor networks are expected to operate over long periods without being attended to. The range of this period may span from some months to even years. However, due to its resource constraints i.e. limited battery power, low bandwidth, limited sensing range and low memory. It is pertinent that its resources must be optimally utilized. This paper addresses a secured routing protocol to a specified sink in a multi-sink scenario. It is a subset of a novel algorithm required to securely route to multiple mobile sinks in WSN. It employs reinforcement learning paradigm and in particular (Q-learning) while the transition (action) is modeled as a Partially Observable Markov Decision Process.*

Index Terms: *reinforcement learning, Q-learning, Trust mechanism, computational intelligence, localization*

1. Introduction

WSNs are composed of spatially distributed sensor nodes that cooperatively monitor environmental changes over time. Sensors sense the data and transmit it to the sink (gateway between sensor nodes and end users) through multi-hop routing. A key challenge in the WSN environment is that the resource-constraint (Mac Ruair and Keane, 2007) sensor nodes need to be deployed for prolonged time periods, frequently unattended in remote environments, which not only requires the optimal use of network resources but also the provision of strong security measures. The unreliable wireless channels and unattended operations make it very easy to compromise/capture the nodes. In Nigeria the menace of crude oil pipeline vandalization has cost the federal government huge fortune, hence no cost should be spared in protecting this huge resource. This can be implemented through an appropriate deployment of a secured and energy efficient routing protocol using wireless sensor networks to monitor online environmental phenomenon such as, pressure, temperature and flow rate of the crude oil in the pipes. A lot of effort has gone into secured routing in Wireless sensor networks. The current approach is the combined use of cryptography and trust mechanism, this can be found in the works of RFSN

(Ganeriwal et al., 2008) and TARP (Rezgui and Eltoweissy, 2007). However this approach is not resilient to adversarial nodes capable of compromising this trust mechanism. These adversarial nodes can achieve this by giving false recommendation about neighbour nodes. Secondly these protocols require the explicit model of the network topology, a requirement that will be too much for the memory constrained wireless sensor network nodes. Thirdly in an attempt to isolate adversarial nodes using the trust mechanism, a lot of control packets is included in the protocol which increases network overhead.

This paper employs Q-learning for the protocol design. Q-learning (Walkins C 1995) is a reinforcement learning technique that models sequential decision making in a partially observable environment making it an ideal choice for nodes in WSNs that need to choose a suitable next- hop neighbour to route packets with only limited information Its strength lies in the fact that it doesn't require an explicit model of the network topology, (It updates its Q-value based on the agent's interaction with the environment). It only stores the outcome of the agent's interaction with the environment, hence it can be easily deployed on the memory constrained WSN. It has been shown that Q-learning converges to the optimal action-value function (Bertsekas D. P and Tsitsiklis J. N 2001), (Jaakkola T. et al 1999) However, it suffers from slow convergence, especially when the discount factor γ is close to one (Even E. and Mansour Y 2006), (Szepesv. C.S. 2000) The main reason for the slow convergence of Q-learning is the combination of the sample-based stochastic approximation (that makes use of a decaying learning rate) and the fact that the Bellman operator propagates information throughout the whole space (specially when γ is close to 1). This is taken care of in this protocol because the learning rate here is 1, i.e. the initial Q-value is a function of the number of nodes and neighbour to each nodes, unlike the random value used in the original Q-learning Hence the Q-value is bound to successively reduce and converge more quickly to the optimal value instead of oscillating as in the original Q-value model and secondly each node stores only the routing table of its neighbour nodes instead of all the nodes in the network. This gives the protocol its localized nature.

The use of the Partially Observable Markov Decision Process (POMDP) model within the Q-learning model, (Irissappane et al., 2014) will help to simultaneously address security issues and energy constraints while routing in WSNs. But, the POMDP model for such a decision making problem is large, and even when representing it using factored representations [Poupart, 2005], state-of-the-art off-line solution methods fail to find acceptable POMDP solutions. Though on-line methods [Ross et al., 2008] can improve scalability, they are not applicable due to the energy constraints of WSNs. To overcome the above issues, the transition parameter (routing) in the Q-learning will be modeled using a hierarchical POMDP (called Secure Routing POMDP (SRP)). Factored representation will be employed to address the complexity in solving each SRP component. The SRP hierarchy (Fig. 1) consists of the routing POMDP for making routing decisions, the alarm POMDP for sending/receiving alarms about malicious nodes and the fitness POMDP to compute the fitness (suitability) of nodes to route packets. As major contributions, we: 1) present the SRP model which can optimize the tradeoff between better security and energy savings in WSNs; 2) demonstrate that SRP can effectively deal with black-hole, on-off attacks and other attacks targeting the trust system; 3) conduct extensive evaluation in a simulated and a real-world testbed, showing the effectiveness of SRP against state-of-the-art trust based routing schemes. The above contributions greatly help to facilitate the employment of WSNs in hostile environments., The rest of the paper is organized as follows: Section 2 looks into related works, here current security enabled WSN routing algorithm are highlighted, section 3 considers the problem of secure routing in wireless sensor networks and goes on to explain the different components of the SFROMS protocol proffered in this paper, section 4 provides a detailed description of the SFROMS model using Q-learning, section 5 describes the SFROMS protocol, section 6 shows the results obtained through simulation and hardware test-bed, while section 7 concludes the paper and highlights areas for future research.

2 Related Work

In RFSN: Reputation based Framework for High Integrity Sensor Networks. (Ganeriwal et al., 2008), the quality of a node was determined using the Beta distribution on the cooperation information collected from a watchdog (Marti S et al., 2000) mechanism as well as from recommendations given by other nodes. In TARP (Rezgui and Eltoweissy, 2007) a trust mechanism is employed which isolates routing through malicious nodes by assessing each node neighbour's forwarding ratio using both direct evaluation (RSSI) and recommendation information from other nodes. However, the above trust schemes are not resilient to sophisticated unfair rating attacks which target the trust systems and do not effectively consider the energy constraints of WSNs. In CONFIDANT (Buchegger and Le Boudec, 2002) the authors use a broadcasting mechanism to send alarms about malicious nodes, however it is still susceptible to unfair ratings, where nodes can send false alarms in a sophisticated manner. The broadcast nature of the protocol also makes it memory intensive (i.e it doesn't employ the neighbourhood mechanism where the routing table comprises of routes to only neighbour nodes), and hence infeasible in WSN. In Nurmi, (2007) the author proposed a POMDP based routing scheme that estimates its component states composed of neighbour nodes local parameters (selfishness and energy limitation). However, it uses gradient techniques (shortest path) to determine policies which (as is shown empirically), can be far from optimal. Also, it does not use recommendation information from other sensor nodes. This it does in other to reduce the overhead in memory requirements, taking into cognizance the limited memory capability in WSNs.

Hierarchical POMDP based approaches have been studied in literature to improve on the scalability of the algorithm such as is done in LEACH for routing. (Zhang and Sridharan, 2012; Pineau and Thrun, 2002; Theocharous, 2002; Foka and Trahanias, 2007). Hierarchical POMDP uses action based decomposition (action hierarchy), state space abstraction, or both. In our approach, we consider the action hierarchy as in [Pineau and Thrun, 2002] because the routing problem can be easily partitioned into sub-problems based on the actions (see Fig. 1).

3 Secure Routing Problem for WSNs

A network $N = \{n_i \mid i = 1 \ldots\ldots |N|\}$ of sensor nodes is considered. The neighbourhood of each node n_i consists of sensors reachable within the transmission radius r. Every node independently optimizes its routing behaviour and chooses a next hop neighbour (using the SFROMS model described in Section 5) to route packets to the specified sink. The following paragraphs describe the important aspects involved in this decision problem.

Fitness: (for routing purposes) This parameter denotes the next hop neighbour n_k to a node n_i. A neighbour n_k is chosen based on its fitness (f_k ε { good, bad }) in routing packets, calculated using the fitness factors: residual-energy, distance and routing behaviour

Residual-Energy: $f.e_j$ ε {high, low} denotes the remaining energy in n_i , to route packets. The parameters in (Heinzelman et al., 2000) was used to determine n_i's actual energy $e(n_i)$ and then discretize it (to use standard POMDP solvers): f.ej = high, if e(nj) is greater than half its initial value and low, otherwise.

Distance: Distance $D(n_j$, sink) This parameter denotes the distance of n_j from the sink. It is determined by broadcasting HELLO (source, hopCount) messages, during the sink announcement phase. Initially, (source = sink; hopCount = 0) is broadcast from the sink. The neighbouring nodes of the sink receive this message and determine their distance by incrementing hopCount by 1. The new hopCount is then re-broadcast to each node's neighbours. To discretize the distance values, for node n_k , $f.d_k$ = near, if $D(n_k$, sink) < $D(n_i$, sink), else $f.d_k$ = far.

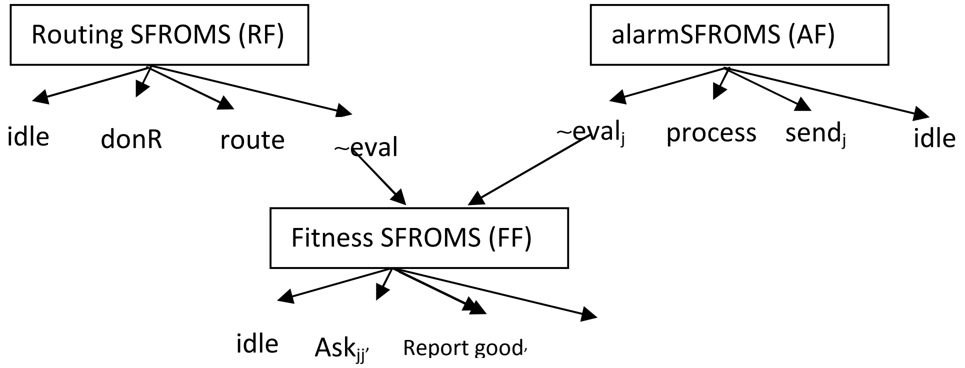
Routing Behavior: This parameter is used to denote the various adversarial capabilities of a node. Node n_k can forward the packets sent to it i.e., $f:rb_k$ = forward, or drop packets $f:rb_k$ = drop, exhibiting network based attacks [Karlof and Wagner, 2003], such as: (1) black-hole attack, where node n_k drops packets with probability pd=1, always; (2) on-off attack, where pd=1 only during specific intervals of time, and pd=0 otherwise, etc.

Message Protocols: In order to determine the above fitness factors of $f_{\mathbf{k}}$, node n_i can request opinions (query action) about node n_k from another neighbour $n_{k'}$ using a QUERY (ni, n_k' , n_k) message . A REPLY (n_k' , n_i, n_k , $f.e_k$, $f.d_k$, $f.rb_k$) message is then sent by n_k' about the fitness factors of n_k . Node n_i can also opt to route packets (route action) to n_k , to determine its fitness factors. Once n_i routes packets to n_k , n_k sends an acknowledgement ACK (n_k , n_i, $f.e_k$, $f.d_k$) message, informing n_i about its residual-energy and distance values. Node n_i also employs a watchdog mechanism (Marti S et al., 2000) to peek n_k's transmission packets (RSSI) and monitor its routing behaviour $f.rb_k$, whether it actually forwards/or drops the sent packets. Thus, the actual values of $f.e_k$, $f.d_k$ and $f.rb_k$ can be determined by routing packets to n_k. Node n_i can also send/receive ALARM (n_i, n_k , $f.e_k$, $f.d_k$, $f.rb_k$) messages, carrying information about a malicious node n_k .

Unfair Ratings: This models neighbour nodes that send false reputation to a QUERY message. In this scenario, node n_k' can be unfair by providing misleading opinions about n_k . A variable r_k' is used to denote the trustworthiness of n_k' in its rating behaviour, when providing opinions about other nodes. n_k' can be truthful (r_k' = true) or provide unfair ratings about n_k , exhibiting any of the following attacks (Jiang et al., 2013): (1) random (r_k' = rand), where n_k' randomly provides fair and unfair ratings, (2) adversarial (r_k' = adv), where n_k' always provides unfair ratings, (3) camouflage (r_k' = dec), where n_k' is honest in the beginning and provides unfair ratings after γ packet transmissions, (4) collusive-unfair (rk' = imp), where attackers form the majority in the system and always promote their neighbours.

Overall Goal: Given that ni can use query and route actions to determine the fitness factors of its neighbours, there exists a tradeoff as querying information can lead to energy drain, while routing through malicious nodes can lead to packet drop. To balance the tradeoff of information gaining (query) actions and exploitation (route/alarm) actions, a Partially Observable Markov Decision Process (POMDP) model is used, which selectively queries for information to select a suitable next-hop neighbour to successfully route packets (and send alarms, if necessary), thereby minimizing energy consumption and maximizing lifetime of the sensor nodes. The next section gives a brief description of the SFROMS model.

Figure 1: Secure Routing SFROMS (SRS) action hierarchy.



4 The SFROMS Model

A SFROMS can be described by the tuple (S, A, T, R, Ω, B): where:

Agent State (S): is defined as $\left(D_p, routes_{D_p}^{N}\right)$ where $D_p \subseteq D$ are the sinks the packets must reach and $routes_{D_p}^{N}$ is the routing information about all neighbouring nodes N with respect to the individual sinks.

Actions (A): This represents a routing decision through a neighbour node to a desired sink. This step is used to determine the sets of secured neighbour (route) to each sink from the sink announcement phase in the network. It is calculated as the number of hops to a desired sink.

A = $\sum_{d \in D_i} hops_d^{n_i}$ – (1)

Where $hops_d^{n_i}$ are the number of hops to reach destination d ε D_i and | Di | is the number of sinks in D.

Transition (T) : This specifies probabilities Pr(s'|s, a) i.e. the probability of transiting from state s to s' given that a certain action 'a' has occurred. It is based on the Partially Observable Markov Decision Process (POMDP) model.

P(s'|s, a) = $\sum_{s \in S} P(s^{'}|s, s^{'}).\ b(s)$ -- (2)

Observation (O) : The agent also receives observations (o ε Ω) based on the observation model O, specifying the probabilities Pr(o|a, s') i.e. the probability of observing a certain reward given that the agent performs an action 'a' having transited to s'. The observation represents the probability distribution of the states. It is given by:

P(o|a, s') = $\sum_{s \in S} P(o|s^{'}).\ P(s^{'}|s, a)$ -- (3)

Reward R(s, a, s') : the reward that an action 'a' causes transition from s to s'. An infinite horizon problem is assumed . It is given by:

R(s, a, s') = $\sum_{s\in S} r(s,a).\ b(s)$ --- (4)
Where r(s, a) = $C_{a_i} + {}^{min}_{\ a}Q(a))$ and b(s) = P(s)
Here C_{a_i} is the action's cost (always 1 in the hop count metric) and ${}^{min}_{\ a}Q(a)$ is the lowest (best) Q-value from the fit neighbours), b(s) is the probability distribution among the neighbour nodes..
Belief (B): This is a probability distribution over states via Bayes' rule. If b(s) specifies the probability of s (∀ s), the updated belief b' after taking action a and receiving observation o is given by,

$$b'(s') = \frac{\Pr(s',o|b,a)}{\Pr(o|b,a)} = \frac{\Pr(o|a,s')}{\Pr(o|b,a)} \sum_s \Pr(s'|s,a)\, b(s) \qquad (5)$$

A SFROMS policy maps beliefs to actions and is associated with a value function Π (b) which evaluates the expected total reward of executing policy Π starting from b. The objective of a SFROMS agent is to find an optimal SFROMS policy Π, which maximizes the expected total reward. Normally, the routing problem in WSNs is too large to be modeled as a single POMDP (finding the optimal policy is intractable, PSPACE complete). I therefore propose an hierarchical approach, consisting of Routing, Fitness and the Alarm subfunction.

.

5 Secure SFROMS (SS)

From the secured model proposed in section 4, a large state space will be required to model parameters needed in the network. This will result in an infinite convergence time for the protocol. In order to address this situation, a hierarchical formulation is proposed here (as shown in Fig 1). This is achieved as follows: Anytime data packet is to be routed from a particular source node, to the sink the Routing SFROMS sub-function is activated. The Routing SFROMS sub-function then calls the Integrity sub-function. The integrity sub-function determines the fitness of a neighbour node using the following parameter: (i) distance of the neighbour to the sink, (ii) Percentage of the full energy remaining in the neighbour node, (iii) the integrity of the neighbour node. In order for this to function appropriately, a danger sub-function may be activated when it is realized that a neighbour node is not fit to route data. The identity of such node is stored in the sub-function so that data packet will not be routed through the node in subsequent time.

5.1 Routing SFROMS (RS)

The aim of the Secure Routing SFROMS sub-function is to determine a neighbour node to route data through. First it must be noted that a node in the network is expected to have more than one immediate neighbour. At the same time, due to the nature of the network, not all the neighbour to a particular node is expected to have a high integrity value. The POMDP model helps build a belief about the neighbour to a node. The ~comp$_k$ variable is used to determine the integrity of a neighbour node. (n$_k$) i.e., f$_k$ = fit(f), if n$_k$ is an honest node and will not drop data packet. and f$_k$ = unfit(u), if n$_k$ is is not honest and will not successfully route data to the destination. The avoid sub-function is activated when there are no fit neighbour nodes to route data packet, in other words the data packet is stuck and may have to be sent back to its previous node so that other alternative route will be sought. The use of the ~comp$_k$ function denotes an effort in the protocol and hence the cost of performing this function is denoted by a negative reward of C(f'$_k$ = u.path) if the neighbour node is an not honest node, while a positive reward of C(f'k = f.path) is provided if the neighbour node is a honest one. The action-space hierarchy for SRS is shown in Fig. 2).

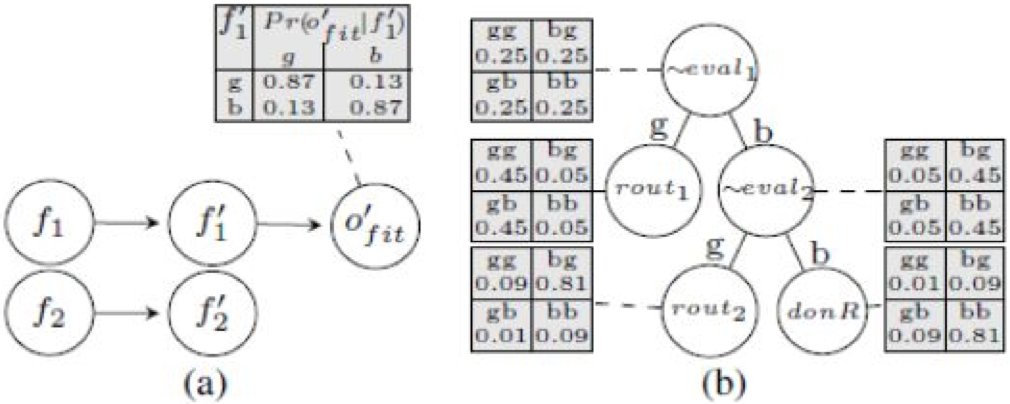


Figure 2: SS: (a) DBN for ~eval1 action; (b) (Partial) Policy.

An example of the ~comp$_k$ action is now explained for a node that has two neighbours, i.e. here, a decision will be made to route data among the two alternatives. Initially equal observation is given to different combination possible under this scenario. This includes: (i) the first and the second neighbour leads to good nodes, this is denote as ff, (ii) the first neighbour is a good node, but the second neighbour is a bad node. This is denoted as fu (iii) the first neighbour is a bad node while the second neighbour is a good node. This is denoted as uf and (iv) the first and second neighbours are bad nodes. This is denoted as uu. The transition and observation probabilities for the ~comp$_{k=1}$ action is shown in Fig.2a). In order to describe this operation, the four different combinations possible from the permutation of the actions are initially assigned the value 0.25. However after the Q-learning model is applied and the nodes perform the actual transition for the cases involved, the actual observation probability is then obtained and it is shown in fig 2. This is explained as follows: after the ~comp$_k$ sub-function is activated actions uu has observation probability of 0.05, fu has observation probability of 0.02, ff has observation probability of 0.45, and uf has observation probability of 0.05. These are the assumed probability given in each case and it served well for the simulation experiments shown in section 6.

Table 1: The SFROMS ~comp$_k$ function table.

SFROMS	States	Observation	Action	Reward
Routing	F_k ε {f,u}	O_{fit} ε {f,u}	~comp$_k$ path$_j$ avoid lazy	C(~comp$_k$) = -20 C(f$_k$'=f,path$_k$) = 200 C(f$_k$'=u,path) = -200 C(lazy) = -5
Danger	F_k ε {f,u} Dangersent ε	O_{fit} ε {f,u}	~comp$_k$	C(~comp$_k$) = -15 C(f'sent=f/u,calculate)=100/

	{1…….m}
	Dangerreceived ε {1…….m}
	DangerDSent_jε {T,F}
		O_{rec} ε {1…….m}
		O_{rand} ε {1……m}
		O_{trans} ε {1…….m}
			Calculate
			trans_j
			lazy
				-200
				C(f'j = u/f,trans = 100/-200
				C(lazy) = -5

Integrity	$f.e_k$ ε {good, bad} $f.d_k$ ε {close, nclose} $f.rb_j$ ε {transmit, no transmit} t_j ε {good,,prob, imp.idle imp.on}	T_{power} ε {good, bad} T_{sept} ε {close, nclose} T_{rb} ε {transmit, no transmit} T_r ε {good, bad}	$\text{Seek}_{jj'}$ Send_fit Send_unfit	$C(\text{seek}_{jj'}) = 20$ C(f'$_{mod}$ = f,sendfit) = 200 C(f$_{mode}$ = u,sendunfit) = -300 C(f'$_{mod}$ = u,sendunfit) = 200 C(f'$_{mod}$ = f,sendunfit) = -300 C(lazy) = -5

prob = random, imp = collusive unfair, dec = camouflag,

5.2 Alarm FROMS (AS)

This sub-function is responsible for triggering the danger signal when a neighbour node is an adversarial node. In other words it is not an honest node. This scenario can happen in two ways: (i) when a node routes to a neighbour node and fails to get an acknowledgement from it. This will translate to either the data packets got lost in transit or the neighbour node is not honest. In this case the Alarm sub-function may or may not be triggered. (ii) When a node seek the integrity of a neighbour node from other nodes in the network. The cumulative rating or belief offered by the other nodes may necessitate activating the Alarm sub-function. In the first instance a provision is made in the protocol to resend the data packet in case it got lost in transit, however if an acknowledgement control packet is not still received, then the Alarm sub-function is activated, and the node is stored as an adversarial node, while further data packet will not be routed through it. In the second case if most of the other nodes which control data packet is sent to determine the integrity of a node replies with negative rating for the node, then the Alarms SFROMS sub-function is activated.

5.3 Fitness SFROMS (FS)

The purpose of this sub-function is to determine the integrity of a neighbour node. The sub-function uses the following parameters to determine the integrity of a node, (i) the neighbour's node distance from the sinks. This is done as follows if the neighbour nodes distance to the sink is farther than that of the requesting node, then the node is unfit otherwise the node is fit. (ii) Percentage of the full energy of the node remaining. This is done as follows if the percentage of the full energy remaining in the neighbour node is more than 60%, then the node is fit, otherwise it is unfit (iii) Node characteristics, this are adversarial behaviour of a neighbour node that makes it to always drop or seldom drop data packets routed through it.. The other use of this sub-function is to identify nodes that give false recommendation about other nodes in the network. This is performed by the rating procedure r_k. When the ~comp$_k$ sub-function is activated it assigns probability to the different types of false recommendation that can be received from other nodes in the network. This includes (i) good recommendation, which shows that a neighbour node is good and fit to route data, (ii) probabilistic recommendation, which shows that a neighbour node may or may not route data packet sent to it, (iii) collusive unfair node recommendation, this shows that an adversarial node has many other sub-nodes which depend on it, routing through such a node will forward data to its sub-nodes which will lead to a condition referred to as face routing. Here a data packet routed through a node will be involved in a high latency path to the destination, such that it leads to short network life times.

Equations 6 – 8 below show the conditional probability for the rating sub-function., In the equation, N_{good} denotes what constitutes a honest node. i.e., number of times f.e$_k$=good, f.d$_k$=close, f.se$_1$=send while N_{bad} denotes what constitutes adversarial nodes i.e., number of times f.e$_k$= no good, f.d$_k$=no close, f.se$_1$=no send.

$b^0(s) = b^0(f_1|f.e_1,f.d_1,frb_1)\ b^0(f.d_1)\ b^0(f.rb_1)$ ------------- (6)

$b^0(f_1 = g \mid f.e_1,f.d_1,frb_1) = \frac{N_{good}}{N_{good+}N_{bad}}$ (7)

$b^0(f_1 = b \mid f.e_1,f.d_1,frb_1) = \frac{N_{bad}}{N_{good+}N_{bad}}$ (8)

The belief updating process in FS follows a similar approach as that of RS (shown in Fig. 3(b)).

5.4 Parallel Belief Update

The belief update takes place separately for RS, AS and FS. For a routing task, RS and FS are active i.e., when RS takes the ~eval$_k$ action, FS is called and based on the reportGood/reportBad action of FS, observation good/bad is received by RS. However, in this case the beliefs about n_k are updated only in RS and left outdated in AS. In order to update the knowledge about n_k in AS, the idle action (in each SFROMS hierarchy) is introduced. Whenever the _evalj action is called in RS and beliefs are updated about n_k , the idle action of AS is also called to update the beliefs on nj . Similarly, when ~evalj action is called by AS, idle action is taken in RS. This idle action gives an update in AS. Whenever the actual fitness factors i.e., f.ek , f:dk and f.rbk of nk are determined after the routing process (as described in Sec. 3), the idle action is taken by FS to update the beliefs about n_k , resulting in reportGood/reportBad action. Based on these actions, the idle actions of both the RP and AP are taken, to update the beliefs about n_k .

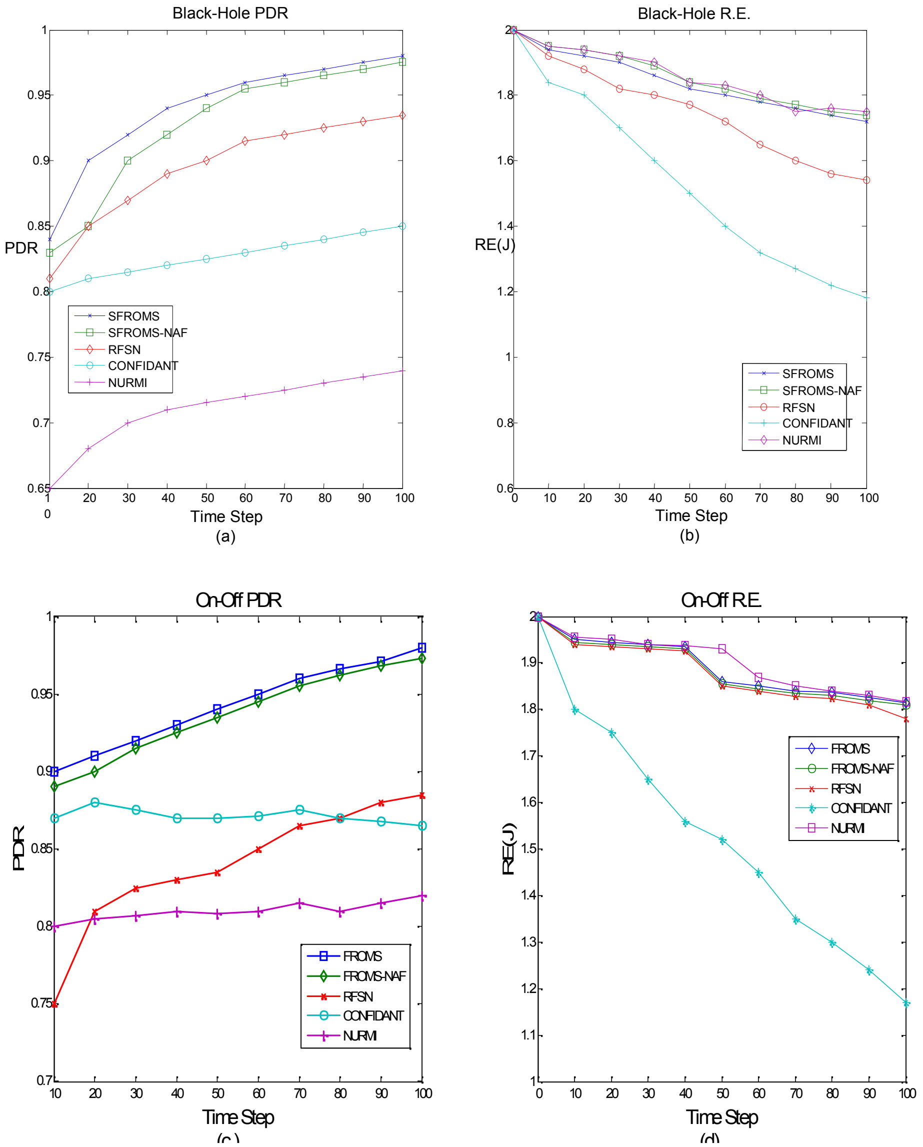
6 Performance Evaluation

Experiments were conducted in a simulated environment as well as on hardware test-bed to compare the performance of SRS with RFSN [Ganeriwal et al., 2008], CONFIDANT [Buchegger and Le Boudec, 2002] and (Nurmi, 2007). To show the usefulness of AS, the results of SRS with and without AS was compared. It is denoted by SRS and SRS-NAS, respectively. To verify the usefulness of the hierarchical structure, SRS was implemented without any hierarchy, but the method failed to find a reasonable solution (due to the large state/action space), thus not shown in the results. The metrics used for characterizing the WSN security are: the average Packet Delivery Ratio (PDR) i.e, ratio of data packets successfully delivered to the sink and Residual Energy (RE) i.e., average (remaining) energy of each sensor node in the network.

In order to learn the observation probabilities for ~eval$_k$ action of RS (AS) based on the policy of FS (for which the probabilities were manually specified). Using (offline) simulations, the protocol was run and 400 actions of RS (AS) which invoke FS were randomly selected. The number of instances where FS correctly identifies a node's quality was measured. The probability of receiving a correct/incorrect observation for ~eval$_k$ action was 0.86/0.14, respectively.

For simulation, the MATLAB Simulator [Barr et al., 2005] was used. The topology includes 100 stationary nodes, uniformly randomly distributed within a 1000m X 1000m square, with the sinks at its right end. The transmission radius is 100m and M=5 (i.e. number of neighbours). Each node generates packets at the rate λ=1 per time step. The size of each data packet is 512 bytes, HELLO packet is 60 bytes, QUERY, ALARM and ACK packet is 125 bytes. The initial energy of each sensor node is 2J. The radio dissipates 50 nJ / bit to run the transmitter/receiver circuitry and 100 pJ / bit for the transmitter amplifier. 20% of the nodes were assumed to be compromised. The experiments are run for 100 time steps, transmitting over 10,000 data packets. Fig. 4(a-h) show the PDR and residual-energy of the routing schemes under different attacking scenarios. In Fig. 3(a), under black-hole attack, SRS and SRS-NAS achieve the highest PDR (97% after 100 time steps). SRS performs slightly better than SRS-NAS, especially in the beginning, as it identifies malicious nodes earlier by propagating alarms. SRP, SRP-NAE, RFSN, CONFIDANT use both direct evaluation and recommendations, while Nurmi uses only direct evaluation, one of the reasons for its low PDR (73%), apart from the limitation of using gradient techniques for computing policy. In Fig. 3(b), SRP obtains a lower residual-energy than SRP-NAE, as SRP additionally sends alarms. RFSN queries all neighbors and CONFIDANT relentlessly sends alarms about malicious nodes, obtaining a lower residual energy. Nurmi does not query other nodes, obtaining a high residual-energy. Fig. 3(c-d) show similar results, where on-off attackers drop packets every 5 time steps.

In Fig. 3(e-h), the 20% compromised nodes (black-hole attackers) also target the trust system by providing unfair ratings (showing random, collusive-unfair behavior). SRP and SRP-NAE can effectively identify unfair raters as they model such behaviours as a part of their POMDP states. In Fig. 3(e-f), under random attack, SRP, SRP-NAE achieve high performance. In Fig. 3(g-h), under collusive-unfair attack, unfair raters are increased to 60%, forming the majority. SRP (PDR 96%) performs better than SRP-NAE as it easily identifies attackers by propagating alarms, while SRP-NAE (PDR 93%) initially obtains misleading opinions from the colluders, thereby routing through malicious nodes, until their actual behaviour is identified after routing. Further Fig. 3(a-h) also show that AE indeed improves the performance of SRP (PDR of SRP is always greater than SRP-NAE, though AE involves additional energy drain, in some cases). Fig. 3(i-l) show the results (under collusive-unfair attack5), when network environment changes. SRP performs better under uniform load (λ=1 per node) as well as under non-uniform load (λ ε [0, 1] is selected randomly per node). Also, PDR of all schemes increase with the number of nodes, as probability of finding a more reliable route to sink increases.

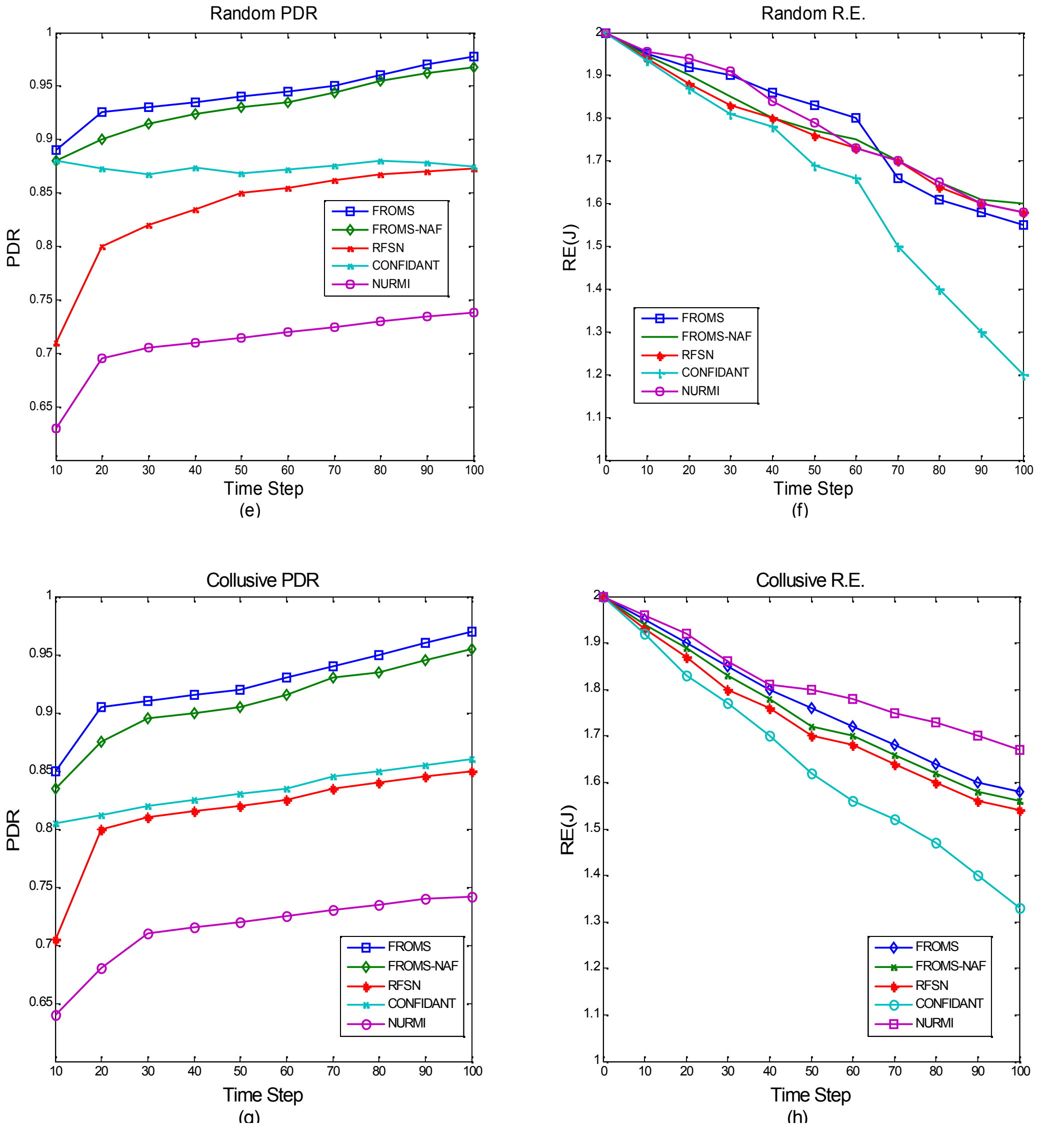

Black-Hole PDR
PDR
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
10
20
30
40
50
60
70
80
90
100
Time Step
SFROMS
SFROMS-NAF
RFSN
CONFIDANT
NURMI
Black-Hole R.E.
RE(J)
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0
10
20
30
40
50
60
70
80
90
100
Time Step
SFROMS
SFROMS-NAF
RFSN
CONFIDANT
NURMI
On-Off PDR
PDR
1
0.95
0.9
0.85
0.8
0.75
0.7
10
20
30
40
50
60
70
80
90
100
Time Step
FROMS
FROMS-NAF
RFSN
CONFIDANT
NURMI
On-Off R.E.
RE(J)
2
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
0
10
20
30
40
50
60
70
80
90
100
Time Step
FROMS
FROMS-NAF
RFSN
CONFIDANT
NURMI

(a)

(b)

(c)

(d)


Random PDR
PDR
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
10 20 30 40 50 60 70 80 90 100
Time Step
FROMS
FROMS-NAF
RFSN
CONFIDANT
NURMI
Random R.E.
RE(J)
2
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
0 10 20 30 40 50 60 70 80 90 100
Time Step
FROMS
FROMS-NAF
RFSN
CONFIDANT
NURMI
Collusive PDR
PDR
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
10 20 30 40 50 60 70 80 90 100
Time Step
FROMS
FROMS-NAF
RFSN
CONFIDANT
NURMI
Collusive R.E.
RE(J)
2
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
0 10 20 30 40 50 60 70 80 90 100
Time Step
FROMS
FROMS-NAF
RFSN
CONFIDANT
NURMI

(e)

(f)

(g)

(h)

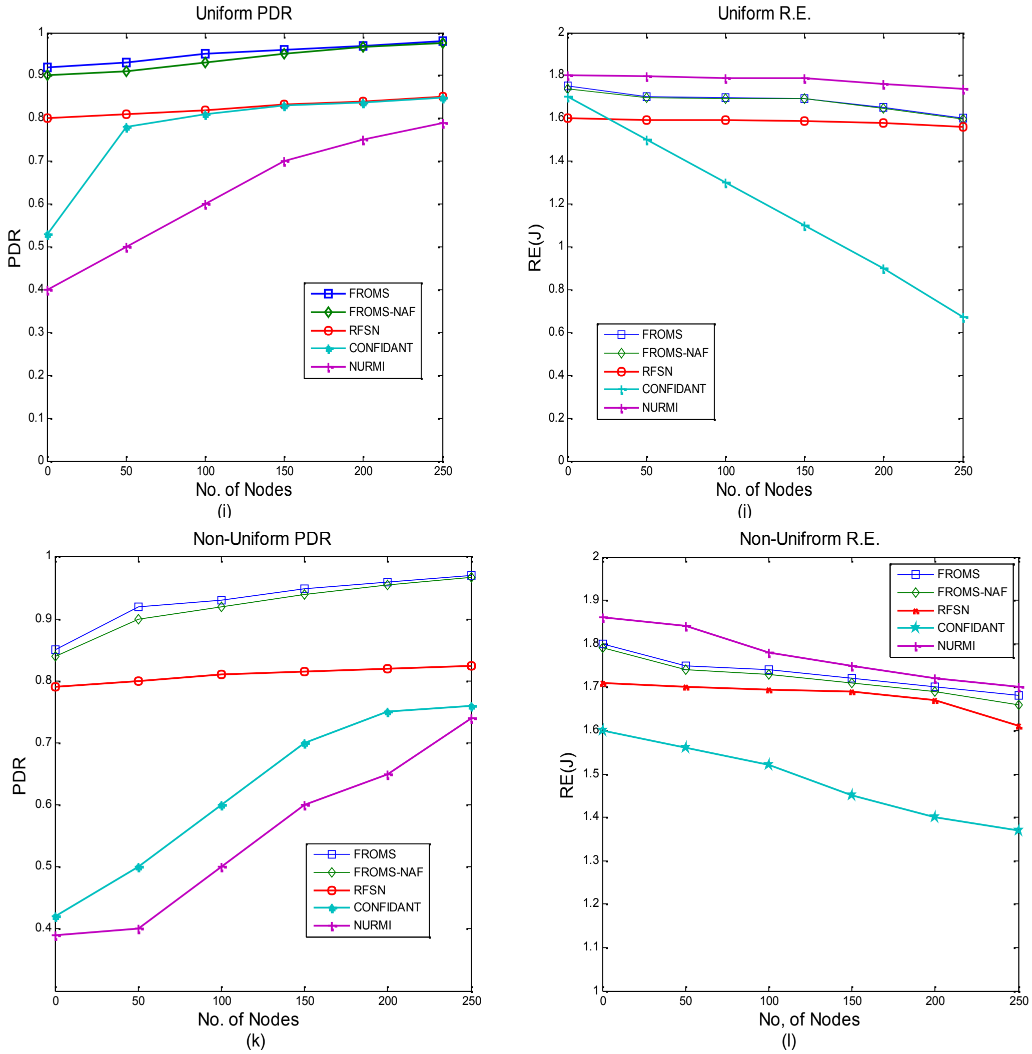


Figure 3: Performance in simulated environment in terms of Packet Delivery Ratio (PDR) and Residual-Energy (RE): (a-h) different attacking scenarios; (i-l) different load characteristics.

Since most probabilities in SRS are manually specified (e.g., random nodes provide unfair ratings with probability pu=50%), we analyze the robustness of SRS to the specification of such values when the actual behaviours of random nodes change: 1) SRP-60, where pu=60% instead of 50% (as assumed in FE); 2) SRP-40, where pu=40%; 3) SRP-50 for perfectly random nodes with pu=50%.

In order to validate the secured routing protocol, in a real-world test-bed, the performance of SFROMS was compared with RFSN, CONFIDANT and Nurmi. The experimental setup consists of arduino-uno (microcontroller), programmable xbees (radio transceiver) and LM 35 temperature sensor (sensing device) the combination of arduino uno xbee and LM 35 temperature sensor acts as the end device while the combination of the arduino uno and xbees acts as the router and co-ordinator nodes. The results of SFROMS were compared with and without AS denoted by SFROMS and SFROMS-NAS, respectively. Two performance metrics were used for comparison: The average Packet Delivery Ratio (PDR) i.e, the ratio of data packets successfully delivered to the sink together with Residual-Energy (RE) i.e., average (remaining) energy of each sensor node in the network

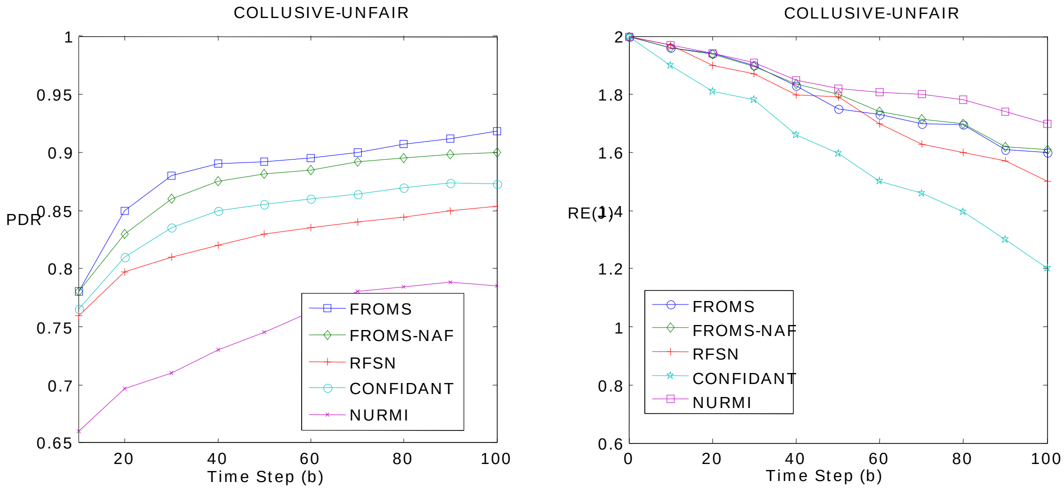


Figure 4 : Graph on the Collusive unfair adversarial nodes (Hardware Test-bed)

7 Conclusion and Future Work

The Secure Routing SFROMS (SRS) approach is presented in this paper, to select suitable next-hop neighbours and successfully route packets to the sink. It is a subset of the protocol to route to multiple mobile sinks. SRS can deal with black-hole, on-off attacks, etc., and attacks targeting the trust system. It balances the exploration/exploitation tradeoff in gaining/exploiting information about sensor nodes, thereby effectively addressing their energy constraints. SRS is modeled using hierarchical and factored representations to address the complexity in solving POMDPs. Experiments show that SRS consistently achieves higher packet delivery rates by coping with various attacks, while still maintaining high residual energy. Hence it guarantees reliable, energy-efficient routing in WSNs, which are key factors in sustainable development. While it has been established that SRS is robust against the choice of parameters for transition and observation models, an interesting direction of future work is to automatically optimize these. Attempt will be made in future to investigate using finite-state controllers, which can be more energy-efficient (Grze´s et al., 2013).

References

Amalia Foka and Panos Trahanias, (2007). Real-time hierarchical POMDPs for autonomous robot navigation. Robotics and Autonomous Systems, 55(7):561–571, 2007.

Barr et al., (2005). Scalable wireless ad hoc network simulation. Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad hoc Wireless, and Peer-to-Peer Networks, pages 297–311.

Buchegger and Le Boudec, (2002) Performance analysis of the CONFIDANT protocol (Cooperation of nodes: Fairness in dynamic ad-hoc networks). In MobiHoc, 2002, pages 102 - 116

Dyo et al., (2010). Evolution and sustainability of a wildlife monitoring sensor network. In SenSys, 2010, pages 34 – 52.

Ganeriwal Saurabh, Ganeriwal, Laura K Balzano, and Mani B Srivastava, (2008) Reputation-based framework for high integrity sensor networks. ACM Transactions on Sensor Networks (TOSN), Vol 4 pages 3 -15.

Grze´s et al., (2013). Controller compilation and compression for resource constrained applications. Algorithmic Decision Theory, pages 193–207.

Heinzelman et al., (2000). Energy efficient communication protocol for wireless microsensor networks. Las Vegas McGraw Hill Publication

Irissappane Athirai, Frans A Oliehoek, and Jie Zhang (2014). A POMDP based approach to optimally select sellers in electronic marketplaces. In AAMAS, pages 67 - 79.

[Jiang Siwei , Jie Zhang, and Yew-Soon Ong (2013) . An evolutionary model for constructing robust trust networks. In AAMAS, pages 180 - 194.

Kaelbling Leslie, Michael L Littman, and Anthony R Cassandra (1998). Planning and acting in partially observable stochastic domains. Artificial intelligence, Vol 101(1) pages 99–134.

Karlof Chris and David Wagner, (2003). Secure routing in wireless sensor networks: Attacks and countermeasures in Ad hoc networks, Vol 1 pages 293–315.

Kok-Lim Alvin Yau A, Peter Komsarczuk, Paul D. Teal (2012). Reinforcement Learning For Context awareness & Intelligence in Wireless Networks: Review, New Feature & Open Is-sues." Elsevier‟s Journal of Network and Com-puter Applications. Vol 35 pages 253-267.

Liu fang , Xiuzhen Cheng, and Dechang Chen (2007). Insider attacker detection in wireless sensor networks. In INFOCOM, pages 123 - 136.

Mac Ruair´ı and Mark T Keane, (2007). An energy-efficient, multi-agent sensor network for detecting diffuse events. In IJCAI, pages 23-40.

Marti Sergio, Thomas J Giuli, Kevin Lai, and Mary Baker (2000). Mitigating routing misbehavior in mobile ad hoc networks. In MobiCom, pages 56 -74.

Nurmi Petteri (2007). Reinforcement learning for routing in ad hoc networks. In WiOpt, pages 124 - 138.

Pineau Joelle and Sebastian Thrun, (2002). An integrated approach to hierarchy and abstraction for POMDPs. Carnegie Mellon University Technical Report CMU-RI-TR pages 2-21.

Poupart Pascal (2005). Exploiting structure to efficiently solve large scale Partially Observable Markov Decision Processes. PhD thesis, University of Toronto.

Rom´an Rodrigo, Carmen Fernandez Gago, Javier L´opez, and Hsiao Hwa Chen (2009). Trust and reputation systems for wireless sensor networks. Security and Privacy in Mobile and Wireless Networking, pages 105– 128.

Ross St'ephane, Joelle Pineau, S´ebastien Paquet, and Brahim Chaib-draa (2008). Online planning algorithms for POMDPs. Journal of Artificial Intelligence Research, Vol 32(1) pages 663–704.

Stone Peter and Manuella Veloso, (1998). Layered approach to learning client behaviours in the robocup soccer server. Applied Artificial Intelligence, Vol 12 pages 165–188.

Sun Zhu Yan, Lindsay,Han, Wei Yu, and KJ Ray Liu (2002). A trust evaluation framework in distributed networks: Vulnerability analysis and defense against attacks. In INFOCOM, pages 78 - 91.

Theocharous Georgios (2002). Hierarchical learning and planning in partially observable Markov decision processes. PhD thesis, Michigan State University.

Yu Yanli , Keqiu Li, Wanlei Zhou, and Ping Li (2012). Trust mechanisms in wireless sensor networks: Attack analysis and countermeasures. Journal of Network and Computer Applications, Vol 35 pages 867–880.

Zhang Shiqi and Mohan Sridharan, (2012). Active visual sensing and collaboration on mobile robots using hierarchical POMDPs. In AAMAS, pages 45 – 54.